SELF-TUNING PREDICTIVE CONTROL

by

Maciej W. Rogoziński, M.Sc.

A thesis submitted in fulfilment of the requirement for the degree of

Doctor of Philosophy

in the

Department of Electrical and Electronic Engineering

The University of Adelaide

December, 1987

Awarded: March 21, 1988.

to my parents

moim Rodzicom

Contents

Abstract.

This thesis is concerned with the development of self-tuning predictive controllers for both deterministic and stochastic systems and assessment of their properties by analysis and/or by simulation. Self-tuning controllers for both multivariable and scalar systems are proposed.

New characterizations, called left and right nilpotent interactor matrices, of the delay structure of a multivariable linear system are introduced. Computational algorithms operating on a numerical representation of polynomial matrices are proposed for the on-line calculation of such characterizations. The algorithms are amenable to computer-based calculations using a matrix-oriented software. These results are employed in the development of a new self-tuning, minimum prediction error controller which requires significantly less prior system knowledge than other strategies.

Long-range predictive controllers (resulting from minimization of multi-stage cost functions) are considered for application in self-tuning control. Among the new developments presented are the self-tuning, long-range predictive, receding horizon controller proposed for a multivariable deterministic system and the analysis of the closed-loop system resulting from the long-range predictive controller for a scalar stochastic system.

The problem of utilization of additional system outputs for feedback in self-tuning control is addressed. Self-tuning, minimum prediction error controllers for one output of multi-output deterministic and stochastic systems are proposed. It is shown that utilization of additional outputs for feedback is likely to improve the convergence rate of parameter estimates of self tuners markedly and does modify certain assumptions about the controlled stochastic system. Of particular significance is the removal of the strictly positive real condition required for convergence of self-tuning controllers based on the feedback from the controlled output only. The global convergence of self-tuning controllers with feedback from additional outputs is established using existing methods of convergence analysis.

Likewise adaptive prediction which is based on the measurement of additional system outputs is considered. Adaptive predictors for one output of multi-output deterministic and stochastic systems are developed. Possible benefits resulting from measurement of additional outputs involve, in particular, improvement in the convergence rate of parameter estimates of adaptive predictors and removal of the strictly positive real condition.

Statement.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any University. To the best of the author's knowledge and belief the thesis contains no material previously published or written by another person, except where due reference is made in the text.

The author hereby consents to this thesis being made available for photocopying and loan if accepted for the award of the degree.

M. W. Rogoziński

Acknowledgements.

I gratefully acknowledge the guidance, inspiration and constructive criticism of my supervisor, Dr. M. J. Gibbard, during the course of work for this thesis. In particular I would like to thank him for helping in the style and presentation of the thesis.

I would also like to thank sincerely my co-supervisor, Dr. A. P. Papliński, for his unfailing assistance and many helpful discussions.

I wish to thank the staff of the Department of Electrical and Electronic Engineering of The University of Adelaide and my colleagues for creating a stimulating and supportive environment for research.

List of principal symbols.

Symbol	Description	Defined in section
A	matrix of coefficients of $A(p)$ (eqn. 3.2, p. 59)	3.1
A_c	block top-companion matrix (eqn. 3.33, p. 68)	3.1.1
A_i	coefficient matrix of $A(p)$ (eqn. 3.2, p. 59)	3.1
A_p	matrix consisting of coefficients of $\alpha^{(k_p)}(p)$	5.2.1
A_P	matrix consisting of coefficients of $\alpha^{(j)}(p)$ (eqn. 5.11, p. 155)	5.1.1
A_s	matrix for state-space models	2.1.1
$\tilde{A}$	matrix of the state-space model of the closed-loop system with the LRP controller (eqn. 5.37, p. 191)	5.2.2
$\widehat{A}_i$	estimate of A_i	4.2
$\widehat{A}_P$	matrix A_P calculated using $\widehat{A}(p)$	5.1.2
$A(p)$	polynomial matrix of DARMA and ARMAX models	2.1.2
$\widehat{A}(p)$	estimate of $A(p)$	4.2
$\hat{A}(q)$	polynomial matrix of LDO models	2.1.2
$A(z^{-1})$	denominator polynomial matrix of the LMF description	2.3.3
b^R	vector defined by eqn. 3.114, p. 99	3.3.1
B	matrix of coefficients of $B(p)$ (eqn. 3.3, p. 59)	3.1
B_f, B_{fT}	matrices consisting of coefficients of $\beta_F^{(k_p)}(p)$	5.2.1
B_F, B_{FT}	matrices consisting of coefficients of $\beta^{(j)}(p)$	5.1.1
B_g	matrix consisting of coefficients of $\beta_G^{(k_p)}(p)$	5.2.1
B_i	coefficient matrix of $B(p)$ (eqn. 3.3, p. 59)	3.1
B_p	matrix consisting of coefficients of $\beta_F^{(k_p)}(p)$	5.2.1
B_P	matrix consisting of coefficients of $\beta^{(j)}(p)$ (eqn. 5.10, p. 155)	5.1.1
B_{rt}	matrix in the LRP control law for a TISO system	5.2.1
B_{rT}	matrix in the LRP control law (eqn. 5.16, 156)	5.1.1
B_s	matrix for state-space models	2.1.1
B_S, B_{st}, B_{ST}, B_X	matrices of the LRP control law in the state-space form	5.2.2

Symbol	Description	Defined in section
$\tilde{B}$	matrix of the state-space model of the closed-loop system with the LRP controller for a TISO system	5.2.2
$\widehat{B}$	estimate of B	4.2
$\widehat{B}_i$	estimate of B_i	4.2
$\widehat{B}_{FT}$, $\widehat{B}_P$	matrices B_{FT} and B_P calculated using $\widehat{A}(p)$ and $\widehat{B}(p)$	5.1.2
$\mathcal{B}^{(t)}$	matrix of coefficients of $\mathcal{B}^{(t)}(z^{-1})$	2.3.3
$B(p)$	polynomial matrix of DARMA and ARMAX models	2.1.2
$\widehat{B}(p)$	estimate of $B(p)$	4.2
$\overline{B}(p)$	polynomial of the ARMAX model with delay factored from $B(p)$	5.2.1
$\hat{B}(q)$	polynomial matrix of LDO models	2.1.2
$B(z^{-1})$	numerator polynomial matrix of the LMF description	2.3.3
$\mathcal{B}^{(t)}(z^{-1})$	polynomial matrix resulting from the final iteration of the algorithm calculating the RNI matrix (eqn. 2.66, p. 48)	2.3.3
C	control horizon for the LRP control laws	5.1; 5.2
C, C_F, C_s, C_S	matrices for state-space models	2.1.1
c_ω	vector of coefficients of $c_\omega(p)$ (eqn. 3.92, p. 93)	3.3
c_ω^L, c_ω^R	vectors defined by eqns. 3.107 (p. 97) and 3.111 (p. 98)	3.3.1
C_χ	parameter of the CNM technique	App. E
C_ω	matrix of coefficients of $C_\omega(p)$ (eqn. 3.94, p. 93)	3.3
$C(p)$	polynomial matrix of the ARMAX model	2.1.2
$\hat{C}(q)$	polynomial matrix of the LDO model	2.1.2
$c_\omega(p)$	noise polynomial of the RDO model of TIMO and TISO systems (eqn. 3.92, p. 93)	3.3
$C_\omega(p)$	noise polynomial vector of the RDO model of a TIMO system (eqn. 3.94, p. 93)	3.3
$\hat{c}_\omega(q)$, $\hat{C}_\omega(q)$	polynomial matrices of the RDO model	2.1.2
$\widehat{c}_\omega(t,p)$	estimate of $c_\omega(p)$	6.2.2
$c_A(z^{-1})$	polynomial approximating $c_\omega(z^{-1})$	6.2.3
$c_\omega(z^{-1})$	noise polynomial of TIMO and TISO systems	3.3.3

Symbol	Description	Defined in section
d	vector of coefficients of $d(p)$ (eqn. 3.62, p. 79)	3.2
D, D_F	matrices for state-space models	2.1.1
$d(p)$, $\hat{d}(q)$	open-loop system characteristic polynomial	3.2; 2.1.2
$\hat{D}(q)$	polynomial matrix of RDO models	2.1.2
$\hat{D}_\omega(q)$	polynomial matrix of the RDO model	2.1.2
$D(t,p)$	filter for the RELS or RML estimators	App. E
$d_L(z)$	diagonal factor of $\xi_L(z)$ (eqn. 2.36, p. 37)	2.3.1
$d_R(z)$	diagonal factor of $\xi_R(z)$ (eqn. 2.38, p. 37)	2.3.1
$D(z^{-1})$	denominator polynomial matrix of the RMF description	2.3.2
$e(t)$	equation error of a recursive parameter estimator	App. E
$e_r(t)$	closed-loop system tracking error (eqn. F.8, p. 305)	App. F
$E\{\ \}$	expectation operator	
f	$\dim y_F(t)$	2.1
F	matrix of coefficients of $F(p)$ (eqns. 3.46 (p. 72), 3.100, p. 94)	3.1.2; 3.3
F_s	vector of coefficients of $F_s(p)$	3.3.3
$F^{(j)}$	matrix of coefficients of $F^{(j)}(q)$ (eqn. 3.9, p. 62)	3.1.1
$\mathcal{F}_t$	sequence of increasing sub-sigma algebras	3.3
$f^{(k_p-k+1)}(p)$	polynomial in the Diophantine equation 5.27, p. 182	5.2.1
$F(p)$	polynomial (matrix) in Diophantine equations 3.44 (p. 72) and 3.99 (p. 94)	3.1.2; 3.3.1
$F_s(p)$	polynomial in the Diophantine equation 3.138, p. 108	3.3.3
$F^{(k_p)}(p)$	polynomial in the Diophantine equation 5.26, p. 181	5.2.1
$F^{(j)}(q)$	polynomial matrix in the Diophantine eqn. 3.7, p. 62	3.1.1
$\widehat{F}(p)$	polynomial matrix $F(p)$ calculated using $\widehat{A}(p)$	4.2
$g^{(k_p-k+1)}(p)$	polynomial in the Diophantine equation 5.27, p. 182	5.2.1
$H_{y,u}(z)$	system transfer matrix	2.1.2
I_m	$m \times m$ identity matrix	
J_k	cost function 4.1, p. 120	4.1
$J_{P,C}$	cost functions 5.1 (p. 152), 5.3 (p. 153), 5.23 (p. 181)	5.1; 5.2

Symbol	Description	Defined in section
$J(t+k)$	cost functions 6.1 (p. 197), 6.38 (p. 217)	6.1.1; 6.2.1
k	delay for a scalar system and degree of the interactor matrix for a multivariable system	2.3
k_l	degree of the determinant of the LNI matrix	2.3.2
k_p	prediction horizon for predictors for TISO and TIMO systems	3.3.1; 5.2.1
k_r	degree of the determinant of the RNI matrix	2.3.3
K	matrix for state-space models	2.1.1
$\overline{K}_L$, $\overline{K}_R$	matrices of coefficients of the left and right interactor matrices	3.1.1; 3.1.2
$K(p)$	feedback polynomial matrix in the GLCL	2.2
$\tilde{K}_R(p)$	polynomial matrix defined by eqn. 3.42, p. 72	3.1.2
$\widehat{K}_R(q)$	polynomial matrix $K_R(q)$ calculated using $\widehat{B}(p)$	4.2
$K_L(z)$, $K_R(z)$	left and right nilpotent interactor matrices	2.3.2; 2.3.3
$\overline{K}_L(z), \overline{K}_R(z)$	left and right interactor matrices	2.3.1
l	$\dim u(t)$	2.1
m	$\dim y(t)$	2.1
M_L, M_R	matrices defined by eqns. 2.40, 2.41, p. 42	2.3.1
$M(p)$	precompensator polynomial matrix in the GLCL	2.2
n	degree of the polynomial matrix $\hat{A}(q)$ or of the polynomial $d(p)$	2.1.2; 3.2
n_p	degree of polynomials of the SIMO-type and TIMO-type predictors	3.2.1; 3.3.1
n_A, n_B, n_C	degrees of polynomials $A(p)$, $\overline{B}(p)$, $C(p)$ of the ARMAX model 5.22, p. 180	5.2.1
N, N_F	matrices of coefficients of $N(p)$ and $N_F(p)$	3.2; 3.3
$\widehat{n}$	upper bound for n	4.2
$N(p)$	polynomial of RDO models of SIMO and TIMO systems (eqns. 3.63 (p. 79), 3.91, p. 93)	3.2; 3.3
$N_F(p)$	polynomial vector of RDO models of SIMO and TIMO systems (eqns. 3.64 (p. 79), 3.93, p. 93)	3.2; 3.3
$\hat{N}(q)$, $\hat{N}_F(q)$	polynomial matrices of RDO models	2.1.2
$N(z^{-1})$	numerator polynomial matrix of the RMF description	2.3.2

Symbol	Description	Defined in section
p	forward shift operator	App. A
P_s	$(s+1)\times 1$ vector of powers of the indeterminate p of a polynomial matrix	App. B
$P(t)$	covariance matrix of the estimate error of a recursive parameter estimator	App. E
$P_m(t)$	covariance matrix of the estimate error of the RLS-CNM estimator	App. E
q	backward shift operator	App. A
Q_1, Q_2	operating points of the robot manipulator	4.3
R	vector of future values of the filtered reference sequence (eqn. 5.8, 154)	5.1.1
$R_i^{r,c}$	$r \times c$ matrix containing ones along the $(i+1)$-th diagonal and zeros elsewhere (eqn. B.4, p. 283)	App. B
R_s	vector of future values of the reference sequence (eqn. 5.30, p. 182)	5.2.1
$r(t)$	reference sequence value at time t	2.2
s	$\dim u_D(t)$	2.1
S	shift matrix defined by eqn. 3.70, p. 81	3.2
S_F, S_P	matrices consisting of coefficients of $S(p)$ (eqns. 5.12, 5.13, p. 155)	5.1.1
$s(p)$	weighting polynomial (eqn. 6.4, p. 197)	6.1.1
$S(p)$	weighting polynomial matrix	4.1; 5.1.1
$S_L^{(i)}(z)$	factor of the LNI matrix (eqn. 2.45, p. 44)	2.3.2
$S_R^{(i)}(z)$	factor of the RNI matrix (eqn. 2.61, p. 47)	2.3.3
t	sample instant	App. A
T	matrix defined by eqn. 5.14, p. 155	5.1.1
T_s	sampling period	4.3
$t(p)$	weighting polynomial (eqn. 6.3, p. 197)	6.1.1
$T_L(z)$	triangular factor of $\xi_L(z)$ (eqn. 2.36, p. 37)	2.3.1
$T_R(z)$	triangular factor of $\xi_R(z)$ (eqn. 2.38, p. 37)	2.3.1
U_c, U_p, U_P, U_S	vectors of past control signals in LRP control laws	5.2.1; 5.1.1
U_t, U_T	truncated control sequence vectors (eqns. 5.28 (p. 182), 5.4, p. 154)	5.2; 5.1.1
$u(t)$	control signal	2.1

Symbol	Description	Defined in section
$u_c(t)$	filtered control signal (eqn. 5.25, p. 181)	5.2.1
$u_D(t)$	plant disturbances	2.1
$u_k(t)$	auxiliary control signal	4.1
$u^o(t)$	control signal defined by eqn. 6.28, p. 203	6.1.2
$\tilde{u}(t)$	auxiliary control signal (eqn. 6.2, p. 197)	6.1.1
$\overline{u}(t)$	system inputs	2.1.1
$U(z)$	r.s.p.m. (eqn. 3, App. C)	2.3.2
$U_R(z)$	c.s.p.m. (eqn. 2.57, p. 47)	2.3.3
v_{k_p}	vector defined by eqn. 3.129, p. 103	3.3.2
$V(p)$	polynomial matrix in the GLCL	2.2
$w(p)$	closed-loop system characteristic polynomial (eqn. 6.11, p. 198)	6.1.1
$x(t)$, $x(0)$	state vector and initial state	2.1.1
$x_L(t)$	left partial state vector	2.1.2
$x_R(t)$, $x_{R\omega}(t)$	right partial state vectors	2.1.2
$\hat{x}(t\|t-1)$	Kalman filter estimate of $x(t)$ given data up to time $t-1$	5.2.2
$X(z)$	z-transform of $x(t)$	App. A
Y_c	vector of past filtered outputs for the LRP control law	5.2.1
Y_P	vector of past outputs (eqn. 5.7, p. 154)	5.2.1
$y(t)$	controlled outputs	2.1
$y_c(t)$	filtered output (eqn. 5.25, p. 181)	5.2.1
$y_F(t)$	feedback outputs	2.1
$\tilde{y}(t)$	*a posteriori* prediction of $y(t)$	App. F
$\overline{y}(t)$	system outputs	2.1.1
$y_k(t)$	auxiliary system outputs (eqn. 3.5, p. 61)	3.1.1
$y^o(t+k_p)$	optimal k_p-step-ahead prediction	3.3.1; 5.2.1
$\hat{y}(t)$	adaptive prediction of $y(t)$	3.2.2; 3.3.4
z_{pi}	system poles	4.3
z_{zi}	roots of the equation $\det B(z^{-1}) = 0$	4.3

Symbol	Description	Defined in section
Z_s	$(s+1)\times 1$ vector of powers of the indeterminate z^{-1} of a polynomial matrix	App. B
α	vector of coefficients of $\alpha(p)$ (eqn. 3.65, p. 79)	3.2.1
α_{ij}	element of α_j associated with the i-th output	3.2.1; 3.3.2
α_j	coefficient vector of $\alpha(p)$ or $\alpha_\omega(p)$	3.2.1; 3.3.1
$\alpha^{(j)}$	matrix of coefficients of $\alpha^{(j)}(p)$ (eqn. 3.10, p. 62)	3.1.1
$\alpha_i^{(j)}$	coefficient matrix of $\alpha^{(j)}(p)$ (eqn. 3.10, p. 62)	3.1.1
$\widehat{\alpha}^{(j)}$	$\alpha^{(j)}$ calculated using $\widehat{A}(p)$	5.1.2
α_ω	vector of coefficients of $\alpha_\omega(p)$ (eqn. 3.97, p. 94)	5.2.1
$\alpha(p)$	polynomial vector of the SIMO-type predictor (eqn. 3.65, p. 79)	3.2.1
$\alpha^{(j)}(p)$	polynomial matrix of the multi-step-ahead predictor (eqn. 3.10, p. 62)	3.1.1
$\alpha^{(k_p)}(p)$	polynomial of the predictor 5.24, p. 181	5.2.1
$\alpha_s(p)$	polynomial of the SISO-type predictor 3.71, p. 81	3.2.1
$\alpha_{s\omega}(p)$	polynomial of the TISO-type predictor 3.137, p. 108	3.3.3
$\alpha_\omega(p)$	polynomial vector of the TIMO-type predictor (eqn. 3.97, p. 94)	5.2.1
$\alpha(t)$	forgetting factor	App. E
$\widehat{\alpha}(t)$	estimate of α	3.2.2
$\widehat{\alpha}_\omega(t)$	estimate of α_ω	3.3.4
β	vector of coefficients of $\beta(p)$ (eqn. 3.66, p. 79)	3.2.1
β_j	coefficient of $\beta(p)$ or $\beta_\omega(p)$	3.2.1; 3.3.1
$\beta^{(j)}$	matrix of coefficients of $\beta^{(j)}(p)$ (eqn. 3.11, p. 62)	3.1.1
$\beta_i^{(j)}$	coefficient matrix of $\beta^{(j)}(p)$ (eqn. 3.11, p. 62)	3.1.1
$\widehat{\beta}^{(j)}$	$\beta^{(j)}$ calculated using $\widehat{A}(p)$ and $\widehat{B}(p)$	5.1.2
β_ω	vector of coefficients of $\beta_\omega(p)$ (eqn. 3.98, p. 94)	5.2.1
β_ω^L, β_ω^R	vectors defined by eqns. 3.115, 3.116, p. 99	3.3.1
$\beta(p)$	polynomial of the SIMO-type predictor (eqn. 3.66, p. 79)	3.2.1
$\beta^{(j)}(p)$	polynomial matrix of the multi-step-ahead predictor (eqn. 3.11, p. 62)	3.1.1

Symbol	Description	Defined in section
$\beta_F^{(k_p)}(p)$	polynomial of the predictor 5.24, p. 181	5.2.1
$\beta_G^{(k_p)}(p)$	polynomial of the predictor 5.24, p. 181	5.2.1
$\beta_s(p)$	polynomial of the SISO-type predictor 3.71, p. 81	3.2.1
$\beta_{s\omega}(p)$	polynomial of the TISO-type predictor 3.137, p. 108	3.3.3
$\beta_\omega(p)$	polynomial of the TIMO-type predictor (eqn. 3.98, p. 94)	5.2.1
$\widehat{\beta}(t)$	estimate of β	3.2.2
$\widehat{\beta}_\omega(t)$	estimate of β_ω	3.3.4
γ	matrix of coefficients of $\gamma(p)$ (eqn. 3.47, p. 72)	3.1.2
γ_i	coefficient matrix of $\gamma(p)$ (eqn. 3.47, p. 72)	3.1.2
$\gamma(p)$	polynomial matrix of the k-step-ahead predictor (eqn. 3.47, p. 72)	3.1.2
$\widehat{\gamma}$	γ calculated using $\widehat{A}(p)$ and $\widehat{B}(p)$	4.2
δ	matrix of coefficients of $\delta(p)$ (eqn. 3.48, p. 72)	3.1.2
δ_i	coefficient matrix of $\delta(p)$ (eqn. 3.48, p. 72)	3.1.2
$\delta(p)$	polynomial matrix of the k-step-ahead predictor (eqn. 3.48, p. 72)	3.1.2
$\widehat{\delta}$	δ calculated using $\widehat{A}(p)$ and $\widehat{B}(p)$	4.2
$\Delta(p)$	weighting polynomial matrix	5.1.1
$\delta(t-s)$	Kronecker delta	2.1.1
$\epsilon(t)$	prediction error (eqn. F.2, p. 305)	App. F
$\zeta(t)$	scaling factor for the CNM technique (eqn. E.10, p. 300)	App. E
$\eta(t)$	*a posteriori* prediction error (eqn. F.7, p. 305)	App. F
$\theta(t)$	vector of robot joint angles	4.3
Θ	matrix of parameters to be estimated	App. E
Θ_s	vector of parameters defined by eqn. E.12, p. 300	App. E
$\widehat{\Theta}(t)$	recursive estimate of Θ	App. E
$\widehat{\Theta}_s(t)$	recursive estimate of Θ_s	App. E
ϑ	matrix defined by eqn. 3.77, p. 84	3.2.1
ϑ_ω	matrix defined by eqn. 3.87, p. 92	3.3
λ	control weighting coefficient	5.2.1

Symbol	Description	Defined in section
$\lambda_{\max}$, $\lambda_{\min}$	maximum and minimum eigenvalue of the estimate error covariance matrix	6.2.3
$\overline{\lambda}$	control weighting coefficient defined by eqn. 6.12, p. 198	6.1.1
Λ	control weighting matrix	4.1; 5.1.1
ν_F	observability index of the pair (C_F, A_s)	3.2
$\xi_L(z)$	(unique) left interactor matrix (eqn. 2.36, p. 37)	2.3.1
$\xi_R(z)$	(unique) right interactor matrix (eqn. 2.38, p. 37)	2.3.1
ρ	constant defined by eqn. 6.6, p. 198	6.1.1
σ^2	variance of the scalar noise sequence $\{\omega(t)\}$	3.3
$\sigma^2_{e_r}$	tracking error variance (eqn. 6.67, p. 240)	6.2.3
$\sigma^2_{e_r}(t)$	on-line tracking error variance (eqn. 6.68, p. 240)	6.2.3
σ^2_y	variance of the output sequence $\{y(t)\}$	6.2.3
Υ	output weighting matrix	4.1; 5.1.1
$\phi(t)$	regression vector	App. E
$\phi_s(t)$	regression vector defined by eqn. E.13, p. 300	App. E
$\chi(t)$	variable of CNM defined by eqn. E.6, p. 299	App. E
$\chi(-1)$	initial value of $\chi(t)$ for CNM	App. E
$\psi(t)$	gradient vector defined by eqn. E.17, p. 301	App. E
$\omega(t)$	white noise	2.1.1
deg $Q(x)$	degree of a polynomial matrix $Q(x)$	App. B
det Q	determinant of a matrix Q	
$\dim x$	dimension of a (column) vector x	
rank Q	rank of a matrix Q	
Q'	transpose of a matrix Q	
$Q \otimes R$	Kronecker product of matrices Q and R	
$\langle D \rangle_s$	resultant of order s of the polynomial matrix $D(x)$ (eqn. B.5, p. 283)	App. B
$\langle D \rangle_s^L$	matrix consisting of the first $(s+1)m$ columns of $\langle D \rangle_s$, where m is the number of columns of $D(x)$	
$\|x\|^2_L$	vector norm defined as $\|x\|^2_L = x'Lx$	
$\{x(t)\}$	sequence $x(t)$	App. A

List of abbreviations.

Abbreviation	Description	Defined in section
AML	Approximate Maximum Likelihood	App. E
ARMAX	AutoRegressive Moving-Average with auXiliary input	2.1.2
a.s.	Almost Surely	3.3
CARIMA	Controllable AutoRegressive Integrated Moving-Average	2.1.2
CARMA	Controllable AutoRegressive Moving-Average	2.1.2
CDC	Continually Disturbed Control	App. G
CNM	Condition Number Monitoring	App. E
c.s.p.m.	Column Shift Polynomial Matrix	2.3.3
DARMA	Deterministic AutoRegressive Moving-Average	2.1.2
DMC	Dynamic Matrix Control	5
FD	feedback configuration FD	2.1
FI	feedback configuration FI	2.1
FIR	Finite Impulse Response	4.1
GLCL	General Linear Control Law	2.2
GMV	Generalized Minimum Variance	5.2.1
GPC	Generalized Predictive Control	5
KTL	Key Technical Lemma	App. F
LDO	Left Difference Operator	2.1.2
LDU	Lower-Diagonal-Upper (or Gaussian elimination) decomposition	2.3.4
LMF	Left Matrix Fraction	2.3.3
LNI	Left Nilpotent Interactor	2.3.2
LNI-LDU	LNI matrix calculated with LDU decomposition	5.1.2
LNI-QR	LNI matrix calculated with QR decomposition	5.1.2

Abbreviation	Description	Defined in section
LQG	Linear Quadratic Gaussian	1
LRP	Long Range Predictive	5
LS	Least Squares	App. G
MIMOS	Multi-Input, Multi-Output, Square	3.1
MPE	Minimum Prediction Error	2.3
MPE-SIMO	MPE controller for a SIMO system	6.1.1
MPE-SISO	MPE controller for a SISO system	6.1.1
MUSMAR	MUltiStep Multivariable Adaptive Regulator	App. G
MV	Minimum Variance	6.2
MVC	MV Controller	5.1.1
MV-TIMO	MV controller for a TIMO system	6.2.1
NPEC	Normalized Prediction Error Convergence	App. F
ODE	Ordinary Differential Equation	App. G
PID	Proportional-Integral-Derivative	1
PRBS	Pseudo-Random Binary Sequence	3.2.2
QR	QR decomposition	App. C
RDO	Right Difference Operator	2.1.2
RELS	Recursive Extended Least Squares	App. E
RELS-CNM	RELS with CNM	App. G
RLS	Recursive Least Squares	App. E
RLS-CNM	RLS with CNM	App. E
RMF	Right Matrix Fraction	2.3.2
RML	Recursive Maximum Likelihood	App. E
RML1	RML version 1	App. E
RML2	RML version 2	App. E
RNI	Right Nilpotent Interactor	2.3.3

Abbreviation	Description	Defined in section
r.s.p.m.	Row Shift Polynomial Matrix	2.3.2
SFT	Spectral Factorization Theorem	3.3.3
SG	Stochastic Gradient	3.3.4
SIMO	Single-Input, Multi-Output	3.2
SISO	Single-Input, Single-Output	3.2
SKTL	Stochastic KTL	App. F
SPR	Strictly Positive Real	App. G
ST-LRP	Self-Tuning LRP controller	5.1.2
ST-MPE-L-SIMO	Self-Tuning MPE-SIMO controller in a linear form	6.1.2
ST-MPE-RNI	Self-Tuning MPE controller based on the RNI matrix	4.2
ST-MPE-SIMO	Self-Tuning MPE-SIMO controller	6.1.2
ST-MPE-SISO	Self-Tuning MPE-SISO controller	6.1.2
ST-MV-TIMO	Self-Tuning MV-TIMO controller	6.2.2
ST-MV-TISO-RELS	Self-tuning MV controller for a TISO system employing the RELS estimator	6.2.3
ST-MV-TISO-RML	Self-tuning MV controller for a TISO system employing the RML estimator	6.2.3
ST-WMV-TISO	Self-Tuning WMV-TISO controller	6.2.2
SVD	Singular Value Decomposition	2.3.4
TIMO	Two-Input, Multi-Output	3.3
TISO	Two-Input, Single-Output	3.3
U-D	U-D decomposition	5.1.2
WLS	Weighted LS	App. G
WMV	Weighted MV	6.2
WMV-TIMO	WMV controller for a TIMO system	6.2.1
WMV-TISO	WMV controller for a TISO system	6.2.1

Chapter 1

Introduction.

Subject coverage.

During the last two decades self-tuning controllers have become increasingly important in a wide range of process control applications. This thesis is concerned with the development of self-tuning predictive controllers for both deterministic and stochastic systems, and an assessment of their properties by analysis and/or by simulation. Both the multivariable and scalar systems are considered.

One of the essential problems is to relax prior system knowledge required by a self-tuning controller, especially in applications involving multivariable systems. For this purpose new characterizations of the delay structure of a multivariable linear system are introduced; such characterizations can be determined on-line using a new class of computational algorithms. These results are employed in the development of a new minimum prediction error self tuner which requires significantly less prior system knowledge than other strategies.

A self-tuning controller should stabilize a wide variety of plants (e.g., nonminimum phase and/or unstable open-loop plants). Furthermore, the self tuner should be robust to modeling errors (e.g., to model overparameterization). Moreover, it is desirable for the control law embedded in the self tuner to have performance-oriented tuning knobs which are meaningful in the time domain. In order to achieve these features, a new long-range predictive self-tuning controller is developed.

The convergence rate of self-tuning controllers is often insufficient for practical applications. The problem of improving the convergence rate of self tuners is addressed by utilization of additional system outputs for feedback. Possible benefits resulting from such an approach are examined and compared with the conventional approach based on feedback from the controlled output only. Not only is the convergence rate shown to be superior to that of conventional self tuners, but also some restrictive system assumptions, such as the strictly positive real condition associated with stochastic disturbances, are removed.

Adaptive and self-tuning control.

Adaptive control theory is often described as the most recent phase in the development of feedback control systems theory. An adaptive control system can be defined as follows [1]:

> **control system, adaptive.** A control system within which automatic means are used to change the system parameters in a way intended to improve the performance of the control system.

The task of improving the performance of the control system is accomplished by an adaptive controller, i.e., a controller which is capable of modifying its behaviour in response to changes in the controlled system (plant) dynamics and the disturbances. For this purpose a mechanism is needed for "learning" about the characteristics of the system during its operation. Such a mechanism facilitates adaptation of the control strategy not only to a changing environment, but also to the unknown characteristics of the plant and disturbances.

Various approaches to adaptive control have been considered. (Surveys describing adaptive control schemes developed from stochastic control theory, gain scheduling, model reference adaptive systems, and self-tuning controllers are presented in [2,3,4,5].) For example, adaptive controllers can be developed within the framework of optimal nonlinear stochastic control theory [2,3,5]. The solution is very complex even for simple cases (see example in [6]). In general, the strategies resulting from optimal stochastic control theory perform the simultaneous tasks of realizing the desired performance (control) and reducing system uncertainty (estimation) [7]. Uncertainty arises due to unpredictable stochastic variables (e.g., measurement noise) and unknown parameters of the system [7]. There are two methods of reducing uncertainty [7]: passive observation of system variables, and active experimentation on the system (probing). Controllers involving the latter method are called *dual*, and those based on passive observation are referred to as *non-dual* [8]. The dual controller must compromise between the conflicting tasks of control and estimation. The non-dual controller is based on separation of estimation and control.

Non-dual controllers can be divided into two classes depending on the information which is passed from the estimator to the controller [2]. If the information consists of the estimates of unknown parameters and the uncertainties of the estimates, then the controller can take into account errors in the estimates to exercise cautious control action; such a controller is called *cautious*. If the information consists only of parameter estimates, then the control signal is determined as if the estimates were the true parameters, i.e., the controller does not take into account inaccuracy of estimates. Such a controller is called a *certainty equivalence controller* [7,8].

The adaptive controllers which are considered in this thesis are usually referred to as *self-tuning controllers* because they have facilities for tuning their own parameters [5]. Self-tuning controllers are certainty equivalence controllers. The purpose of a self-tuning controller is to control a plant the parameters of which are unknown.

A diagram which represents a self-tuning control system is depicted in fig. 1.1. In order to design a self-tuning controller, the structure of a model of the unknown plant is specified. If the self-tuning controller is to be implemented on a digital computer (as are those considered in this thesis), it is convenient to choose a parametric, discrete-time plant model. It is usually assumed that the plant is linear and time-invariant. The next step in the design is to select the feedback control strategy (i.e., the control law) as for the plant with known parameters. One can also try to reparameterize the plant model so that the model is expressed in terms of the parameters of the chosen control law.

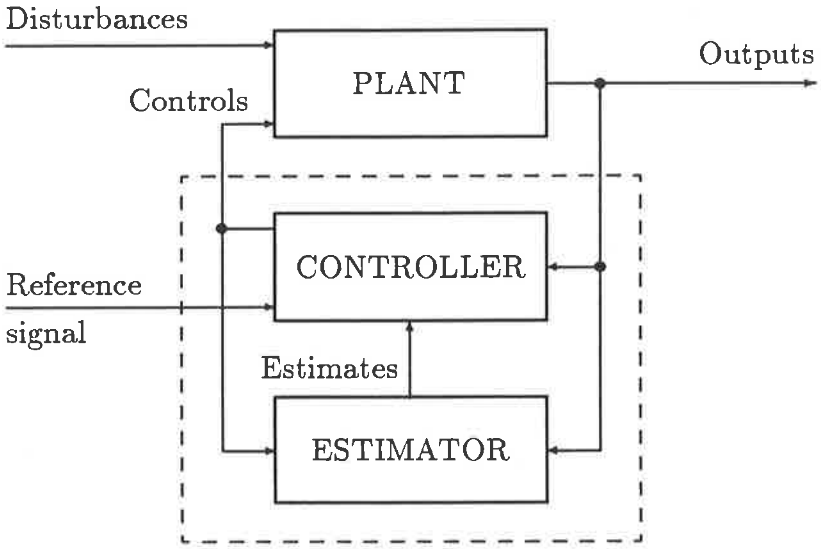

Figure 1.1: Block diagram of a self-tuning control system.

During the operation of the self-tuning controller, the unknown parameters of the model are identified on-line, by a recursive parameter estimator, from measurements of plant outputs and inputs. The estimated model is either that of the plant or that of the control law if the above-mentioned reparameterization is performed. If the parameters of the plant are estimated then the block labeled "controller" in fig. 1.1 performs two tasks. The first task is to calculate parameters of the control law from the estimated plant parameters. (This task represents an on-line solution to a design problem for a plant with known parameters.) The second task is to calculate the control signal given parameters of the control law, past control signals, measured plant outputs and reference signal. Such a self tuner is called *indirect* or *explicit* [5]. However, if the parameters of the control law are estimated then the design calculations are eliminated and the controller performs only the second task. Such a self tuner is called *direct* or *implicit* [9].

Various plant models, control strategies and estimation methods can be combined to yield a self-tuning controller [10,5]. Examples of (i) plant models are the difference operator and state-space models, (ii) control strategies are minimum prediction error, long-range predictive, pole-placement, LQG, and PID control laws, (iii) estimation methods are stochastic gradient, least squares, extended and generalized least squares, instrumental variables, extended Kalman filtering, recursive maximum likelihood.

Self-tuning controllers were initially proposed to control time-invariant plants. However, the strategies can be modified - or made robust - to cope with slowly time-varying plants (see e.g., [11,12,13,14]). Recently self-tuning strategies have been proposed for time-varying systems (see e.g., [15,16,17,18]).

Self-tuning controllers have found numerous applications despite initial criticism following unsuccessful attempts to design adaptive controllers for high-performance aircrafts in the 1950s [5]. The revolutionary progress in microelectronics, dating from the early 1970s, has made it possible to implement self-tuning controllers relatively simply and cheaply using microprocessor systems. A number of commercial self-tuning controllers is available on the market (e.g., Electromax V based on the PID structure (1981), ASEA Novatune (1982) [5]); some self tuners are being developed at universities (e.g., UNAC [19]). As pointed out in [3], possible realizations of self-tuning controllers range from single-loop PID controllers to self tuners capable of controlling complex industrial processes involving significant delays, multiple resonances, nonlinearities etc. The applications of self tuners involve control of autopilots for aircraft, missiles and ships, chemical reactors, distillation columns, glass furnaces, heat exchangers, paper machines, power systems, rolling mills, ore crushers, diesel engines, robots, biomedical devices, etc. (see [20,4,5] for further examples and references).

Review of the state of the art and contributions of this thesis.

Considerable research devoted to the development and analysis of self-tuning controllers has been carried out over the last two decades. The highlights of the main contributions to the state of the art, which are relevant to the present work, will now be reviewed. Furthermore, some contributions of this thesis will also be mentioned.

The first self-tuning controller (although not described as such) was proposed by Kalman (1958) who built a dedicated hybrid computer to implement an adaptive controller [21]. The strategy is based on *ad hoc* separation of identification and control. Such a heuristic approach is often considered for the development of self-tuning controllers [10].

A self-tuning regulator minimizing the variance of the measured output and employing the recursive least squares estimator was introduced by Åström and Wittenmark in 1973 [22]. This self tuner is simple to implement, but is *sensitive* to nonminimum phase plant characteristics which often arises as a result of sampling. Moreover, *excessive* control effort is often required and can only be decreased by increasing the sampling period. This self tuner does *not* facilitate tracking of the reference signal.

In order to overcome the above-mentioned drawbacks of the minimum variance self-tuning regulator, the generalized minimum variance self-tuning controller was proposed by Clarke and Gawthrop (1975) [23], (1979) [24]. This self tuner facilitates control of nonminimum phase plants by weighting the control effort and is simple to implement. Furthermore, this self tuner has a number of useful interpretations such as, for example, a (detuned) model-reference adaptive controller, and a self-tuning least-squares predictor in a Smith-type control configuration [25].

The single-stage cost functions postulated for the development of the above strategies penalize the error between the controlled output and the reference signal k-step-ahead, where k represents the plant delay of k sample intervals from the control input to the controlled output. We shall call the controllers resulting from minimization of such a performance criterion *minimum prediction error controllers.*

Minimum prediction error controllers are highly *sensitive* to the assumptions made about the value of delay k and to time-varying delay. Furthermore, for nonminimum phase systems the choice of the values of the tuning knobs which guarantees closed-loop system stability is not straight-forward for plants with unknown parameters. On the other hand, as pointed out in [26], for application in self-tuning control the control strategy is required:

- to *stabilize* a wide variety of plants (e.g., nonminimum phase and/or unstable open-loop plants);

- to have *performance-oriented tuning knobs* which are meaningful in the time domain;
- to *eliminate* the effect of disturbances;
- to be *robust* to modeling errors (e.g., under- and overparameterization of the plant model order, incorrect assumptions about delay).

In order to overcome drawbacks in minimum prediction error controllers and to fulfil the above requirements for the control strategy, *long-range predictive controllers* were postulated for application in self-tuning control (see Peterka (1984) [27], Ydstie (1984) [28], Clarke *et. al.* (1984) [29]). These controllers are based on minimization of multi-stage cost functions with an optimization horizon extending beyond the plant delay. At the cost of increased complexity and computational burden, the resulting self tuners have performance-oriented tuning knobs which *facilitate* control of nonminimum phase and/or unstable plants, are *robust* to unknown or variable plant delay, and to model order under- and overparameterization.

One of the contributions of this thesis is the development of a new long-range predictive self-tuning controller for multivariable deterministic systems. This strategy possesses a number of features desirable in self-tuning control which are not achievable for the minimum prediction error self tuners. Furthermore, a rigorous closed-loop system analysis is presented for the long-range predictive controller for scalar stochastic systems. Such an analysis is not available for the related strategies which were proposed in [30,29].

Both the minimum prediction error and the long-range predictive controllers will be referred to as the *predictive controllers.*

Self-tuning predictive controllers were originally developed for scalar systems. Certain difficulties have been encountered, however, in the extension of self-tuning schemes to multivariable systems. These difficulties are associated with the requirement of prior knowledge of the system delay structure.

In early work on self-tuning control of multivariable systems only *special* forms of the delay structure were considered (see e.g., Borisson (1979) [31], Goodwin *et. al.* (1980) [32]). The *general* delay structure of a multivariable linear system is characterized by the interactor matrix introduced by Wolovich and Falb (1976) [33]. The left interactor matrix was employed by Goodwin and Long (1980) [34], and the right interactor matrix by Tsiligiannis and Sovronos (1986) [35], to develop self-tuning minimum prediction error controllers for deterministic systems with the general delay structure. Corresponding results for self-tuning minimum variance control of multivariable stochastic systems were presented by Dugard *et. al.* (1984) [36].

The above-mentioned self-tuning strategies for multivariable systems require *prior knowledge* of the system delay structure. In the case of a general form of the delay structure, prior knowledge of the interactor matrix is *assumed.* This is, however, tantamount to the knowledge of the system transfer matrix, thus greatly reducing the appealing features of self-tuning control.

The following approaches were proposed to relax the requirement of complete prior knowledge of the interactor matrix (further references are presented in subsection 2.3.1). Dugard *et. al.* (1984) [37] developed a self tuner in which a part of the interactor matrix is estimated on-line. Long-range predictive self tuners for deterministic and stochastic systems were introduced by Ydstie and Liu (1984) [38] and by Dugard *et. al.* (1984) [36], respectively; for these strategies knowledge of the interactor (polynomial) matrix is reduced to an upper bound on its degree. It was reported, however, that the long-range predictive strategy of [38] may fail to stabilize open-loop unstable plants [29]. The indirect minimum prediction error self-tuning controller of Elliott and Wolovich (1984) [39] involves the on-line calculation of the interactor matrix; this scheme obviates the need for prior knowledge of the interactor matrix.

One of the contributions of this thesis is a new minimum prediction error self-tuning controller for which the requirement of prior knowledge of the interactor matrix is *eliminated.* The required prior system knowledge is reduced not only in comparison with other predictive strategies, but also in comparison with the pole-placement techniques which are discussed below. For the purpose of development of this self tuner, *new* characterizations of the delay structure are proposed as an alternative to the interactor matrix (which has been so far the only such a characterization). A new class of computational algorithms, which are employed in the self tuner, is introduced for the *on-line* calculation of such characterizations.

As an alternative to predictive controllers, the pole placement method was considered for application in self-tuning control. Self-tuning controllers were proposed which assign the poles of the closed-loop system [40,41,42,43], or both its poles and zeros [44]. The advantage of pole placement self tuners is that they do *not* require prior knowledge of the system delay structure, can be made *robust* to variable delays, and are not sensitive to nonminimum phase systems. However, the *indirect* self-tuning pole-placement controllers are computationally demanding due to the on-line solution of a Diophantine equation, and are *sensitive* to the plant model order overparameterization (when a near-cancellation of a pole-zero pair is encountered). Furthermore, for multivariable systems *prior knowledge* of the system observability indexes and an upper bound on the controllability index is required [39]. The *direct* pole-placement self tuners involve estimation of a large number of parameters and are *not robust* to underparameterization [45]. Furthermore, for multivariable systems *prior knowledge* of the system controllability indexes and an upper bound on the observability index is required [43]. The open-loop system is assumed to be controllable which implies that some *ad hoc* methods for elimination of the effect of deterministic disturbances must be employed.

Self-tuning controllers have been also developed for linear state-space system models involving Gaussian noise processes and based on quadratic cost functions (LQG self tuners) (see e.g., [46,26]). Generally, the LQG self tuners involve the following operations [26]: plant parameter estimation, state estimation, solution of the Riccati equation, and calculation of the control signal. Hence, such self-tuning controllers are computationally involved, especially for multivariable systems. The LQG self tuners are *sensitive* to model overparameterization; furthermore, the choice of appropriate control weighting may be difficult [29]. However, they are robust to incorrect assumptions about plant delay [26].

Self-tuning PID controllers have been proposed (see e.g., [47]) mainly because of a wide use of nonadaptive PID controllers. These self tuners can be applied to a *limited* range of systems (e.g., described by a second order transfer function), and yield unsatisfactory performance for plants with significant delay [20].

In this thesis we shall consider self-tuning predictive controllers. Some of their advantages and drawbacks in comparison with other approaches to self-tuning control were discussed above. Furthermore, for self-tuning predictive controllers the desired performance of the closed-loop system is defined in terms of a criterion which determines a *trade-off* between output behaviour and control effort. Such an approach is especially useful if the plant is *time-varying*, since then the requirement of a fixed closed-loop behaviour (resulting for example from pole placement) may lead to excessive control effort [48].

Another advantage of predictive control is that it is possible to use future values of the reference signal (programmed control [27]) to introduce *anticipatory* control action in order to improve the input-output performance of sluggish systems, possibly with large delays [49].

Self-tuning control can be applied to *nonlinear* plants. The first approach is to develop a self tuner for a linearized plant model. This approach relies on estimation of parameters which are time-varying due to changes in the plant operating point. The second approach is to develop a self tuner for the *nonlinear* plant model. *Predictive* controllers are well-suited for the latter approach [9]. Potential improvements in the performance of self-tuning predictive controllers applied to nonlinear plants and developed for nonlinear plant models in comparison with those based on linear models are illustrated, for example, in [50]. This constitutes another reason to consider predictive controllers for application in self-tuning control.

Let us now turn our attention to *analysis* of self-tuning controllers. Two important aspects of such an analysis are *stability* (boundedness of the control signal and outputs) and *convergence* (in the sense that desired system performance is asymptotically achieved).

It was shown for the self-tuning minimum variance regulator employing the least squares estimator that *if* the parameter estimates converge to *any* limiting values, then the resulting control law is optimal [22]. However, the problem of convergence of parameter estimates remained unresolved; furthermore, the closed-loop system stability was addressed in [51] only with a heuristic argument.

The next major step in the analysis of self-tuning controllers was due to Ljung (1977) [52] who associated the asymptotic properties of recursive stochastic algorithms with the solutions of an ordinary differential equation (ODE method). The ODE method *assumes* closed-loop system stability. Using the ODE method, Ljung established sufficient conditions for the convergence of the minimum variance/stochastic gradient self tuner to the minimum variance control law [53]. This analysis revealed importance of the *strictly positive real* condition which is related to the characteristics of the stochastic disturbance.

The convergence results of [52,53] were obtained assuming closed-loop system stability. However, such an assumption is *restrictive* for analysis of self-tuning control systems [54]. Therefore, it is desirable to establish conditions for *global convergence* of self-tuning controllers. A self-tuning controller is said to be globally convergent if, for all initial states of the system and the algorithm, the control signal sequence is bounded, and (i) for a deterministic system, outputs track asymptotically the reference sequence [32], or (ii) for a stochastic system, the mean-square output tracking error is minimized (with probability one) [55].

The breakthrough in the analysis of self-tuning controllers is due to Goodwin, Ramadge, and Caines. They established *global convergence* of the minimum prediction error self-tuning controllers for deterministic (1980) [32] and stochastic (1981) [55] systems. Martingale method of analysis was employed to establish global convergence of the strategy [55] for stochastic systems.

The minimum variance self-tuning controller of [55] employs the stochastic gradient estimator, i.e., a *scalar gain* algorithm. However, the convergence rate of scalar gain estimators is *inferior* to that of estimators based on the least squares method, i.e., *matrix gain* algorithms. Considerable research has been devoted to *improving* *the convergence rate* of (globally convergent) self tuners by employing matrix gain cstimators. Therefore, the work of Kumar and Moore (1982) [56], and of Sin and Goodwin (1982) [57], is of particular importance (further references are presented in Appendix G). They proposed *modifications* to the matrix gain estimators required in order to guarantee global convergence of stochastic self-tuning controllers based on such estimation algorithms.

One of the contributions of this thesis is the development of self-tuning, minimum prediction error controllers for both deterministic and stochastic systems for which there some other outputs available for feedback apart of the controlled output. It is shown that utilization of such *additional outputs* for feedback *improves markedly the convergence rate* of self-tuning controllers. The new self tuners employ matrix gain estimators. Global convergence of the self-tuning controllers is established.

Convergence of self-tuning controllers for stochastic systems has been established subject to the strictly positive real condition [53]. Such a condition appears in analysis of recursive stochastic algorithms using the ODE method, martingale method, or combination of both. Only a few methods have been developed to overcome this restrictive condition. For example, a sophisticated, globally convergent self-tuning strategy, which overcomes the strictly positive real condition, was proposed by Moore (1984) [58] (further references are presented in Appendix G). The methods of overcoming the strictly positive real condition are *computationally demanding*, involve dither injection and therefore may lead to *suboptimal* performance, or require estimator overparameterization which affects the *convergence rate*, etc.

It is shown as one of the contributions of this thesis that utilization of additional outputs for feedback in self-tuning control is a *simple* and *effective* method of overcoming the strictly positive real condition.

Outline of thesis.

Chapter 2 is concerned with system modeling in self-tuning predictive control.

A unified model of a system with the outputs which are to be *controlled* and the outputs which are available for *feedback* is postulated in section 2.1. Both state-space models and difference operator models are defined; these models are commonly used in self-tuning control, and are presented here for the completeness of exposition.

The requirements of an admissible control law are presented in section 2.2. Furthermore, aspects of the output measurement configuration are discussed for self-tuning control of systems with two groups of outputs (i.e., with the output variables which are to be controlled and those which are available for feedback).

In subsection 2.3.1 a summary is presented of approaches to multivariable self-tuning predictive control with respect to prior system knowledge required by the self tuner. Such knowledge involves the interactor matrices as characterizations of a multivariable system delay structure. An assessment of the requirements of an admissible control law and of the properties of the interactor matrices leads to the development of new characterizations of the delay structure (subsections 2.3.2 and 2.3.3). These characterizations are called the left and right *nilpotent interactor matrices.* A new class of *computational algorithms,* which operate on a numerical representation of polynomial matrices, is proposed for the calculation of nilpotent interactor matrices. The details and examples are presented in Appendix C, reprinted from the IEEE Transactions on Automatic Control.

In chapter 3, *output predictors* are introduced for the development of self-tuning controllers in the following chapters. Nilpotent interactor matrices are employed in predictors for multivariable systems in section 3.1. New algorithms, which are amenable to *computer-based calculations* using matrix-oriented software, are derived for the calculation of predictor parameters. Furthermore, a predictor structure is postulated which utilizes *additional system outputs.* Based on this structure, new predictors are derived for one output of both a multi-output deterministic system (section 3.2) and a stochastic system (section 3.3). Computational methods based on a numerical representation of the Diophantine equations are developed for the *off-line* calculation of predictor parameters for systems with known parameters. For systems with unknown parameters, *adaptive* predictors are proposed. The properties of the new predictors are discussed and compared with those of conventional predictors.

The results of sections 2.3 and 3.1 are employed in chapters 4 and 5. In chapter

4 a new self-tuning minimum prediction error strategy is developed for multivariable deterministic systems, possibly affected by a deterministic disturbance. The novelty of the strategy lies, in particular, in the application of the algorithm for the *on-line* calculation of the right nilpotent interactor matrix from the estimates of the system model. This approach is aimed at *reducing* the required system prior knowledge. An example of application of such a self-tuning controller to a robot manipulator is described.

Self-tuning, long-range predictive, receding horizon control is proposed for both multivariable deterministic systems (section 5.1) and scalar stochastic systems (section 5.2). The left nilpotent interactor matrix is employed in the development of the controller for multivariable systems. The self-tuning long-range predictive controllers possess two time-oriented tuning knobs which facilitate control of nonminimum phase and/or unstable open-loop systems. The influence of the tuning knobs on the closed-loop system performance is established by simulation studies. An example of application of such a self-tuning controller to a robot manipulator is described. Furthermore, closed-loop system analysis, involving a stability criterion, is presented for scalar stochastic systems.

The problem of *utilization of additional system outputs* for feedback in self-tuning minimum prediction error control of one output of a multi-output system is considered in chapter 6. The results of sections 3.2 and 3.3 are employed in the development of self-tuning controllers for deterministic and stochastic systems, respectively. It is shown that utilization of additional outputs for feedback is likely to *improve the convergence rate* of self tuners markedly in comparison with that of conventional self-tuning controllers. This improvement is due to the reduction in the number of estimated parameters in the case of deterministic systems, and due to the replacement of the pseudo-linear regression estimators by the linear regression estimator in the case of stochastic systems. Moreover, the utilization of additional outputs for feedback modifies certain assumptions about the controlled stochastic system. Of particular significance is the *removal of the strictly positive real condition* required for convergence of conventional self-tuning controllers.

Theoretical results obtained for self-tuning controllers with utilization of additional outputs for feedback are illustrated by simulation studies. *Global convergence* of these self-tuning controllers is established, using existing methods of convergence analysis, in Appendix F. A survey of self-tuning control strategies developed for stochastic systems and aimed at similar goals to those achieved by utilization of additional outputs for feedback is presented in Appendix G.

In chapter 7 conclusions are drawn regarding the new results presented in the thesis, and from these conclusions a number of suggestions for future research is presented.

The computer software used in the simulation studies presented in chapters 4 and 5 forms a useful tool for assessment of the performance of self-tuning multivariable control systems. It consists of macro-instructions written in MATLAB (or MATRIX$_X$) commands and is included in Appendix D. Appendices A, B, and E, contain material frequently used throughout the thesis.

Chapter 2

System modeling in self-tuning predictive control.

This chapter discusses modeling aspects of linear systems which are relevant to self-tuning predictive control, in situations where the systems are affected by deterministic or stochastic disturbances.

A unified model of a system with the outputs which are to be controlled and the outputs available for feedback is introduced in section 2.1. Two feedback configurations are considered for the model: firstly, a configuration in which the controlled and feedback output variables are identical; secondly, a configuration in which some, or perhaps none, of the feedback variables are the same as controlled variables. The derivation of the difference operator system representation results in (i) the left difference operator (DARMA and ARMAX) models associated with the first feedback configuration; (ii) the right difference operator models associated with the second feedback configuration.

The requirements of an admissible control law and the measurement configuration of a system with two groups of outputs for self-tuning control are presented in section 2.2.

In order to fulfil the requirements of the admissible control law, a characterization of the delay structure of a linear multivariable system will be employed in the development of predictive controllers. New characterizations of the delay structure, called the nilpotent interactor matrices, are introduced in section 2.3. Given the right and left difference operator representations of the system, algorithms for the evaluation of the left and right nilpotent interactor matrices are developed. The material presented in section 2.3 is the main contribution of this chapter.

2.1 System modeling.

In this section:

- the structure of a system with two groups of inputs and two groups of outputs is postulated for the development and analysis of self-tuning control algorithms;
- mathematical models of the system are introduced;
- aspects of the modeling of both deterministic and stochastic disturbances are discussed.

Consider a plant with two groups of vector inputs, $u(t)$ and $u_D(t)$, and two groups of vector outputs, $y(t)$ and $y_F(t)$, as shown in fig. 2.1.

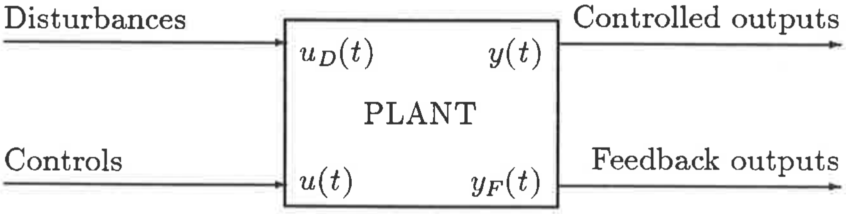

Figure 2.1: Plant.

The $l \times 1$ vector of inputs $u(t)$ represents the control inputs, i.e., signals which are applied to the plant in order to achieve the control system objectives. The $s \times 1$ vector of inputs $u_D(t)$ represents (unmeasurable) deterministic and/or stochastic disturbances affecting the plant, i.e., signals over which one has no influence [1]. The

[1] For the discussion of elimination of the measurable disturbances, and for further references, see [59, chapter 6]).

separation of output variables into groups of the outputs which are to be controlled, $y(t)$ ($m \times 1$ vector), and the feedback outputs, $y_F(t)$ ($f \times 1$ vector), provides a means for

- formalization of the control of outputs which are not fed back;
- inclusion in the control law of system outputs which are not strictly necessary to achieve the control objective by the selected control method.

The above process structure has been postulated in linear system synthesis theory [60,61,62,63,64] but, to the author's knowledge, has not been considered explicitly in adaptive control theory, although its applicability was suggested in [9, pp. 232–234 and p. 458]. Furthermore, we shall classify system outputs according to the feedback configuration in the following two ways:

Feedback configuration FI: the feedback and controlled outputs vectors are identical, i.e., $y_F(t) \equiv y(t)$;

Feedback configuration FD: some, or perhaps none, of the feedback outputs are the same as controlled outputs, i.e., $y_F(t) \neq y(t)$, but the number of the feedback outputs is not smaller than the number of the controlled outputs, i.e., $\dim y_F(t) \geq \dim y(t)$.

The self-tuning control strategies have been developed for systems having the feedback configuration FI. The case FI is considered in section 3.1, and chapters 4 and 5. The main reason for considering systems having the feedback configuration FD is to develop for self-tuning control an approach which accommodates feedback outputs others than only the controlled ones. The feedback configuration FD implies that there are some *additional system outputs* which can be utilized for feedback. The case FD is considered in sections 3.2, 3.3, and in chapter 6.

The system models, which are used in the following chapters, are introduced in subsections 2.1.1 and 2.1.2.

2.1.1 System state-space models.

Consider a *linear, time-invariant, finite-dimensional, discrete-time* system described by a state-space model

$$\begin{bmatrix} qx(t) \\ \overline{y}(t) \end{bmatrix} = S \begin{bmatrix} x(t) \\ \overline{u}(t) \end{bmatrix}, \quad t \geq 0 \tag{2.1}$$

with initial state $x(0)$, where $x(t)$ is the state vector, $\overline{y}(t)$ and $\overline{u}(t)$ are system output and input vectors, respectively. The forward shift operator q is defined in Appendix A, S is a real matrix, and $t = 0, 1, 2 \ldots$. Special cases of the above general form of the state-space model are introduced below. The input variables included in the input vector, $\overline{u}(t)$, depend on the type of disturbances affecting the system. The output variables included in the output vector, $\overline{y}(t)$, depend on the system feedback configuration employed.

State-space model of a system with deterministic disturbance.

Assume that the plant is subject to a deterministic disturbance only. The generator of the deterministic disturbance can be considered as external to the plant model. Alternatively, the model of the generator of the deterministic disturbance can be incorporated in an extended plant model, leading to the system [2] structure shown in fig. 2.2. The latter approach will be assumed here for the analysis of deterministic systems. Note that there is no disturbance input to the system shown in fig. 2.2.

The deterministic disturbances can be thought of as the solution to a set of linear differential equations, with nonzero initial conditions, describing the disturbance generator [59, p. 127]. In order to incorporate the model of the generator of deterministic disturbances in the system model, the plant state variables are augmented by the state variables of the state-space representation of the disturbance generator. The vector of the plant initial conditions is augmented by the set of initial conditions appropriate to the disturbances. This approach to the modeling of the deterministic

[2] The model of a system with the incorporated model of the disturbance generator is sometimes called exosystem [60].

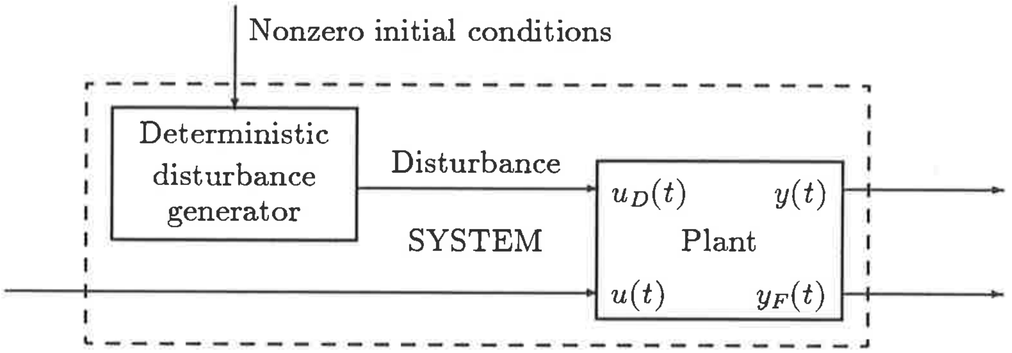
Nonzero initial conditions
Deterministic disturbance generator
Disturbance
SYSTEM
$u_D(t)$
$y(t)$
Plant
$u(t)$
$y_F(t)$

Figure 2.2: Plant with the model of a generator of deterministic disturbance incorporated in the system model.

disturbances allows us to set $\overline{u}(t) = u(t)$ in the state-space system representation 2.1 since there is no disturbance input to the system. In relation to the feedback configuration, the following two cases are considered.

State-space model of a deterministic system having the feedback configuration FI.

In this case $y_F(t) \equiv y(t)$; hence, we set $\overline{y}(t) = y(t)$. The state-space system representation is thus given by

$$\begin{bmatrix} qx(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A_s & B_s \\ C & D \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}, \qquad t \geq 0 \tag{2.2}$$

with initial state $x(0)$. It is assumed that $D = 0$.

State-space model of a deterministic system having the feedback configuration FD.

In this case only some, or perhaps none, of the feedback variables are the same as the controlled ones. Therefore, we set $\overline{y}(t) = [y(t)'\ y_F(t)']'$ ($'$ denotes transposition). The state-space system representation is thus given by

$$\begin{bmatrix} qx(t) \\ y(t) \\ y_F(t) \end{bmatrix} = \begin{bmatrix} A_s & B_s \\ C & D \\ C_F & D_F \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}, \qquad t \geq 0 \tag{2.3}$$

with initial state $x(0)$. It is assumed that $D = 0$ and $D_F = 0$.

State-space model of a system with stochastic disturbance.

Consider a plant which is subject to stochastic disturbance only [3]. It is assumed that the stochastic disturbance affecting the plant is generated by a linear, time-invariant, finite-dimensional, casual, discrete-time dynamic system (stochastic disturbance filter) driven by a sequence $\{\omega(t)\}$. Furthermore, $\{\omega(t)\}$ is a zero-mean, covariance-stationary, vector white sequence [65, chapter 2], i.e.,

$$\begin{aligned} E\{\omega(t)\} &= 0, && (2.4) \\ E\{\omega(t)\omega(s)'\} &= \Sigma\delta(t-s), && (2.5) \end{aligned}$$

where the covariance matrix Σ is a positive-definite, real symmetric matrix, and $\delta(t-s)$ is the Kronecker delta. Moreover, it is assumed that

$$\dim \omega(t) = \dim y(t) = m. \tag{2.6}$$

The above assumptions for the stochastic disturbance are commonly used in self-tuning control theory [31,66,67,9].

A similar approach to that of modeling the deterministic disturbances is considered for stochastic disturbances. The model of the stochastic disturbance filter is incorporated in the plant model, leading to the stochastic system structure shown in fig. 2.3.

In the state-space representation 2.1 of a system which incorporates the model of the plant and stochastic disturbance filter we set $\overline{u}(t) = [\omega(t)'\ u(t)']'$. In relation to the feedback configuration, the following two cases are considered.

[3] A more general case of a plant which is subject to both stochastic and deterministic disturbances is discussed on p. 31.

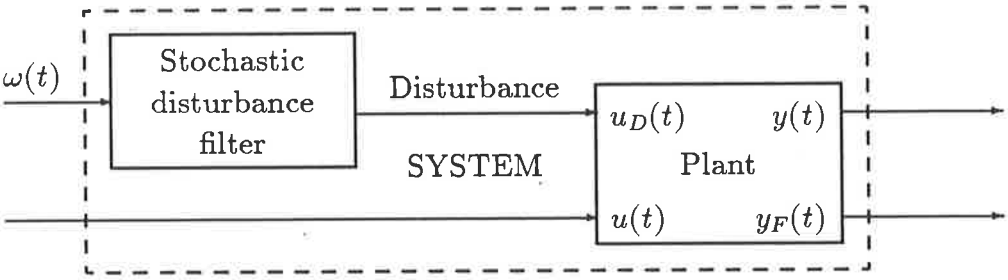

Figure 2.3: Plant with the model of a stochastic disturbance filter incorporated in the system model.

State-space model of a stochastic system having the feedback configuration FI.

In this case $y_F(t) \equiv y(t)$; hence, we set $\overline{y}(t) = y(t)$. The state-space model 2.1 is thus given by

$$\begin{bmatrix} qx(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A_s & K & B_s \\ C & C_s & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \omega(t) \\ u(t) \end{bmatrix}, \qquad t \geq 0 \qquad (2.7)$$

with initial state $x(0)$. It is assumed that the matrix C_s is nonsingular and, without loss of generality, $C_s = I_m$ (I_m is an $m \times m$ identity matrix) [68,56,69]. The state-space model of the system given by eqn. 2.7 is shown in fig. 2.4.

State-space model of a stochastic system having the feedback configuration FD.

In this case only some, or perhaps none, of the feedback variables are the same as the controlled ones. Therefore, we set $\overline{y}(t) = [y(t)' \; y_F(t)']'$. The state-space model of the system is thus given by

$$\begin{bmatrix} qx(t) \\ y(t) \\ y_F(t) \end{bmatrix} = \begin{bmatrix} A_s & K & B_s \\ C & C_s & 0 \\ C_F & C_S & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \omega(t) \\ u(t) \end{bmatrix}, \qquad t \geq 0 \qquad (2.8)$$

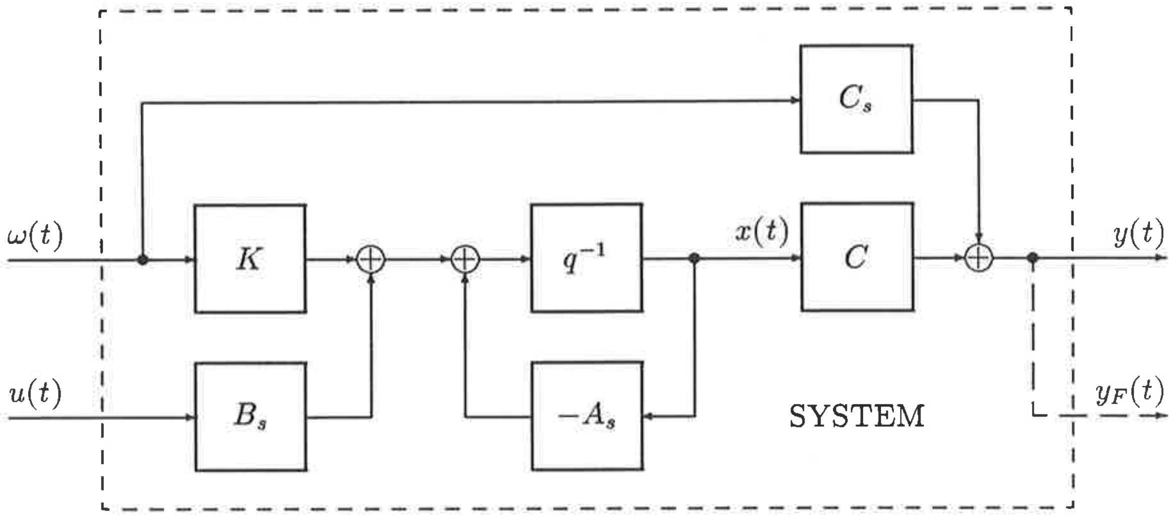

Figure 2.4: State-space model of a stochastic system having the feedback configuration FI.

with initial state $x(0)$. Again it is assumed that the matrix C_s is nonsingular and, without loss of generality, $C_s = I_m$. Furthermore, we assume that the number of the feedback variables is greater than the number of independent noise sources [4], i.e., $\dim y_F(t) > \dim \omega(t)$. Hence, in view of eqn. 2.6, the number of the feedback variables is greater than the number of the outputs which are to be controlled, i.e., $f = \dim y_F(t) > \dim y(t) = m$.

The state-space model of the system given by eqn. 2.8 is shown in fig. 2.5.

Comments on the state-space models of systems with stochastic disturbances.

The state-space model of a stochastic system having the feedback configuration FI (eqn. 2.7) is closely associated with the Kalman filter theory and is known as the (steady-state) *innovations model* [70,9]. The model 2.7 forms a basis for the de-

[4] This assumption is required by the approach to optimal prediction and control developed in sections 3.3 and 6.2 (see Comment 3.5, p. 105).

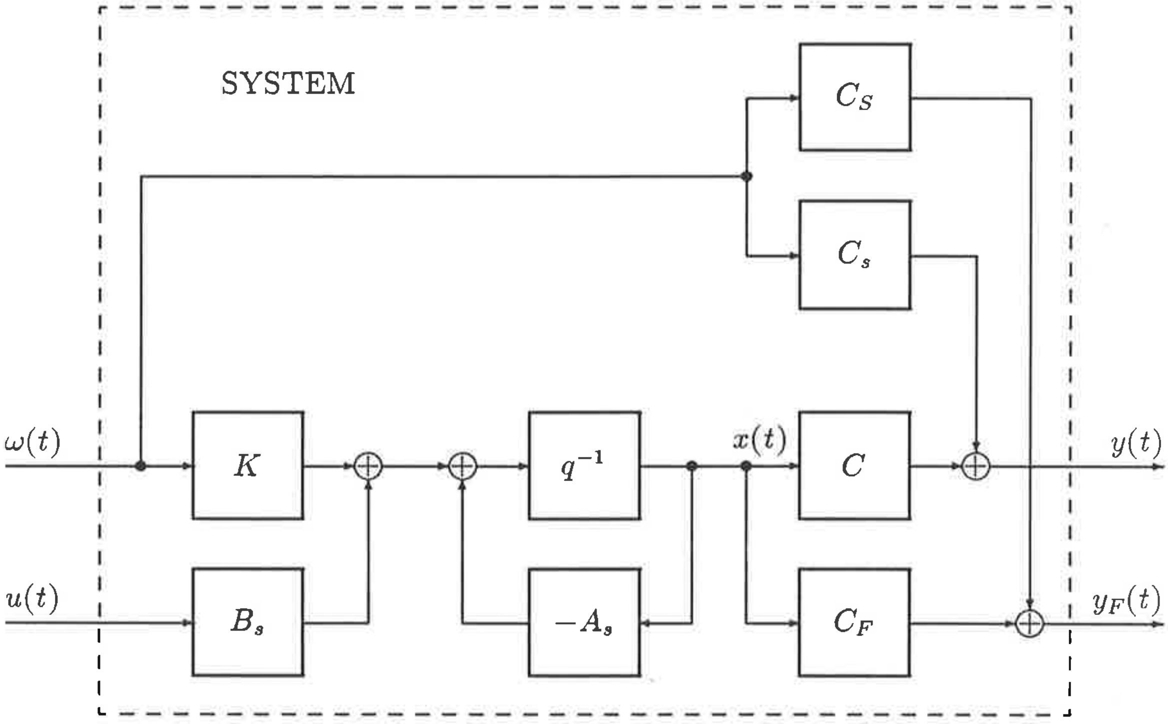

Figure 2.5: State-space model of a stochastic system having the feedback configuration FD.

velopment [68,71,72,73,74,69] and analysis [36] of many adaptive stochastic control strategies.

The exclusive use of systems having the feedback configuration FI in stochastic self-tuning control can be perhaps explained by the fact that many strategies for multivariable systems were developed as extensions of the schemes introduced for scalar systems. A typical example is the multivariable minimum variance regulator [31]. Any scalar system implies the feedback configuration FI.

On the other hand, the system model 2.8 is introduced here to describe a more general case than does model 2.7. In particular, we do not require the output variables, which are to be controlled, to be used in the calculation of the control signal. However, the same control objective as for a system having the feedback configuration FI can be formulated for the case FD.

2.1.2 System difference operator models.

An alternative to the state-space model, based on a set of first-order difference equations, is a high-order difference equation known as the general difference operator representation [75,76,9,77]. In this subsection, the difference operator representations, corresponding to the state-space models presented in subsection 2.1.1, are introduced. The difference operator representations are used for the development of the self-tuning control strategies in the following chapters.

There are two particular forms of the general difference operator representation which are employed in the development of self-tuning control strategies. Firstly the left difference operator (LDO) representation is given by

$$\begin{bmatrix} N_L(q) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \overline{u}(t) \\ \overline{y}(t) \end{bmatrix} = \begin{bmatrix} D_L(q) \\ I \end{bmatrix} x_L(t), \qquad t \geq 0 \tag{2.9}$$

with appropriate initial conditions on the partial state sequence [5] $\{x_L(t)\}$. Secondly the right difference operator (RDO) representation, is described by

$$\begin{bmatrix} \overline{u}(t) \\ \overline{y}(t) \end{bmatrix} = \begin{bmatrix} D_R(q) \\ N_R(q) \end{bmatrix} x_R(t), \qquad t \geq 0 \tag{2.10}$$

with appropriate initial conditions on $\{x_R(t)\}$. In these representations $N_L(q)$, $D_L(q)$, $D_R(q)$, and $N_R(q)$ are polynomial matrices in the forward shift operator q, $D_L(q)$ and $D_R(q)$ are assumed to be nonsingular.

In the implementation of self-tuning controllers, the model which is to be estimated is either that of a system which is to be controlled (indirect strategy), or that of a control law which is to be implemented (direct strategy). The model, the parameters of which are estimated by a recursive parameter estimation algorithm embedded in the self-tuning control strategy, must be in the LDO representation form to be suitable for estimation from the model input-output data [6] [39]. Therefore the LDO

[5] See Appendix A for definition of a sequence.

[6] The parameter estimation algorithms are usually designed for the LDO model. The only exception known to the author is the algorithm proposed in [78] for the estimation of parameters of a special form of the RDO representation having a diagonal matrix $D_R(q)$.

representation of the system which is to be controlled is employed in the development of indirect strategies. The development of the direct strategies evolves usually from the system RDO representation and leads to the LDO representation of the control law, the parameters of which are estimated in the self-tuning controller [39].

In this thesis the LDO and RDO representations of the system are used for the development of the indirect and direct self tuners, respectively.

The indirect self-tuning strategies are proposed for systems having the feedback configuration FI (see chapters 4 and 5). Consequently, the LDO representation is used to describe systems having the feedback configuration FI.

The direct self-tuning strategies are proposed for systems having the feedback configuration FD (see chapter 6). Therefore, the RDO representation is used to describe systems having the feedback configuration FD. On the other hand, the indirect approach to self-tuning control of such systems would involve complex computations in order to determine controller parameters from the estimated system parameters.

Difference operator representations of a deterministic system.

The transformations between state-space and difference operator representations of deterministic systems are described in [75, section 4.3] [9, chapter 2] [79, chapter 5].

The LDO and RDO representations of deterministic systems are defined below.

Left difference operator (LDO) representation of a deterministic system having the feedback configuration FI.

Consider the state-space model 2.2 with $D = 0$. Define the left coprime polynomial matrices $\hat{A}(q)$ and $\hat{B}_u(q)$ satisfying the following polynomial equation

$$\begin{bmatrix} \hat{A}(q) & \hat{B}_u(q) \end{bmatrix} \begin{bmatrix} -C \\ qI - A_s \end{bmatrix} = 0. \tag{2.11}$$

It is assumed that the system (a pair (C, A_s)) is observable. Then matrices C and $qI - A_s$ are right coprime [80, Theorem 6.2-6, p. 366], and the determinant of $\hat{A}(q)$ satisfies

$$\hat{d}(q) \overset{\text{def}}{=} \det(qI - A_s) = c \det \hat{A}(q), \tag{2.12}$$

where c is a constant. Define polynomial matrix $\hat{B}(q)$ as follows

$$\hat{B}(q) = \hat{B}_u(q) B_s. \tag{2.13}$$

The LDO representation follows from eqns. 2.2 and 2.9 as

$$\begin{bmatrix} \hat{B}(q) & 0 \\ 0 & I_m \end{bmatrix} \begin{bmatrix} u(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} \hat{A}(q) \\ I_m \end{bmatrix} x_L(t), \qquad t \geq 0 \tag{2.14}$$

with appropriate initial conditions [7] on $\{x_L(t)\}$.

Assuming zero initial conditions and taking z-transform, the corresponding transfer matrix can be found from eqns. 2.2 and 2.14 as

$$H_{y,u}(z) = C(zI - A_s)^{-1} B_s = \hat{A}^{-1}(z)\hat{B}(z).$$

Since $H_{y,u}(z)$ is strictly proper [8], the degree of the polynomial matrix [9] $\hat{B}(q)$ is less than the degree of $\hat{A}(q)$, i.e., deg $\hat{B}(q) <$ deg $\hat{A}(q) = n$ [80, p. 383]. Denote

$$\begin{aligned} A(p) &= p^n \hat{A}(q), & (2.15) \\ B(p) &= p^n \hat{B}(q), & (2.16) \end{aligned}$$

where p is the backward shift operator defined in Appendix A. Furthermore, assume that the polynomial matrix $\hat{A}(q)$ is monic [10], i.e., $\hat{A}(q) = A_0 q^n + A_1 q^{n-1} + \cdots + A_n$, where $A_0 = I_m$. The LDO representation given by eqn. 2.14 is now written in the ***Deterministic AutoRegressive Moving-Average*** (DARMA) form as [9, chapter 2]

$$A(p)y(t) = B(p)u(t), \qquad t \geq n \tag{2.17}$$

[7] The problem of transforming a set of initial conditions associated with the state-space model to initial conditions for the difference operator model in considered for instance in [77,9].

[8] A rational transfer matrix $H_{y,u}(z)$ is said to be ***strictly proper*** if $\lim_{z\to\infty} H_{y,u}(z) = 0$ [77].

[9] See Appendix B for definition of the degree of a polynomial matrix.

[10] If this is not the case, then the LDO representation can be transformed so that $\hat{A}(q)$ is monic [39,81]. The resulting matrices $\hat{A}(q)$ and $\hat{B}(q)$ are not, in general, left coprime.

with appropriate initial conditions on $\{y(t)\}$ (note from eqn. 2.14 that $x_L(t) = y(t)$).

If the system is subject to deterministic disturbance, the generator model of which is incorporated in the state-space description (see discussion on p. 19), the polynomial matrices $A(p)$ and $B(p)$ of the resulting DARMA model have a common left factor (polynomial matrix) corresponding to the disturbance description [9, chapter 2].

Right difference operator (RDO) representation of a deterministic system having the feedback configuration FD.

Consider the state-space model 2.3 with $D = 0$ and $D_F = 0$. Define the right coprime polynomial matrices $\hat{D}(q)$ and $\hat{N}_u(q)$ satisfying the following polynomial equation

$$\begin{bmatrix} -B_s & qI - A_s \end{bmatrix} \begin{bmatrix} \hat{D}(q) \\ \hat{N}_u(q) \end{bmatrix} = 0. \tag{2.18}$$

It is assumed that the system (a pair (A_s, B_s)) is controllable. Then matrices $qI - A_s$ and B_s are left coprime [80, Theorem 6.2.-6, p. 366], and the determinant of $\hat{D}(q)$ satisfies

$$\hat{d}(q) \overset{\text{def}}{=} \det(qI - A_s) = c \det \hat{D}(q), \tag{2.19}$$

where c is a constant. Define polynomial matrices $\hat{N}(q)$ and $\hat{N}_F(q)$ as follows

$$\hat{N}(q) = C\hat{N}_u(q), \tag{2.20}$$

$$\hat{N}_F(q) = C_F\hat{N}_u(q). \tag{2.21}$$

The RDO representation follows from eqns. 2.3 and 2.10 as

$$\begin{bmatrix} u(t) \\ y(t) \\ y_F(t) \end{bmatrix} = \begin{bmatrix} \hat{D}(q) \\ \hat{N}(q) \\ \hat{N}_F(q) \end{bmatrix} x_R(t), \qquad t \geq 0 \tag{2.22}$$

with appropriate initial conditions on $\{x_R(t)\}$.

Difference operator representations of a stochastic system.

The transformations between the state-space and LDO representations of stochastic systems are considered for instance in [74,69,9]. The LDO and RDO representations are defined below.

Left difference operator (LDO) representation of a stochastic system having the feedback configuration FI.

Consider the state-space model 2.7 with $C_s = I_m$. Let us determine the left coprime polynomial matrices $\hat{A}(q)$ and $\hat{B}_u(q)$ satisfying eqn. 2.11. It is assumed that the pair $(C,\ A_s)$ is observable. Furthermore, define polynomial matrix $\hat{C}(q)$ as follows

$$\hat{C}(q) = \hat{A}(q) + \hat{B}_u(q)K. \tag{2.23}$$

The LDO representation follows from eqns. 2.7 and 2.9 with the matrix $\hat{B}(q)$ defined by eqn. 2.13, and is given by

$$\begin{bmatrix} \hat{C}(q) & \hat{B}(q) & 0 \\ 0 & 0 & I_m \end{bmatrix} \begin{bmatrix} \omega(t) \\ u(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} \hat{A}(q) \\ I_m \end{bmatrix} x_L(t), \qquad t \geq 0 \tag{2.24}$$

with appropriate initial conditions on $\{x_L(t)\}$.

From eqns. 2.11 and 2.13, one has $\deg \hat{B}(q) \leq \deg \hat{B}_u(q) < \deg \hat{A}(q) = n$. Then from eqn. 2.23, $\deg \hat{C}(q) = n$. As for the deterministic system, it is assumed that $\hat{A}(q)$ is monic, and it follows from eqn. 2.23 that $\hat{C}(q)$ is monic as well. Denote

$$C(p) = p^n \hat{C}(q). \tag{2.25}$$

Using eqns. 2.15, 2.16 and 2.25, the LDO representation given by eqn. 2.24 is written in the *AutoRegressive Moving-Average with auXiliary* input (ARMAX) form [9, chapter 7] as

$$A(p)y(t) = \begin{bmatrix} B(p) & C(p) \end{bmatrix} \begin{bmatrix} u(t) \\ \omega(t) \end{bmatrix}, \qquad t \geq n \tag{2.26}$$

with appropriate initial conditions on $\{y(t)\}$.

Right difference operator (RDO) representation of a stochastic system having the feedback configuration FD.

Consider the state-space model 2.8 with $C_s = I_m$. Let us determine the right coprime polynomial matrices $\hat{D}(q)$ and $\hat{N}_u(q)$ satisfying eqn. 2.18. Define the right coprime polynomial matrices $\hat{D}_\omega(q)$ and $\hat{C}_R(q)$ satisfying the following polynomial equation

$$\begin{bmatrix} -K & qI - A_s \end{bmatrix} \begin{bmatrix} \hat{D}_\omega(q) \\ \hat{C}_R(q) \end{bmatrix} = 0. \tag{2.27}$$

It is assumed that the pair (A_s, B_s) is controllable and the matrices $qI - A_s$ and K are left coprime. Then the determinants of $\hat{D}(q)$ and $\hat{D}_\omega(q)$ satisfy (cf. eqn. 2.19)

$$\hat{d}(q) \overset{\text{def}}{=} \det(qI - A_s) = c \det \hat{D}(q) = \overline{c} \det \hat{D}_\omega(q), \tag{2.28}$$

where $\overline{c}$ is a constant.

Furthermore, define polynomial matrices $\hat{c}_\omega(q)$ and $\hat{C}_\omega(q)$ as follows

$$\begin{aligned} \hat{c}_\omega(q) &= \hat{D}_\omega(q) + C\hat{C}_R(q), & (2.29) \\ \hat{C}_\omega(q) &= C_S\hat{D}_\omega(q) + C_F\hat{C}_R(q). & (2.30) \end{aligned}$$

Then using eqns. 2.20, 2.21, 2.29, and 2.30, the RDO plant representation follows from eqns. 2.8 and 2.10

$$\begin{bmatrix} u(t) \\ \omega(t) \\ y(t) \\ y_F(t) \end{bmatrix} = \begin{bmatrix} \hat{D}(q) & 0 \\ 0 & \hat{D}_\omega(q) \\ \hat{N}(q) & \hat{c}_\omega(q) \\ \hat{N}_F(q) & \hat{C}_\omega(q) \end{bmatrix} \begin{bmatrix} x_R(t) \\ x_{R\omega}(t) \end{bmatrix}, \qquad t \geq 0 \tag{2.31}$$

with appropriate initial conditions on $\{[x_R(t)' \; x_{R\omega}(t)']'\}$.

Although the RDO representation has been used commonly to derive (direct) adaptive schemes for deterministic systems [39], it has not been employed for the modeling of stochastic systems. The stochastic plants were restricted to the feedback configuration FI and modeled by the ARMAX model.

Discussion of the approaches to modeling of plants subject to deterministic and stochastic disturbances.

If a plant is subject to both stochastic and deterministic disturbances, then incorporation of the model of the generator of deterministic disturbance in the system description leads to the ARMAX model with roots of the determinant of the noise polynomial $C(z^{-1})$ lying *on the unit circle* [9, chapter 7] [59, chapter 6]. This, in turn, implies that the predictors and predictive controllers developed for such a system model are either optimal but time-varying, or time-invariant but suboptimal [9, chapter 7, 10] [59, chapter 12].

It is often assumed that roots of the determinant of the noise polynomial $C(z^{-1})$ lie *strictly inside the unit circle* [11]. Such an assumption permits derivation of the optimal time-invariant predictors and controllers [83], but excludes the possibility of incorporating the deterministic disturbance generator model into the system model [12]. However, in many practical situations, deterministic disturbances (such as load disturbances, nonzero mean of the noise, an offset due to the local linearization of a nonlinear process) can be modeled by augmenting the ARMAX model with a constant (or slowly time-varying) offset term b, as follows [84,85]

$$A(p)y(t) = \begin{bmatrix} B(p) & C(p) & I_m \end{bmatrix} \begin{bmatrix} u(t) \\ \omega(t) \\ b \end{bmatrix}. \tag{2.32}$$

The above approach corresponds to the model of the generator of deterministic disturbance which is external to the model of the plant and stochastic disturbance filter. It makes the assumption about roots of the noise polynomial $C(z^{-1})$ lying strictly

[11] In the view of the Spectral Factorization Theorem, it can be assumed without loss of generality that roots of the determinant of $C(z^{-1})$ lie *inside or on the unit circle* [82, Theorem 10.1, p. 47].

[12] The corresponding LDO representation is sometimes called the Controllable AutoRegressive Moving Average (CARMA) model [74].

inside the unit circle a reasonable one [9, p. 264].

So far the underlying assumption is that the stochastic disturbances are (covariance) stationary (this assumption will be preserved in the following chapters). It is, however, possible to generalize this assumption to random processes with stationary *increments* [13] [86, chapter 3] [87, chapter 8]. This approach led to the introduction of the Controllable Autoregressive Integrated Moving Average (CARIMA) model

$$A(p)y(t) = \begin{bmatrix} B(p) & I_m \end{bmatrix} \begin{bmatrix} u(t) \\ s(t) \end{bmatrix}, \tag{2.33}$$

$$(1-p)s(t) = C(p)\omega(t), \tag{2.34}$$

where $s(t)$ is a $m \times 1$ vector. If $\omega(t)$ is a zero mean white (gaussian) noise then $s(t)$ is a Brownian motion process; if $\omega(t)$ is zero except at particular sample instants then $s(t)$ consists of a series of steps. The CARIMA model was employed in both identification [88] and adaptive control [89,90,91,29,92]. A further extension of the above model was presented in [93] where a plant affected by both Brownian and generalized Poisson processes was considered.

The argument given in favour of a process with stationary increments as a model of stochastic disturbances in industrial plants is that step discontinuities in state trajectories often occur due to component failures and switching of elements, e.g., load disturbances consisting of random steps at random times. On the other hand, for a plant operating in its linear regime the effective disturbance resulting from a large number of small independent sources is approximately gaussian [14] and may be modeled more appropriately by the ARMAX model with gaussian input noise. A model of a plant with stationary random process is appropriate, for example, if the main source of stochastic disturbance is sensor noise. Examples of industrial applications of self-tuning control based on the ARMAX model and outperforming the corresponding strategies based on the CARIMA model are given, for instance, in [94].

[13] Every stationary process is also a process with stationary increments.

[14] This observation is based on the Central Limit Theorem [87].

2.2 Self-tuning closed-loop system structure.

In this section:

- the closed-loop system structure is described;
- the requirements of an admissible control law are presented;
- the requirements on the output measurement configuration for self-tuning control are introduced.

We shall consider discrete-time controllers resulting from minimization of cost functions involving, in general, control inputs $u(t)$ and controlled outputs $y(t)$. The controller is required to be linear, finite-dimensional, causal, and stable. It is applied to the system as shown in fig. 2.6, where $r(t)$ is a reference signal and $y_F(t)$ represents outputs which are available for feedback (this covers both cases FI and FD).

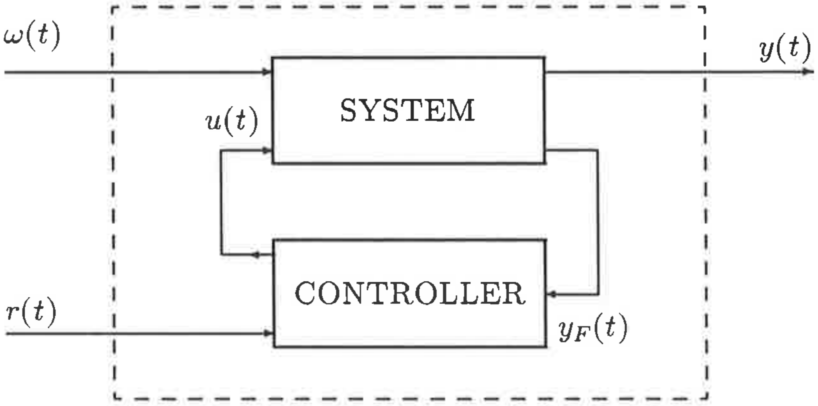

Figure 2.6: Closed-loop system configuration.

The LDO model of the controller is given in the form of the following *general linear control law* (GLCL)

$$V(p)u(t) = \begin{bmatrix} M(p) & -K(p) \end{bmatrix} \begin{bmatrix} r(t+k) \\ y_F(t) \end{bmatrix}, \tag{2.35}$$

where $V(p)$, $M(p)$ and $K(p)$ are polynomial matrices in the backward shift operator p. (It is often assumed in predictive control that future values of the reference signal, i.e., $r(t+i)$ for $i = 1, \ldots, k$, are known or can be predicted at sample instant t [27].)

The controller given by eqn. 2.35 is causal if the coefficient matrix V_0 of the polynomial matrix $V(p) = V_0 + V_1 p + \cdots$ is nonsingular [77, Corollary, p. 343]. This will be guaranteed by appropriate use of a characterization of the multivariable system delay structure in the design of controllers in chapters 4 and 5. The characterizations of the delay structure are developed in section 2.3. Note from eqn. 2.35 that the coefficient matrix V_0 is associated with the value of the control signal, $u(t)$, at time t. Therefore, the nonsingularity of V_0 permits the unique determination of $u(t)$ from eqn. 2.35.

Let us now discuss aspects of the output measurement configuration of the closed-loop system.

The measurement of outputs $y_F(t)$ is required for the nonadaptive as well as self-tuning control involving the control law 2.35.

A self-tuning controller involves recursive estimation of the unknown parameters Θ of the controlled system or of the control law. The update equation for the parameter estimate $\widehat{\Theta}(t)$ of a recursive estimator employed in the self-tuning controller involves a regression vector and an equation error [15] (see Appendix E). Knowledge of the controlled outputs $y(t)$ is required for the regression vector or for the calculation of the equation error. Therefore, the controlled outputs $y(t)$ must be measured for the purpose of parameter estimation. The feedback outputs $y_F(t)$ are available for the estimator since they are measured for the implementation of the control law.

A general closed-loop system configuration involving a self-tuning controller and a system with two groups of outputs ($y(t)$ and $y_F(t)$) is shown in fig. 2.7 (cf. fig. 1.1, p. 4). Note that in fig. 2.7 the measurement of $y(t)$ for the purpose of parameter estimation is shown explicitly. However, for self-tuning control of systems having the feedback configuration FI the requirement of measurement of $y(t)$ is automatically satisfied (since $y_F(t) \equiv y(t)$ is measured for the implementation of the control law). On the other hand, for systems having the feedback configuration FD those of the controlled outputs $y(t)$ which are not included in $y_F(t)$ must be measured for

[15] For certain recursive parameter estimation methods the term *output error* is more appropriate than *equation error* [9, p. 82].

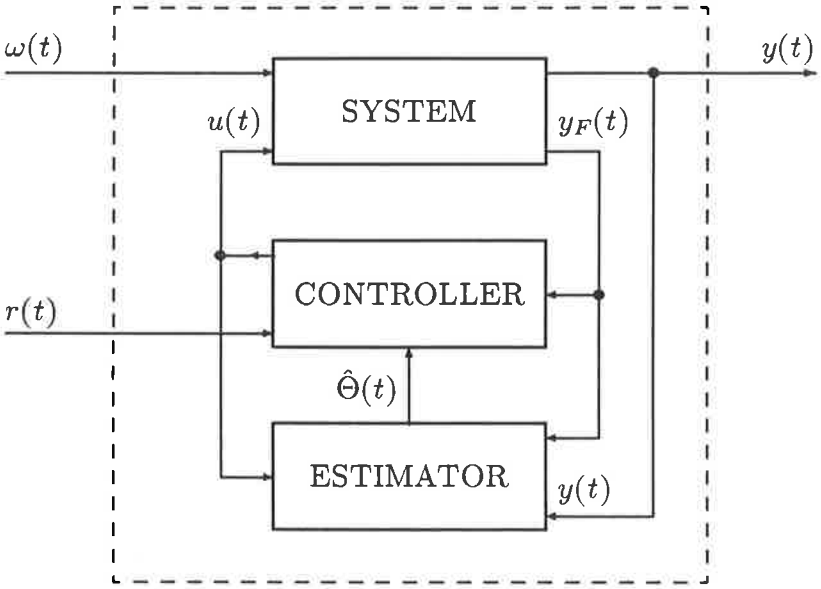

Figure 2.7: Self-tuning closed-loop system configuration.

the purpose of parameter estimation. The total number of sensors required for the implementation of the self-tuning controller is greater for the feedback configuration FD than for the feedback configuration FI.

2.3 System delay structure – interactor and nilpotent interactor matrices.

In this section:

- a brief summary is presented of various assumptions in self-tuning predictive control regarding prior knowledge of the characterizations of a multivariable system delay structure;
- new characterizations of a multivariable system delay structure, called the (left and right) nilpotent interactor matrices, are introduced;
- algorithms for the calculation of the nilpotent interactor matrices are developed.

The delay structure of a scalar system with integer delay of k sample intervals can be characterized by the polynomial z^k [9, p. 133]. The usual assumption in self-tuning minimum prediction error (MPE) control of scalar systems is that the process delay, or more generally the delay structure, is known (e.g., [54,56] [9, chapters 6 and 11], see also Assumptions 6.1 (ii) on page 200, and 6.4 (ii), p. 228).

One of the reasons for difficulties encountered in extension of adaptive schemes developed for scalar systems to the multivariable case is associated with required prior knowledge of the system delay structure. It is shown in [95] that the multivariable system delay structure is characterized by the *interactor* (polynomial) matrix introduced in [33]. Similarly as for the self-tuning predictive control of scalar systems, prior knowledge of the delay structure in the form of the interactor matrix is often assumed in multivariable self-tuning control [67]. However, the interactor matrix cannot be generally deduced from measurements of delays from system inputs to outputs. Furthermore, the assumption of its knowledge is, generally, tantamount to the complete knowledge of the system transfer matrix [96] and thus reduces applicability of self-tuning techniques. Hence, there is a significant motivation to relax or remove requirement of prior knowledge of the interactor matrix. This problem will be tackled using a new characterization of the delay structure in chapter 4; such a characterization is developed in subsection 2.3.3.

2.3.1 Various approaches to the multivariable system delay structure in self-tuning predictive control.

Let us define the (left) interactor matrix as in [33].

Definition 2.1 (Left interactor matrix) *Given any $m \times m$ proper transfer matrix* [16] *$H_{y,u}(z)$ having full (normal) rank* [17]*, there exists a $m \times m$ (unique) lower*

[16] A rational transfer matrix $H_{y,u}(z)$ is said to be *proper* if $\lim_{z\to\infty} H_{y,u}(z) = T$, where T is a finite (possibly zero) real matrix [77].

[17] A rational matrix (or a polynomial matrix) has *normal rank* r if r is the largest of the orders of the minors that are not identically zero [80, p. 373].

triangular polynomial matrix $\xi_L(z)$, called the left interactor matrix, of the form

$$\xi_L(z) = T_L(z) d_L(z), \tag{2.36}$$

where $T_L(z)$ is a lower triangular unimodular matrix with 1's on the main diagonal and with the off diagonal elements being zeros or divisible by z, and $d_L(z) = diag[z^{k_1} \ \dots \ z^{k_m}])$, and satisfying

$$\lim_{z \to \infty} \xi_L(z) H_{y,u}(z) = M_L, \tag{2.37}$$

where M_L is a nonsingular, real matrix.

Although the polynomial matrix $\xi_L(z)$ is known as the interactor matrix, we shall call it the *left* interactor matrix, because of its position as premultiplier in eqn. 2.37. For the definition of the left interactor matrix for nonsquare systems see [33].

Recently the *right* interactor matrix was defined in [35] in a completely analogous manner to the left interactor matrix.

Definition 2.2 (Right interactor matrix) *Given any $m \times m$, proper transfer matrix $H_{y,u}(z)$ having full (normal) rank, there exists a $m \times m$ (unique) upper triangular polynomial matrix $\xi_R(z)$, called the right interactor matrix, of the form*

$$\xi_R(z) = d_R(z) T_R(z), \tag{2.38}$$

where $T_R(z)$ is an upper triangular unimodular matrix with 1's on the main diagonal and with the off diagonal elements being zeros or divisible by z, and $d_R(z) = diag[z^{k_1} \ \dots \ z^{k_m}])$, and satisfying

$$\lim_{z \to \infty} H_{y,u}(z) \xi_R(z) = M_R, \tag{2.39}$$

where M_R is a nonsingular, real matrix.

The interactor matrix contains the structure of the system transfer matrix at infinity and in particular the system zeros at infinity which reduce to the delay for scalar systems [97,77,67].

Comment 2.1 *The triangular factor of the interactor matrix arises if certain linear dependencies occur during successive calculation of the interactor from the transfer matrix using the algorithm developed in [33]. Thus, it was argued that in practical situations the diagonal form of the interactor can often be assumed, since the exact linear dependencies are unlikely to occur and the near-linear dependencies can be ignored [34,95]. However, it is shown in [98] that it is necessary to consider the general triangular form of the interactor matrix for robust minimum prediction error (or model following) control.*

Discussion of assumptions regarding prior knowledge of the interactor matrices in self-tuning predictive control.

The following assumptions, regarding prior knowledge of the system delay structure in self-tuning predictive control, can be identified (representative rather than extensive references are given):

1. complete prior knowledge of the left interactor matrix is assumed, involving
   (a) diagonal interactor matrix with identical elements z^k [31,66,99],
   (b) diagonal interactor matrix with different elements z^{k_i} [32,100,101],
   (c) triangular (general) interactor matrix [34,95,102,103,97,39,36],
   (d) left nilpotent interactor matrix, introduced in subsection 2.3.2 (see section 5.1);
2. complete prior knowledge of the right interactor matrix is assumed [35];
3. incomplete prior knowledge of the left interactor matrix is assumed: the diagonal factor $d_L(z)$ is known
   (a) and degrees of the nondiagonal elements of the interactor are known, with nondiagonal elements $T_L(z)$ estimated on-line [37],
   (b) with nondiagonal elements (possibly overparameterized) estimated on-line [104];

4. incomplete prior knowledge of the left interactor matrix is assumed: an upper bound on the degree of the interactor matrix is known [18]. This approach involves

   (a) introduction of a control signal weighting matrix to guarantee invertibility of the coefficient matrix V_0 of the polynomial matrix $V(p)$ in eqn. 2.35 (p. 33) [105],

   (b) long-range predictive cyclostationary control law for deterministic [38] and stochastic [36] systems,

   (c) long-range predictive receding horizon control law for a system described by the CARIMA model with $C(p) = I$ [92];

5. no prior knowledge of the interactor is required for indirect self-tuning control involving:

   (a) the on-line calculation of the *left* interactor matrix from the estimates of the polynomials in the DARMA model [39],

   (b) the on-line calculation of the *right nilpotent* interactor matrix (introduced in subsection 2.3.3) from the estimates of the polynomials in the DARMA model (see chapter 4).

Approach 1 is associated with direct deterministic and stochastic self-tuning control, approaches 2 and 3 with direct deterministic self-tuning control, approach 4 with direct and indirect stochastic self-tuning control, and approach 5 with indirect deterministic self-tuning control.

The attempted classification includes also some other solutions, e.g., in [106] a fixed precompensator is used to form a system with diagonal interactor matrix before applying adaptive controller (the partial prior knowledge of the Hermite normal form of the system transfer matrix is required which corresponds to partial knowledge of the interactor matrix).

[18] For a square system the degree of the interactor $\xi_L(z)$ equals the largest infinite zero order of the transfer matrix [97].

Approach 1 represents an early stage in the development of self-tuning control strategies for multivariable systems. In particular, strategies for systems with diagonal interactor matrix can be seen as direct generalizations of strategies developed for scalar systems [67]. If the interactor matrix is diagonal, then it can be determined from measurements of delays from system inputs to outputs (see [9, p. 134]). The approach based on the complete prior knowledge of the interactor matrix is considered in section 5.1. However, the delay structure can be characterized by a nonunique left nilpotent interactor matrix or by the unique interactor matrix $\xi_L(z)$.

It is shown in [35] that for systems for which the transfer matrix (with polynomial elements rather than rational function elements) is column- but not row-reduced, the right interactor matrix is diagonal, but the left is not. Therefore, for such systems approach 2 requires prior knowledge of less parameters ($\xi_R(z) = d_R(z)$) than approach 1 ($\xi_L(z) = T_L(z)d_L(z)$).

Approach 3 presents a significant relaxation of the prior system knowledge. Only the diagonal factor $d_L(z)$ of the left interactor matrix must be known. The real coefficients of polynomials in the lower triangular factor of the left interactor matrix are estimated on-line. For a general triangular interactor matrix, however, its diagonal factor cannot be determined from measurements of delays from inputs to outputs. Again this makes the choice of $d_L(z)$ difficult.

Approach 4 can be seen as an attempt to overcome disadvantages of previous approaches by considering various predictive control laws involving detuning or multistep optimization horizon. One would have to examine which control law is more meaningful for a particular application, the computational burden involved, etc.

As already mentioned, it is desirable to consider a general triangular interactor matrix in order to ensure robustness of controllers (see Comment 2.1, p. 38). On the other hand, assumption of prior (complete or incomplete) knowledge of the interactor matrix in self-tuning control is conflicting with lack of system knowledge. The above two reasons stimulated the search for computationally efficient algorithms, which would allow on-line determination of the interactor matrix (approach 5). This approach is considered in subsections 2.3.2 and 2.3.3, and in chapter 4.

The original algorithm developed in [33] for the calculation of the unique interactor matrix operates on the system transfer matrix (i.e., matrix of rational functions), and is not amenable to on-line, computer-based calculations. Furthermore, it is the LDO representation, in the form of DARMA or ARMAX models, which is estimated in indirect self-tuning control schemes. Thus it is desirable to develop algorithms operating on the LDO system representation. There is, however, only one algorithm (to the author's knowledge) which evaluates the left interactor matrix from the estimated DARMA model [39]. This algorithm involves division of polynomial matrices.

The lack of efficient algorithms for the calculation of the interactor matrix can be, perhaps, explained by the requirements which interactor matrix satisfies. These requirements are of two types: ensuring uniqueness of the interactor and ensuring certain structural features.

Clearly, the uniqueness of the interactor is not required for the controller design, and has been omitted in a number of self-tuning strategies. For example, in [104] the uniqueness of the interactor is lost by overparameterization of the estimator of $T_L(z)$. Furthermore, for nonsquare systems the unique interactor may introduce unstable poles into the closed-loop system (see example in the paper reproduced in Appendix C, or example 4.23 in [33]), hence a nonunique interactor has to be involved. In fact, it will be shown that a nonunique characterization of the delay structure introduces a degree of freedom in the design of controllers which can influence the closed-loop system performance (see Aside 5.1, p. 161, and example 5.1.1, p. 163).

The structural features of the interactor are given by eqns. 2.36 and 2.37 (or, eqns. 2.38 and 2.39). In the references cited above, the interactor matrix is incorporated in the controller design procedure which involves (i) the property given by eqn. 2.37 (or eqn. 2.39) to guarantee causality of the controller and to determine the control signal uniquely, and (ii) the fact that interactor matrix is stable [19] in order to guarantee closed-loop system stability. The latter property of the interactor matrix follows from its structure (see eqns. 2.36 and 2.38), which is triangular with diagonal

[19] A polynomial matrix is said to be *stable* if all roots of its determinant lie strictly inside the unit circle.

elements given by $d_L(z)$ (or $d_R(z)$); hence $\det \xi_L(z) = z^{k_l}$ ($\det \xi_R(z) = z^{k_r}$). The essential property of the interactor matrix in the design of predictive controllers is that the interactor is a stable polynomial matrix irrespective of its structure, which is determined by an algorithm used for its evaluation.

The above considerations concerning the application of the interactor matrix to controller design, led the author to develop new characterizations for the multivariable system delay structure (see subsections 2.3.2 and 2.3.3).

Remark 2.1 *Subsequently, the term* interactor *associated with the unique polynomial matrices $\xi_L(z)$ or $\xi_R(z)$ will be preserved for* any *stable polynomial matrices $\overline{K}_L(z)$ or $\overline{K}_R(z)$ satisfying*

$$\lim_{z\to\infty} \overline{K}_L(z)H_{y,u}(z) = M_L, \tag{2.40}$$

or

$$\lim_{z\to\infty} H_{y,u}(z)\overline{K}_R(z) = M_R, \tag{2.41}$$

where real matrices M_L and M_R have full rank.

This is consistent with the approach of [39], where *any* polynomial matrix having the property 2.37 (with M_L being the identity matrix) is called an identity *interactor* matrix for the system defined by $H_{y,u}(z)$. The properties of an interactor matrix described in Remark 2.1 are more general than those of $\xi_L(z)$ and $\xi_R(z)$. In particular, nothing is said about the structure of an interactor matrix. It is shown in the following chapters that the properties of an interactor matrix described in Remark 2.1 are sufficient for the design of multivariable controllers (see Comments 4.1 (p. 124), and 5.1 (p. 160)).

So far, the properties of an interactor matrix in the field of polynomial matrices were emphasised. However, to develop methods of evaluation of an interactor matrix which are amenable to computer-based calculations, a link between a polynomial matrix and its numerical representation must be established. For this purpose a polynomial matrix is represented by a block-matrix of polynomial coefficients (see Appendix B). Therefore, calculations involving polynomial matrices become operations on real block matrices. This approach avoids the complexity of dealing with polynomial matrices and was undertaken in optimal control and filtering problems [107], and recently in development of computational methods for multivariable pole placement design [79]. Furthermore, the numerical equivalence of the generic expression for the product of two polynomials introduced in [108] [79, chapter 3] (see Appendix B) will be useful in the implementation of algorithms.

2.3.2 The left nilpotent interactor matrix and an algorithm for its calculation.

In this subsection a new characterization of the system delay structure, called the left nilpotent interactor (LNI) matrix and corresponding to the left interactor matrix, is introduced.

The algorithm for calculation of the LNI matrix is covered by reference [109] which is reproduced in Appendix C. Some more details on the implementation of the algorithm, which are not considered in the original paper, are also given in Appendix C. A summary of the results follows (subscript "L" is introduced below to differentiate the left nilpotent interactor from the right one).

Definition 2.3 (Left nilpotent interactor (LNI) matrix) *If a $m \times l$ transfer matrix $H_{y,u}(z)$ has full (normal) rank and is proper, then any $m \times m$ polynomial matrix $K_L(z)$, having the properties*

$$\lim_{z\to\infty} K_L(z)H_{y,u}(z) = M_L, \tag{2.42}$$

where M_L is a full-rank, real matrix, and

$$\det K_L(z) = c_l z^{k_l}, \tag{2.43}$$

will be called the left nilpotent interactor (LNI) matrix for the system defined by $H_{y,u}(z)$ (c_l is a constant).

The term "nilpotent interactor" arises from the property 2.43 and from the algorithm presented in Appendix C which evaluates the interactor matrix as a product of first degree, nilpotent polynomial matrices [20]

$$K_L(z) = S_L^{(t)}(z)S_L^{(t-1)}(z)\dots S_L^{(1)}(z), \tag{2.44}$$

where

$$S_L^{(i)}(z) = U_L^{(i)}(z)Q_L^{(i)}. \tag{2.45}$$

The matrix $U_L^{(i)}(z)$ is a $m \times m$ row shifting polynomial matrix (r.s.p.m.) of order k_i

$$U_L^{(i)}(z) = U_0^{(i)}z + U_1^{(i)} = \begin{bmatrix} 0 & I_{r_i} \\ zI_{k_i} & 0 \end{bmatrix}, \qquad m = r_i + k_i, \tag{2.46}$$

and $Q_L^{(i)}$ is a nonsingular real matrix. The determinant of the r.s.p.m. of order k_i is $\det U_L^{(i)}(z) = (-1)^{k_i r_i} z^{k_i}$. Note that $\det S_L^{(i)}(z) = (-1)^{k_i r_i} z^{k_i} \det Q_L^{(i)} = c_i z^{k_i}$.

The algorithm presented in Appendix C evaluates matrix $K_L(z)$ from the numerator polynomial $N(z^{-1})$ of the right matrix fraction (RMF) description of the system transfer matrix $H_{y,u}(z) = N(z^{-1})D^{-1}(z^{-1})$. The RMF description results from the system transfer matrix $H_{y,u}(z)$, or from the system RDO representation 2.22 by assuming zero initial conditions and taking z-transform. Since the algorithm operates on the numerical representation of polynomial matrices, it is irrelevant whether the RMF or RDO representation is considered.

The macro XLNI, which is written in MATLAB commands and implements the algorithm for calculation of the LNI matrix with QR factorization, is given in Appendix D.

Example 2.1.

Consider the following transfer matrix (see [33, Example 3.33] [9, Example 5.2.5])

$$H_{y,u}(z) = \begin{bmatrix} \frac{1}{z+1} & \frac{1}{z+2} \\ \frac{1}{z+3} & \frac{1}{z+4} \end{bmatrix}. \tag{2.47}$$

[20] A first order polynomial matrix $V(z) = V_0 z + V_1$ is said to be nilpotent if $\det V(z) = cz^k$, where c is a constant. This is a generalization of the definition of a nilpotent real matrix W [80, p. 662], i.e., such a matrix that $\exists_r W^r = 0$, which implies that $\det V(z) = z^k$ where $V(z) = I_k z - W$.

The LNI matrix, calculated with QR factorization using MATLAB software package [110], is given by

$$K_L(z) = \begin{bmatrix} 0.5z^2 + 0.5z & -0.5z^2 + 0.5z \\ -0.5z^3 + 0.5z^2 & 0.5z^3 + 0.5z^2 \end{bmatrix}, \tag{2.48}$$

with its determinant $\det K_L(z) = z^4$ (for details see example in Appendix C, p. 285). For the comparison the unique interactor matrix 2.36 is given by

$$\xi_L(z) = \begin{bmatrix} z & 0 \\ -z^3 + 2z^2 & z^3 \end{bmatrix}. \tag{2.49}$$

2.3.3 The right nilpotent interactor matrix and an algorithm for its calculation.

In this subsection a new characterization of the system delay structure, called the right nilpotent interactor matrix and corresponding to the unique right interactor matrix [35], is introduced.

Definition 2.4 (Right nilpotent interactor (RNI) matrix) *If a $m \times l$ transfer matrix $H_{y,u}(z)$ has full (normal) rank and is proper, then any $l \times l$ polynomial matrix $K_R(z)$, having the properties*

$$\lim_{z \to \infty} H_{y,u}(z) K_R(z) = M_R, \tag{2.50}$$

where M_R is a full-rank, real matrix, and

$$\det K_R(z) = c_r z^{k_r}, \tag{2.51}$$

will be called the right nilpotent interactor (RNI) matrix for the system defined by $H_{y,u}(z)$ (c_r is a constant).

The term "nilpotent interactor" is used here (as for the LNI matrix) to stress the property given by eqn. 2.51 which is satisfied even for nonsquare systems, $H_{y,u}(z)$. This property is guaranteed by the algorithm for calculation of the RNI matrix. For the unique right interactor matrix $\xi_R(z)$ the property 2.51 may not be satisfied for nonsquare systems.

A proper transfer matrix $H_{y,u}(z)$ can be factorized into the left matrix fraction (LMF) description with monic denominator matrix

$$\begin{aligned} H_{y,u}(z) &= \hat{A}^{-1}(z)\hat{B}(z) = (z^n A(z^{-1}))^{-1}(z^n B(z^{-1})) = A^{-1}(z^{-1})B(z^{-1}), && (2.52)\\ \hat{A}(z) &= I_m z^n + A_1 z^{n-1} + \cdots + A_n = z^n A(z^{-1}), && (2.53)\\ \hat{B}(z) &= B_0 z^n + B_1 z^{n-1} + \cdots + B_n = z^n B(z^{-1}). && (2.54) \end{aligned}$$

Alternatively, the LMF representation with monic denominator matrix results from the DARMA model 2.17 (p. 27) by assuming zero initial conditions and taking z-transform. Since the algorithm presented below operates on numerical representation of polynomial matrices, it is irrelevant whether the LMF or DARMA representation is considered.

Using eqns. 2.50, 2.52, 2.53, and 2.54, one has

$$\begin{aligned} &\lim_{z\to\infty} H_{y,u}(z)K_R(z) = \lim_{z\to\infty} A^{-1}(z^{-1})B(z^{-1})K_R(z)\\ &= \lim_{z\to\infty} A^{-1}(z^{-1}) \lim_{z\to\infty} B(z^{-1})K_R(z) = \lim_{z\to\infty} B(z^{-1})K_R(z) = M_R. \qquad (2.55) \end{aligned}$$

Hence, the RNI matrix can be evaluated from the numerator polynomial matrix $B(z^{-1})$ of the LMF description. The algorithm presented below operates on the matrix of coefficients B of the polynomial matrix $B(z^{-1})$ (see eqn. B.1, Appendix B)

$$B(z^{-1}) = [B_0\ B_1\ \ldots\ B_n](Z_n \otimes I_l) = B(Z_n \otimes I_l). \qquad (2.56)$$

The algorithm developed for calculation of the LNI matrix in subsection 2.3.2 can be used for evaluation of the RNI matrix, as it is discussed in subsection 2.3.4. However, the RNI matrix can be calculated directly with a dual algorithm presented below. The original version of the algorithm adopted here for evaluation of the RNI matrix was proposed for the transformation of a polynomial matrix into a monic or comonic form in [81].

Let us define a column shift polynomial matrix (c.s.p.m.) as follows.

Definition 2.5 (Column shift polynomial matrix) *The $l \times l$ first degree polynomial matrix $U_R(z)$ will be called a column shift polynomial matrix of order k_i,* *where*

$$U_R(z) = U_0 z + U_1 = \begin{bmatrix} 0 & zI_{k_i} \\ I_{r_i} & 0 \end{bmatrix}, \qquad l = r_i + k_i. \tag{2.57}$$

The coefficient matrices U_0 and U_1 are defined through the matrix of coefficients

$$\mathcal{U}_R = \begin{bmatrix} U_0 & U_1 \end{bmatrix} = \begin{bmatrix} 0_{r_i} & I_l & 0_{k_i} \end{bmatrix}, \tag{2.58}$$

in which U_0, U_1 are of dimension $l \times l$, I_l is the $l \times l$ identity matrix, and 0_{r_i} is a r_i-column matrix of zeros. The determinant of the c.s.p.m. of order k_i is given by

$$\det U_R(z) = (-1)^{k_i r_i} z^{k_i}. \tag{2.59}$$

The algorithm is given in the form of the proof of the following result.

Theorem 2.1 *For a $m \times l$ full (normal) rank, proper transfer matrix $H_{y,u}(z)$ there exists a right nilpotent interactor matrix consisting of t factors:*

$$K_R(z) = S_R^{(1)}(z) \cdots S_R^{(t-1)}(z) S_R^{(t)}(z), \tag{2.60}$$

where

$$S_R^{(i)}(z) = Q_R^{(i)} U_R^{(i)}(z), \tag{2.61}$$

and $U_R^{(i)}(z)$ is a c.s.p.m. of order k_i and $Q_R^{(i)}$ is a nonsingular $l \times l$ real matrix.

Proof.

The proof consists of two parts: firstly, the factors of $K_R(z)$ in eqn. 2.60 are determined; secondly, it is shown that the number of factors is finite.

Part 1.

Set $i = 0$, $\mathcal{B}^{(0)}(z^{-1}) = B(z^{-1})$, and $K_R(z) = I_l$ to start the algorithm. Consider the i-th iteration in the evaluation of $K_R(z)$ based on eqn. 2.60.

step 1.

- If $r_i = \operatorname{rank} \mathcal{B}_0^{(i-1)} = min(m, l)$, the algorithm terminates and the right nilpotent interactor matrix is $K_R(z) = K_R^{(i-1)}(z)$, set $t = i - 1$;

- if $r_i < min(m, l)$, factorize $\mathcal{B}_0^{(i-1)}$ (using QR, LDU or SVD matrix factorization [111]) into

$$\mathcal{B}_0^{(i-1)} = \left[\begin{array}{cc} 0_{k_i} & \mathcal{B}_{0R}^{(i)} \end{array}\right](Q_R^{(i)})^{-1}, \quad \text{i.e.,} \quad \mathcal{B}_0^{(i-1)}Q_R^{(i)} = \left[\begin{array}{cc} 0_{k_i} & \mathcal{B}_{0R}^{(i)} \end{array}\right]. \tag{2.62}$$

where $Q_R^{(i)}$ is a $l \times l$ nonsingular (usually unitary) matrix, $k_i = l - r_i$, and 0_{k_i} is a k_i-column zero matrix.

step 2. Postmultiply $\mathcal{B}^{(i-1)}(z^{-1})$ by matrix $Q_R^{(i)}$

$$\overline{\mathcal{B}}^{(i)}(z^{-1}) = \mathcal{B}^{(i-1)}(z^{-1})Q_R^{(i)}, \tag{2.63}$$

(the coefficient matrix $\overline{\mathcal{B}}_0^{(i)}$ of $\overline{\mathcal{B}}^{(i)}(z^{-1})$ is now equal to the right hand side of eqn. 2.62).

step 3. Postmultiply $\overline{\mathcal{B}}^{(i)}(z^{-1})$ by the c.s.p.m. of order k_i

$$\mathcal{B}^{(i)}(z^{-1}) = \overline{\mathcal{B}}^{(i)}(z^{-1})U_R^{(i)}(z), \tag{2.64}$$

(this multiplication shifts the matrix of coefficients of $\overline{\mathcal{B}}^{(i)}(z^{-1})$, left by k_i columns of zeros, see eqn. 2.56, 2.62, 2.63). Update the matrix

$$K_R^{(i)}(z) = K_R^{(i-1)}(z)S_R^{(i)}(z). \tag{2.65}$$

This ends the i-th iteration.

Combining eqns. 2.62 to 2.65, the i-th iteration of the algorithm results in

$$\mathcal{B}^{(i)}(z^{-1}) = \mathcal{B}^{(i-1)}(z^{-1})Q_R^{(i)}U_R^{(i)}(z) = \mathcal{B}^{(i-1)}(z^{-1})S_R^{(i)}(z) = B(z^{-1})K_R^{(i)}(z),$$

where $S_R^{(i)}(z)$ and $K_R^{(i)}(z)$ are defined by eqns. 2.61 and 2.65.

The final iteration $(t = i - 1)$ yields

$$\mathcal{B}^{(t)}(z^{-1}) = B(z^{-1})K_R(z), \tag{2.66}$$

where $K_R(z)$ defined by eqn. 2.60 is $K_R(z) = K_R^{(t)}(z)$. Hence the property given by eqn. 2.50 is satisfied and $M_R = \mathcal{B}_0^{(t)}$. Using eqn. 2.59, one has

$$\det K_R^{(t)}(z) = (-1)^{tk_ir_i} \prod_{i=1}^{t} z^{k_i} \det Q_R^{(i)}, \tag{2.67}$$

which satisfies the property given by eqn. 2.51.

Part 2.

The completeness of the algorithm will be established using the following result.

Lemma 2.1 *If a $m \times l$ polynomial matrix $B(z^{-1})$ given by eqn. 2.56 is of full (normal) rank then its matrix of coefficients B is of full rank (in the field of complex numbers).*

Proof. Lemma 2.1 is a direct consequence of [112, Lemma G-2, p. 615]. Q.E.D.

Using the above lemma, part 2 of the proof of Theorem 2.1 is given as follows. From assumption that the transfer matrix $H_{y,u}(z)$ has full rank, and from eqn. 2.52 it follows that the polynomial matrix $B(z^{-1})$ has full (normal) rank. Hence (from Lemma 2.1) it follows that the row block-matrix of coefficients B has full rank. Steps 2 and 3 of the algorithm (eqns. 2.63 and 2.64) do not change the rank of the resulting polynomial matrix, i.e.,

$$\operatorname{rank} B(z^{-1}) = \operatorname{rank} \overline{B}^{(i)}(z^{-1}) = \operatorname{rank} \mathcal{B}^{(i)}(z^{-1}) = \min(m, l),$$

because $S_R^{(i)}(z)$ is a $l \times l$ nonsingular matrix. It follows from the above lemma that the row block-matrix of coefficients $B^{(i)}$ of the polynomial matrix $B^{(i)}(z^{-1})$ has full rank. Note also that the algorithm extracts the linearly independent columns from matrix B (step 1) by searching sequentially through the columns of the matrix of coefficients $\mathcal{B}^{(i)}$ (modified at step 2 and 3). The rank of $\mathcal{B}^{(i)}$ is a nondecreasing function of i and the linearly independent columns are accumulated in $\mathcal{B}_{0R}^{(i)}$ until $\operatorname{rank} \mathcal{B}_0^{(t)} = min(m, l)$, at which step the algorithm stops.

Q.E.D.

It follows from eqns. 2.59 and 2.61 that the constant k_r in eqn. 2.51 is given by

$$k_r = \deg\det K_R(z) = \sum_{i=1}^{t} k_i, \quad k_i = l - r_i.$$

Summarizing, the algorithm described above operates on the row block-matrix B of coefficients of the polynomial matrix $B(z^{-1})$. The full-rank coefficient matrix $\mathcal{B}_0^{(t)}$ of $\mathcal{B}^{(t)}(z)$ results from factorization of $\mathcal{B}_0^{(i)}$ and shifting its columns left. The macro XRNI, which is written in MATLAB commands and implements the above algorithm with QR factorization, is given in Appendix D.

Example 2.2. Calculation of the right nilpotent interactor matrix using MATLAB software package.

This example illustrates calculation of the RNI matrix using MATLAB [110] (or MATRIX$_X$ [113]) software package.

Consider the system of example 2.1 described by the transfer matrix 2.47. This transfer matrix can be factored into the LMF description with the monic denominator polynomial matrix, as follows

$$B(z^{-1}) = \begin{bmatrix} 1+2z^{-1} & 1+z^{-1} \\ 1+4z^{-1} & 1+3z^{-1} \end{bmatrix} \quad A(z^{-1}) = \begin{bmatrix} 1+3z^{-1}+2z^{-2} & 0 \\ 0 & 1+7z^{-1}+12z^{-2} \end{bmatrix}. \tag{2.68}$$

The numerator matrix can be expressed as (see eqn. 2.56)

$$B(z^{-1}) = B(Z_2 \otimes I_2) = [B_0\ B_1\ B_2](Z_2 \otimes I_2) = \begin{bmatrix} 0 & 0 & 1 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 & 4 & 3 \end{bmatrix} (Z_2 \otimes I_2)\,. \tag{2.69}$$

The algorithm is implemented with QR decomposition defined by eqn. C.3 (Appendix C, p. 286).

Step 1 of the i-th iteration of the algorithm groups $k_i = l - \operatorname{rank} \mathcal{B}_0^{(i-1)} = l - r_i$ zero columns on the left hand side of the postmultiplier of the decomposition of $\mathcal{B}_0^{(i-1)}$ (see eqn. 2.62). In order to achieve this with QR decomposition defined by eqn. C.3 (p. 286), the matrix $X = (\mathcal{B}_0^{(i-1)})'$ will be factorized as $X = QR$; then it is necessary to reverse the order of columns of transpose of R and rows of transpose of Q

$$X' = \mathcal{B}_0^{(i-1)} = (QR)' = R'Q' = R'J_l^{-1}J_lQ' = (R'J_l)(J_lQ') = \begin{bmatrix} 0_{k_i} & \mathcal{B}_{0R}^{(i)} \end{bmatrix} (Q_R^{(i)})^{-1},$$

where J_l is the $l \times l$ standard involuntary permutation matrix defined by eqn. C.4 (Appendix C, p. 286).

Step 2 of the i-th iteration of the algorithm postmultiplies the $\mathcal{B}^{(i-1)}(z^{-1})$ matrix by $Q_R^{(i)} = (J_lQ')^{-1} = (Q')^{-1}J_l^{-1} = QJ_l$ (see eqn. 2.63). Thus $Q_R^{(i)}$ is calculated by reversing the order of columns of matrix Q resulting from QR decomposition of $(\mathcal{B}_0^{(i-1)})'$. The row block-matrix of coefficients of the polynomial matrix $\overline{B}^{(i)}(z^{-1})$ resulting from step 2, has the following form

$$\overline{B}^{(i)} = \left[\overline{B}_0^{(i)}\ \overline{B}_1^{(i)}\ \ldots \overline{B}_n^{(i)}\right] = \begin{bmatrix} 0_{k_i}\ \mathcal{B}_{0R}^{(i)} & \mathcal{B}_{1L}^{(i)}\ \mathcal{B}_{1R}^{(i)} & \ldots & \mathcal{B}_{nL}^{(i)}\ \mathcal{B}_{nR}^{(i)} \end{bmatrix}. \tag{2.70}$$

In step 3 of the i-th iteration of the algorithm matrix $\overline{B}^{(i)}(z^{-1})$ of degree n is postmultiplied by a first degree matrix $U_R^{(i)}(z)$

$$\mathcal{B}^{(i)}(z^{-1}) = \overline{B}^{(i)}(z^{-1})U_R^{(i)}(z) = \left[\mathcal{B}_{-1}^{(i)}\;\mathcal{B}_0^{(i)}\;\ldots\;\mathcal{B}_n^{(i)}\right]\left(\left[z\;1\;\ldots\;z^{-n}\right]'\otimes I_l\right).$$

Note that $\mathcal{B}_{-1}^{(i)} = 0$, because

$$\mathcal{B}_{-1}^{(i)} = \overline{B}_0^{(i)}U_0^{(i)} = \mathcal{B}_0^{(i-1)}Q_R^{(i)}U_0^{(i)} = \left[\begin{array}{cc} 0_{k_i} & \mathcal{B}_{0R}^{(i)} \end{array}\right]\left[\begin{array}{cc} 0_{r_i} & I_{k_i} \\ 0_{r_i} & 0_{k_i} \end{array}\right] = 0,$$

where 0_{k_i} denotes k_i-column zero matrix. The row block-matrix of the coefficients of the polynomial matrix $\mathcal{B}^{(i)}(z^{-1})$ resulting from step 3, has the following form (cf. eqn. 2.70)

$$\mathcal{B}^{(i)} = \left[\mathcal{B}_{-1}^{(i)}\;\mathcal{B}_0^{(i)}\;\ldots\;\mathcal{B}_n^{(i)}\right] = \left[\begin{array}{cccccc} 0_{r_i}\;0_{k_i} & & \mathcal{B}_{0R}^{(i)}\;\mathcal{B}_{1L}^{(i)} & \ldots & \mathcal{B}_{nR}^{(i)}\;0_{k_i} \end{array}\right]. \tag{2.71}$$

The operation of multiplication of polynomial matrices is represented according to eqn. B.6.

The algorithm is initialized with $\mathcal{B}^{(0)}(z^{-1}) = B(z^{-1})$ and $K_R^{(0)}(z) = K_0^{(0)} = I_2$. Then

i=1: $r_1 = 0$, $k_1 = 2$

step 1.

Since $r_1 = \operatorname{rank}\mathcal{B}_0^{(0)} = \operatorname{rank}B_0 = 0$ a decomposition of B_0 yields $T = 0$ (see eqn. C.3) and $Q_R^{(1)}$ can be chosen identity. Then

$$U_R^{(1)}(z) = \begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix}, \quad U_R^{(1)} = [U_0^{(1)}\;U_1^{(1)}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix};$$

step 2.

$$\overline{B}^{(1)} = \left[\overline{B}_0^{(1)}\;\overline{B}_1^{(1)}\;\overline{B}_2^{(1)}\right] = \mathcal{B}^{(0)}Q_R^{(1)} = \mathcal{B}^{(0)}I_2 = [B_0\;B_1\;B_2];$$

step 3.

$$\begin{aligned} \mathcal{B}^{(1)} &= [\mathcal{B}_{-1}^{(1)}\;\mathcal{B}_0^{(1)}\;\mathcal{B}_1^{(1)}\;\mathcal{B}_2^{(1)}] = \overline{B}^{(1)}\langle U_R^{(1)}\rangle_2 = \\ &= \begin{bmatrix} 0 & 0 & 1 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 4 & 3 & 0 & 0 \end{bmatrix}, \end{aligned}$$

$$K_R^{(1)} = [K_0^{(1)}\ K_1^{(1)}] = K_R^{(0)} Q_R^{(1)} U_R^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{2.72}$$

Since the system transfer function $H_{y,u}(z)$ is strictly proper, we have $B_0 = 0$ in eqn. 2.69. Hence, this iteration can be omitted and the algorithm initialized with $\mathcal{B}^{(1)}(z^{-1}) = B(z)U_R^{(1)}(z)$ represented by $[\mathcal{B}_0^{(1)}\ \mathcal{B}_1^{(1)}\ \mathcal{B}_2^{(1)}]$, and $K_R^{(1)}(z) = zI_2$.

i=2: $r_2 = 1$, $k_2 = 1$,

step 1.

$$Q = \begin{bmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}, \quad R = \begin{bmatrix} -1.4142 & -1.4142 \\ 0 & 0 \end{bmatrix},$$

$$Q_R^{(2)} = QJ_2 = \begin{bmatrix} -0.7071 & -0.7071 \\ 0.7071 & -0.7071 \end{bmatrix},$$

$$U_R^{(2)}(z) = \begin{bmatrix} 0 & z \\ 1 & 0 \end{bmatrix}, \quad U_R^{(2)} = [U_0^{(2)}\ U_1^{(2)}] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix};$$

step 2.

$$\begin{aligned} \overline{B}^{(2)} &= \left[\overline{B}_0^{(2)}\ \overline{B}_1^{(2)}\ \overline{B}_2^{(2)}\right] = B^{(1)} Q_R^{(2)} = \\ &= \begin{bmatrix} 0 & -1.4142 & -0.7071 & -2.1213 & 0 & 0 \\ 0 & -1.4142 & -0.7071 & -4.9497 & 0 & 0 \end{bmatrix}; \end{aligned}$$

step 3.

$$\begin{aligned} \mathcal{B}^{(2)} &= [\mathcal{B}_{-1}^{(2)}\ \mathcal{B}_0^{(2)}\ \mathcal{B}_1^{(2)}\ \mathcal{B}_2^{(2)}] = \overline{B}^{(2)} \langle U_R^{(2)} \rangle_2 = \\ &= \begin{bmatrix} 0 & 0 & -1.4142 & -0.7071 & -2.1213 & 0 & 0 & 0 \\ 0 & 0 & -1.4142 & -0.7071 & -4.9497 & 0 & 0 & 0 \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} K_R^{(2)} &= [K_0^{(2)}\ K_1^{(2)}\ K_2^{(2)}] = K_R^{(1)} Q_R^{(2)} \langle U_R^{(2)} \rangle_1 = \\ &= \begin{bmatrix} 0 & -0.7071 & -0.7071 & 0 & 0 & 0 \\ 0 & 0.7071 & -0.7071 & 0 & 0 & 0 \end{bmatrix}; \end{aligned}$$

i=3: $r_3 = 1$, $k_3 = 1$,

step 1.

$$Q = \begin{bmatrix} -0.8944 & -0.4472 \\ -0.4472 & 0.8944 \end{bmatrix}, \quad R = \begin{bmatrix} 1.5811 & 1.5811 \\ 0 & 0 \end{bmatrix},$$

$$Q_R^{(3)} = QJ_2 = \begin{bmatrix} -0.4472 & -0.8944 \\ 0.8944 & -0.4472 \end{bmatrix},$$

$$U_R^{(3)}(z) = \begin{bmatrix} 0 & z \\ 1 & 0 \end{bmatrix}, \quad U_R^{(3)} = [U_0^{(3)}\ U_1^{(3)}] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix};$$

step 2.

$$\overline{B}^{(3)} = \left[\overline{B}_0^{(3)}\ \overline{B}_1^{(3)}\ \overline{B}_2^{(3)}\right] = B^{(2)}Q_R^{(3)} = \begin{bmatrix} 0 & 1.5811 & 0.9487 & 1.8974 & 0 & 0 \\ 0 & 1.5811 & 2.2136 & 4.4272 & 0 & 0 \end{bmatrix};$$

step 3.

$$\begin{aligned} \mathcal{B}^{(3)} &= [\mathcal{B}_{-1}^{(3)}\ \mathcal{B}_0^{(3)}\ \mathcal{B}_1^{(3)}\ \mathcal{B}_2^{(3)}] = \overline{B}^{(3)}\langle U_R^{(3)}\rangle_2 = \\ &= \begin{bmatrix} 0 & 0 & 1.5811 & 0.9487 & 1.8974 & 0 & 0 & 0 \\ 0 & 0 & 1.5811 & 2.2136 & 4.4272 & 0 & 0 & 0 \end{bmatrix}, \end{aligned} \tag{2.73}$$

$$\begin{aligned} K_R^{(3)} &= [K_0^{(3)}\ K_1^{(3)}\ K_2^{(3)}\ K_3^{(3)}] = K_R^{(2)}Q_R^{(3)}\langle U_R^{(3)}\rangle_2 = \\ &= \begin{bmatrix} 0 & -0.6325 & 0.3162 & 0.3162 & 0.6325 & 0 & 0 & 0 \\ 0 & 0.6325 & -0.3162 & 0.3162 & 0.6325 & 0 & 0 & 0 \end{bmatrix}; \end{aligned}$$

i=4: $r_4 = 2$ which terminates the algorithm and the right nilpotent interactor matrix is

$$K_R(z) = K_R^{(3)}(z) = \begin{bmatrix} 0.3162z^2 + 0.6325z & -0.6325z^3 + 0.3162z^2 \\ -0.3162z^2 + 0.6325z & 0.6325z^3 + 0.3162z^2 \end{bmatrix}, \tag{2.74}$$

with its determinant $\det K_R(z) = z^4$.

2.3.4 Comments on the algorithms for the calculation of nilpotent interactor matrices.

An important feature of the class of algorithms developed in this section is that the nilpotent interactor matrices are evaluated *directly* from the matrix fraction representation of the system transfer matrix or from the system *difference operator representation.* This is in contrast to the original algorithm proposed in [33] for the calculation of the unique interactor from the system *transfer matrix* (i.e., from the matrix of rational functions). No *division* of polynomial matrices is required in evaluation of nilpotent interactor matrices. This is in contrast to the algorithm proposed in [39] for the calculation of the interactor from the system DARMA model. Furthermore, an inherent feature of the new algorithms is that they operate on the *numerical representation* of polynomial matrices and thus are amenable to *computer-based calculations.* The above features become especially useful for the implementation of the indirect, self-tuning minimum prediction error controller, with the *on-line* calculation of the RNI matrix from the estimates of the polynomial matrices of the DARMA model (see approach 5 (b) on page 39, and chapter 4).

The implementation of the algorithms for calculation of the nilpotent interactor matrices requires only the following operations: factorization of real matrices of dimension not larger than the dimension of the numerator polynomial matrix, shifting and multiplication of real matrices. Various matrix factorization can be employed in the implementation of the algorithms to calculate matrices $Q_L^{(i)}$ or $Q_R^{(i)}$, e.g., Gaussian elimination LDU, QR, or singular value decomposition SVD [111]. The choice of matrix decomposition involves a compromise between numerical properties and computational effort. The factorization involving orthogonal matrices (such as QR or SVD) is desirable since it makes computing of the inverse trivial, with no new rounding errors introduced [114].

Note that for a strictly proper transfer matrix given by a matrix fraction representation, one has $\deg \hat{N}(z) < \deg \hat{D}(z)$ and $\deg \hat{B}(z) < \deg \hat{A}(z)$. Then the first step of the algorithm yields $S_L^{(1)}(z) = S_R^{(1)}(z) = zI$. This implies another property of the nilpotent interactor matrices in common with the interactors $\xi_L(z)$ and $\xi_R(z)$.

Remark 2.2 *For a strictly proper transfer matrix $H_{y,u}(z)$ the elements of the nilpotent interactor matrices $K_L(z)$ and $K_R(z)$ are zeros or are divisible by z.*

Comment 2.2 *Note that if $r_i = 0$ for $i = 1, 2, \ldots, j$ then the initial j iterations can be omitted and the algorithms can be initialized with $\mathcal{N}^{(j)}(z^{-1}) = (U_L^{(1)}(z))^j N(z^{-1})$ and $K_L^{(j)}(z) = z^j I_m$, or $\mathcal{B}^{(j)}(z^{-1}) = B(z^{-1})(U_R^{(1)}(z))^j$ and $K_R^{(j)}(z) = z^j I_l$.*

The algorithms presented in subsections 2.3.2 and 2.3.3 were introduced for calculation of the LNI and RNI matrices from RMF (or RDO) and LMF (or LDO) descriptions, respectively. This is represented by the vertical arrows (2) and (2') in fig. 2.8. However, it is possible to use the algorithm developed for evaluation of the LNI (RNI) matrix for calculation of the RNI (LNI) matrix as well. This is shown below for the RNI matrix calculated using the algorithm developed for the LNI matrix.

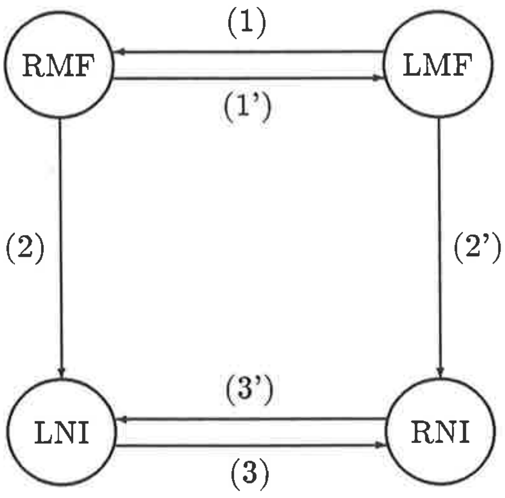

Figure 2.8: Algorithms for evaluation of the LNI and RNI matrices.

We can write from eqn. 2.52 that

$$H_{y,u}(z)' = (A^{-1}(z^{-1})B(z^{-1}))' = N(z^{-1})D^{-1}(z^{-1}), \tag{2.75}$$

where $N(z^{-1}) = B(z^{-1})'$ and $D(z^{-1}) = A(z^{-1})'$. Then using eqn. 2.42 one has $\lim_{z\to\infty} K_L(z)N(z^{-1})D^{-1}(z^{-1}) = M_L$, where $K_L(z)$ is the LNI matrix calculated from

$N(z^{-1})$. Hence, by transposition one has

$$\lim_{z\to\infty}(K_L(z)N(z^{-1})D^{-1}(z^{-1}))' = \lim_{z\to\infty}(D(z^{-1})')^{-1}N(z^{-1})'K_L(z)'$$
$$= \lim_{z\to\infty} A^{-1}(z^{-1})B(z^{-1})K_R(z) = M_R, \tag{2.76}$$

where $K_R(z) = K_L(z)'$ and $M_R = M_L'$. Thus the RNI matrix can be determined from the LMF description using the algorithm for calculation of the LNI matrix in three steps (see fig. 2.8): (1) calculate transposition of the numerator polynomial matrix of the LMF description, i.e., $N(z^{-1}) = B(z^{-1})'$; (2) calculate the LNI matrix $K_L(z)$ from $N(z^{-1})$ using algorithm of subsection 2.3.2; (3) calculate the RNI matrix by transposition of the LNI matrix resulting from step (2), i.e., $K_R(z) = K_L(z)'$.

It can be shown in a similar way that the LNI matrix can be calculated from the RMF description using algorithm for the calculation of the RNI matrix in the following steps (see fig. 2.8): (1') calculate transposition of the numerator of the RMF description $B(z^{-1}) = N(z^{-1})'$; (2') calculate the RNI matrix $K_R(z)$ from $B(z^{-1})$ using algorithm of subsection 2.3.3; (3') calculate the LNI matrix by transposition of the RNI matrix resulting from step (2'), i.e., $K_L(z) = K_R(z)'$. (Note that step (1) ((1')) in fig. 2.8 denotes transposition of the matrix fraction description, not the transformation from the LMF to RMF (RMF to LMF) descriptions of the system transfer matrix.)

The above discussion explains duality of the algorithms and shows that only one of them is required for calculation of the LNI and RNI matrices. However, the choice of the direct calculation of the interactor matrix, i.e., (2) for the LNI and (2') for the RNI matrix, can further reduce computational burden for particular applications (see Aside 4.1, p. 127).

Chapter 3

Output predictors for self-tuning predictive control.

In this chapter problem of extrapolating a time series into future, i.e., prediction, is considered. Prediction is an important topic in itself since it is encountered in many engineering or scientific problems [115,83,116,117,118]. However, it is the approach to the derivation of self-tuning predictive control strategies, undertaken in next chapters, which will characterize developments in this chapter. Nevertheless, certain aspects of prediction techniques (in particular adaptive predictors for systems having the feedback configuration FD derived in subsections 3.2.2 and 3.3.4), and of computational methods presented in this chapter, can be treated separately from the developments in the following chapters.

For the derivation of predictors we assume knowledge of the structure of the model of a system generating the time series the future values of which are to be predicted [1]. Furthermore, it is assumed that the prediction is a linear function of past and present system outputs and (measurable or known) inputs.

The problem of developing a predictor for a time series, generated by a deterministic system with known model structure, reduces to finding the system model in a predictor form; such a form of the system model permits prediction without

[1] An alternative approach is to postulate the predictor form which does not necessarily correspond to the actual structure of the system model (the so-called *restricted complexity predictors*) [9, p. 110].

error. For stochastic systems, perfect prediction is not possible. In the following, the predictors are designed to minimize the variance of the prediction error.

The organization of this chapter is as follows. The predictors for deterministic systems having the feedback configuration [2] FI and FD are introduced in sections 3.1 and 3.2, respectively. The predictor for a stochastic system having the feedback configuration FD is derived in section 3.3 (case FI will be discussed briefly in subsection 5.2.1).

The development of predictors for multivariable systems in section 3.1 is related to the results presented in [9,35]. New aspects involve multi-step-ahead prediction and application of the nilpotent interactor matrices in the design of predictors. New computational algorithms for predictor parameters are derived which are amenable to computer-based calculations using any matrix-oriented software. The developments of sections 3.2 and 3.3 are new.

3.1 Output predictors for a multi-input, multi-output deterministic system having the feedback configuration FI.

The problem of predicting the output of a deterministic multi-input, multi-output system is addressed below.

We make the following assumptions.

Assumption 3.1 *A deterministic multivariable system is given having the feedback configuration FI (see page 18), i.e.,* $y_F(t) \equiv y(t)$*, with the number of outputs* $y(t)$ *equal to the number of inputs* $u(t)$*, i.e.,* $\dim y(t) = \dim u(t) = m$.

Such a system will be called a multi-input, multi-output square (MIMOS) system.

[2] In the context of prediction, the feedback outputs $y_F(t)$ represent the system outputs which can be utilized in the predictor for the output $y(t)$ (cf. discussion on p. 18 and fig. 2.1).

Assumption 3.2 *The MIMOS system is described by the DARMA model (see eqn. 2.17, p. 27)*

$$A(p)y(t) = B(p)u(t), \tag{3.1}$$

with appropriate initial conditions on $\{y(t)\}$. The $m \times m$ polynomial matrices $A(p)$ and $B(p)$ are defined using the polynomial matrix representation of Appendix B (see eqns. B.1 to B.3)

$$\begin{aligned} A(p) &= [I_m \ A_1 \ \ldots \ A_n](P_n \otimes I_m) = A(P_n \otimes I_m), & (3.2) \\ B(p) &= p[B_1 \ B_2 \ \ldots \ B_n](P_{n-1} \otimes I_m) = pB(P_{n-1} \otimes I_m). & (3.3) \end{aligned}$$

Note that A and B denote the $m \times (n+1)m$ and $m \times nm$ matrices of coefficients of the polynomial matrices $A(p)$ and $B(p)$.

The DARMA model permits the description of a deterministic disturbance affecting the plant (this corresponds to the generator of deterministic disturbance incorporated in the system model).

The system transfer matrix $H_{y,u}(z)$ corresponding to the DARMA model 3.1, which is found by taking the z-transform and assuming zero initial conditions, is

$$H_{y,u}(z) = (z^n A(z^{-1}))^{-1}(z^n B(z^{-1})) = \hat{A}^{-1}(z)\hat{B}(z). \tag{3.4}$$

Assumption 3.3 *The system transfer matrix $H_{y,u}(z)$ has full normal rank and is strictly proper.*

Note that Assumption 3.3 ensures output function controllability [75, Theorem 5.5.7, p. 164] [9, Lemma 5.2.2, p. 132], and guarantees existence of the interactor matrices (see Definitions 2.1 (p. 36), 2.2 (p. 37), 2.3 (p. 43), and 2.4 (p. 45)).

The predictors developed in this section will be employed in the development of the long-range predictive control law based on the minimization of a multi-stage cost function (section 5.1), and of the minimum prediction error control law based on a single-stage cost function (chapter 4). In order to guarantee causality and stability of the controllers, the cost functions involve system output or input variables which are filtered by the system interactor matrix (rather than the actual output and input variables). The choice of the left interactor matrix is associated with the cost function involving filtered tracking error, i.e., involving filtered output variables (see eqn. 5.1, p. 152). The choice of the right interactor matrix is associated with the cost function involving filtered control signal, i.e., involving filtered input variables (see $u_k(t)$ in eqn. 4.1, p. 120).

For the purpose of minimizing the multi-stage cost function, the multi-step-ahead predictor is needed; for the purpose of minimizing the single-stage k-step-ahead cost function, the k-step-ahead predictor is employed. The choice of the left or right interactor matrix determines respectively whether the prediction of filtered system outputs will be required, or whether the predictor will involve filtered system inputs. Therefore, the following approaches to multivariable predictive control can be postulated (among many other possibilities [67]):

(i) the single-stage cost function penalizing both the filtered tracking error and the control signal (the k-step-ahead predictor based on the left interactor matrix is employed) [34,9,39];

(ii) the multi-stage cost function penalizing both the filtered tracking error and the control signal, see section 5.1 (the multi-step-ahead predictor based on the left interactor matrix is employed, see subsection 3.1.1);

(iii) the single-stage cost function penalizing both the tracking error and the filtered control signal, see chapter 4 (the k-step-ahead predictor based on the right interactor matrix is employed, see subsection 3.1.2) [35];

(iv) the multi-stage cost function penalizing both the tracking error and the filtered control signal (the multi-step-ahead predictor based on the right interactor matrix is employed) – not considered.

Approaches (ii) and (iii) will be explored below.

3.1.1 Multi-step-ahead predictor based on the left interactor matrix for a multi-input, multi-output, square deterministic system.

In this subsection:

- the multi-step-ahead predictor based on the left interactor matrix for a MIMOS system is derived;
- algorithms are derived for both the direct and recursive calculation of the matrices of coefficients which represent the polynomial matrices defining the multi-step-ahead predictor.

In this subsection the approach (ii) in the classification on p. 60 is considered. The idea of employing the unique left interactor matrix for the development of the predictor was originally proposed in [34,9] for the k-step-ahead case (approach (i)). Here a multi-step-ahead predictor is given by the following result.

Lemma 3.1 *Consider a MIMOS system satisfying Assumptions 3.1, 3.2, and 3.3 (pp. 58-59). For such a system let us define an auxiliary system as follows*

$$y_k(t) = \overline{K}_L(q)y(t), \tag{3.5}$$

where $\overline{K}_L(q)$ is the $m \times m$ left interactor matrix [3] *of the system 3.4. Then the multi-step-ahead prediction of filtered system outputs, $y_k(t+j)$, is given by the following predictor expression*

$$y_k(t+j) = \begin{bmatrix} \alpha^{(j)}(p) & \beta^{(j)}(p) \end{bmatrix} \begin{bmatrix} y(t) \\ u(t+j) \end{bmatrix}. \tag{3.6}$$

[3] To be precise, the polynomial matrix $\overline{K}_L(q)$ is the matrix which results from the left interactor matrix $\overline{K}_L(z)$ by substituting q for z.

The $m \times m$ polynomial matrix [4] $\alpha^{(j)}(p)$ *results from the following Diophantine equation* [5]

$$q^j \overline{K}_L(q) = \left[\begin{array}{cc} F^{(j)}(q) & \alpha^{(j)}(p) \end{array} \right] \left[\begin{array}{c} A(p) \\ I_m \end{array} \right], \tag{3.7}$$

and the $m \times m$ polynomial matrix $\beta^{(j)}(p)$ is given by

$$\beta^{(j)}(p) = p^j F^{(j)}(q) B(p), \tag{3.8}$$

where

$$\begin{aligned} F^{(j)}(q) &= \left[F_0^{(j)} \ F_1^{(j)} \ \ldots \ F_{k+j-1}^{(j)}\right] \left(\left[q^{k+j} \ q^{k+j-1} \ \ldots \ q\right]' \otimes I_m\right) \\ &= F^{(j)} \left(\left[q^{k+j} \ q^{k+j-1} \ \ldots \ q\right]' \otimes I_m\right), \qquad (3.9) \\ \alpha^{(j)}(p) &= \left[\alpha_0^{(j)} \ \alpha_1^{(j)} \ \ldots \ \alpha_{n-1}^{(j)}\right] (P_{n-1} \otimes I_m) = \alpha^{(j)}(P_{n-1} \otimes I_m), \qquad (3.10) \\ \beta^{(j)}(p) &= \left[\beta_{-j}^{(j)} \ \beta_{-j+1}^{(j)} \ \ldots \ \beta_0^{(j)} \ \ldots \ \beta_{n-1}^{(j)}\right] (P_{n+j-1} \otimes I_m) \\ &= \beta^{(j)}(P_{n+j-1} \otimes I_m), \qquad (3.11) \end{aligned}$$

k is the degree of the interactor matrix $\overline{K}_L(q)$, and $\beta_{-j}^{(j)} = M_L$, where the nonsingular matrix M_L is defined by eqn. 2.40 (p. 42).

The proof of the above lemma is similar to that of [9, Theorem 5.2.4, p. 137] and is omitted.

In eqns. 3.9 to 3.11, the representation of polynomial matrices given by eqns. B.1 to B.3 (Appendix B) is used. The following convention is chosen to denote coefficient matrices $\beta_i^{(j)}$ of the polynomial matrix $\beta^{(j)}(p)$: the subscript i is positive for coefficients associated with past values of the input signal $u(t)$ in the predictor expression 3.6, $i = 0$ for the coefficient corresponding to the present value of the input, i is negative for coefficients associated with future values of the input.

The above lemma is an extension of [9, Theorem 5.2.4, p. 137] where the predictor expression for $y_k(t)$ (i.e., $j = 0$) was introduced for the auxiliary system 3.5 defined with the unique interactor matrix. Here, the left interactor matrix $\overline{K}_L(q)$ may be

[4] The superscript (j) is used to denote polynomial matrices associated with the predictor expression for $y_k(t + j)$.

[5] Recall that $p = q^{-1}$ (eqn. A.1, Appendix A).

either the unique interactor matrix $\overline{K}_L(q) = \xi_L(q)$ or the left nilpotent interactor (LNI) matrix $\overline{K}_L(q) = K_L(q)$ (introduced in subsection 2.3.2).

It follows from Remark 2.2 (p. 55) and from Definition 2.1 (p. 36) that for a strictly proper transfer matrix $H_{y,u}(z)$ the coefficient matrix of the left interactor matrix $\overline{K}_L(q)$ associated with q^0 is zero. Thus the left interactor matrix can be represented by

$$\begin{aligned}\overline{K}_L(q) &= \left[\overline{K}_0\ \overline{K}_1\ \cdots\ \overline{K}_{k-1}\right]\left(\left[q^k\ q^{k-1}\ \ldots q\right]' \otimes I_m\right)\\ &= \overline{K}_L\left(\left[q^k\ q^{k-1}\ \ldots\ q\right]' \otimes I_m\right).\end{aligned} \tag{3.12}$$

The multi-step-ahead prediction $y_k(t+j)$ can be written as

$$y_k(t+j) = q^j y_k(t) = \overline{K}_L(q)y(t+j) = \overline{K}_0 y(t+j+k) + \cdots + \overline{K}_{k-1}y(t+j+1). \tag{3.13}$$

Note from eqn. 3.13 that $y_k(t+j)$ involves future values of the system outputs $y(t)$ up to sample instant $t+k+j$. Therefore, the term "multi-step-ahead" prediction is used to differentiate from the k-step-ahead prediction involving system outputs up to time $t+k$, where k is the degree of the polynomial matrix $\overline{K}_L(q)$.

The *coefficient matrices* of the polynomial matrices $F^{(j)}(p)$ and $\alpha^{(j)}(p)$ can be found by equating coefficients in the polynomial equation 3.7 [9]. Alternatively, one can equate *matrices of coefficients* of polynomial matrices in eqn. 3.7, using the numerical representation of polynomial matrices given in Appendix B. This leads to a compact matrix equation for *matrices of coefficients* of polynomial matrices $F^{(j)}(q)$ and $\alpha^{(j)}(p)$ as shown below.

Using eqns. B.4 to B.6 (Appendix B), polynomial equation 3.7 can be represented by

$$\overline{K}_L(R_0^{k,n+k+j} \otimes I_m) = \left[\begin{array}{cc} F^{(j)} & \alpha^{(j)} \end{array}\right] \left[\begin{array}{c} \langle A\rangle_{k+j-1} \\ R_{k+j}^{n,n+k+j} \otimes I_m \end{array}\right], \tag{3.14}$$

where matrices $\overline{K}_L$, $F^{(j)}$, $\alpha^{(j)}$, and A, are defined by eqns. 3.12, 3.9, 3.10, and 3.2, respectively. The structure of eqn. 3.14 allows it to be separated into two equations as follows

$$\overline{K}_L(R_0^{k,k+j} \otimes I_m) = F^{(j)}\langle A\rangle_{k+j-1}^L, \tag{3.15}$$

$$0 \;=\; F^{(j)}\langle A\rangle^R_{k+j-1} + \alpha^{(j)}, \tag{3.16}$$

where $\langle A\rangle^L_{k+j-1}$ is a $(k+j)m \times (k+j)m$ upper-triangular block Toeplitz matrix

$$\langle A\rangle^L_{k+j-1} = \begin{bmatrix} I_m & A_1 & \dots & A_{k+j-1} \\ 0 & I_m & \dots & A_{k+j-2} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & I_m \end{bmatrix}, \tag{3.17}$$

and $\langle A\rangle^R_{k+j-1}$ is a $(k+j)m \times nm$ matrix

$$\langle A\rangle^R_{k+j-1} = \begin{bmatrix} A_{k+j} & \dots & A_n & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ A_1 & A_2 & \dots & A_k & \dots & A_n \end{bmatrix}, \tag{3.18}$$

($A_i = 0$ for $i > n$). The matrix of coefficients of the polynomial matrix $F^{(j)}(q)$ can be evaluated from eqn. 3.15 as

$$F^{(j)} = \overline{K}_L(R_0^{k,k+j} \otimes I_m)((\langle A\rangle^L_{k+j-1})^{-1}, \tag{3.19}$$

in which the matrix to be inverted, given by eqn. 3.17, is nonsingular.

Once matrix $F^{(j)}$ is found, the matrix of coefficients of the predictor polynomial $\alpha^{(j)}(p)$ follows from eqn. 3.16 as

$$\alpha^{(j)} = -F^{(j)}\langle A\rangle^R_{k+j-1}. \tag{3.20}$$

Since $\langle A\rangle^L_{k+j-1}$ is a triangular block Toeplitz matrix, its inverse is also triangular block Toeplitz given by

$$((\langle A\rangle^L_{k+j-1})^{-1} = \begin{bmatrix} E_0 & E_1 & \dots & E_{k+j-1} \\ 0 & E_0 & \dots & E_{k+j-2} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & E_0 \end{bmatrix}. \tag{3.21}$$

The matrix E_i can be found from the following algorithm [119,120,121,79]

$$\begin{aligned} E_0 &= A_0^{-1} = I_m, \\ E_i &= -\begin{bmatrix} E_0 & \dots & E_{i-1} \end{bmatrix} \begin{bmatrix} A_i \\ \vdots \\ A_1 \end{bmatrix}, \end{aligned} \tag{3.22}$$

for $i = 1, 2, \ldots, k + j - 1$.

The expression for the matrix of coefficients $\beta^{(j)}$ of the polynomial matrix $\beta^{(j)}(p)$ follows from eqn. 3.8 using eqn. B.6 (Appendix B) and noting that the first $k - 1$ coefficient matrices of the polynomial matrix resulting from eqn. 3.8 are zero

$$\beta^{(j)} = F^{(j)} \langle B \rangle^R_{k+j-1}, \tag{3.23}$$

where matrix B is given by eqn. 3.3, and the $(k+j)m \times (n+j)m$ matrix $\langle B \rangle^R_{k+j-1}$ is

$$\langle B \rangle^R_{k+j-1} = \begin{bmatrix} B_k & \ldots & B_n & 0 & \ldots & \ldots & 0 \\ \vdots & \ddots & & \ddots & & \ddots & \vdots \\ B_1 & \ldots & B_k & \ldots & B_n & \ldots & 0 \\ 0 & \ddots & & \ddots & & \ddots & \vdots \\ 0 & \ldots & B_1 & \ldots & B_k & \ldots & B_n \end{bmatrix}. \tag{3.24}$$

Let us now summarize the above results in the form of the following remark.

Remark 3.1 *The matrices of coefficients of the polynomials of the j-step-ahead predictor 3.6 are given by*

$$\alpha^{(j)} = -F^{(j)} \langle A \rangle^R_{k+j-1}, \tag{3.25}$$

$$\beta^{(j)} = F^{(j)} \langle B \rangle^R_{k+j-1}, \tag{3.26}$$

where

$$F^{(j)} = \overline{K}_L (R_0^{k,k+j} \otimes I_m)((\langle A \rangle^L_{k+j-1})^{-1}, \tag{3.27}$$

can be calculated using eqn. 3.21 and algorithm 3.22. Note that for $j = 0$ eqn. 3.27 reduces to

$$F^{(0)} = \overline{K}_L ((\langle A \rangle^L_{k-1})^{-1}. \tag{3.28}$$

The macro XPRL written in MATLAB commands and implementing algorithm 3.25 and 3.26 (for $j = 0$) is given in Appendix D.

A recursive algorithm for the calculation of the matrices of coefficients of the multi-step-ahead predictor polynomials.

For the application of the multi-step-ahead predictor in the long-range predictive control, knowledge of the predictions over an extended range of the prediction horizon

j is of interest, rather than for one value of the prediction horizon. The polynomial matrices $\alpha^{(j)}(p)$ and $\beta^{(j)}(p)$ can be found using the algorithm of Remark 3.1 for each value of j. An alternative approach, considered below, is to develop a recursive algorithm which evaluates (via matrices of coefficients) the polynomial matrices $\alpha^{(j)}(p)$ and $\beta^{(j)}(p)$ of the j-step-ahead predictor from the polynomial matrices $\alpha^{(j-1)}(p)$ and $\beta^{(j-1)}(p)$ of the $(j-1)$-step-ahead predictor. The idea of employing a recursive procedure for updating coefficients of a predictor for increasing prediction horizons was introduced in [100] for multi-input, single-output systems.

Since $\langle A\rangle_{k+j-1}^{L}$ is upper-triangular Toeplitz, and matrix $\overline{K}_L(R_0^{k,k+j} \otimes I_m)$ in eqn. 3.27 contains a $m \times jm$ null matrix on its right-hand side, it follows from eqn. 3.27 used for $j-1$ and for j, that the matrix of coefficients $F^{(j)}$ associated with the j-step-ahead predictor can be expressed in terms of $F^{(j-1)}$ as

$$F^{(j)} = \left[\begin{array}{cc} F^{(j-1)} & F^{(j)}_{k+j-1} \end{array} \right], \tag{3.29}$$

where the coefficient matrix $F^{(j)}_{k+j-1}$ is given by

$$F^{(j)}_{k+j-1} = \left[\begin{array}{ccc} \overline{K}_0 & \ldots & \overline{K}_{k-1} \end{array} \right] \left[\begin{array}{c} E_{k+j-1} \\ \vdots \\ E_j \end{array} \right]. \tag{3.30}$$

Now the result relating the coefficient matrix $F^{(j)}_{k+j-1}$ to the coefficient matrix $\alpha_0^{(j-1)}$ of the polynomial $\alpha^{(j-1)}(p)$ is established. It follows from eqn. 3.25 for $j-1$ that

$$\alpha_0^{(j-1)} = -F^{(j-1)} \left[\begin{array}{c} A_{k+j-1} \\ \vdots \\ A_1 \end{array} \right],$$

and substituting from eqn. 3.27 for $F^{(j-1)}$, and using eqn. 3.21

$$\alpha_0^{(j-1)} = -\left[\begin{array}{ccc} \overline{K}_0 & \ldots & \overline{K}_{k-1} \end{array} \right] \left[\begin{array}{ccccc} I_m & E_1 & E_2 & \ldots & E_{k+j-2} \\ 0 & \ddots & & & \vdots \\ 0 & \ldots & I_m & \ldots & E_{j-1} \end{array} \right] \left[\begin{array}{c} A_{k+j-1} \\ \vdots \\ A_1 \end{array} \right],$$

and from eqn. 3.22

$$\alpha_0^{(j-1)} = \begin{bmatrix} \overline{K}_0 & \dots & \overline{K}_{k-1} \end{bmatrix} \begin{bmatrix} E_{k+j-1} \\ \vdots \\ E_j \end{bmatrix}.$$

Finally using eqn. 3.30 one has

$$\alpha_0^{(j-1)} = F_{k+j-1}^{(j)}. \tag{3.31}$$

This result is used below.

The matrix of coefficients of $\alpha^{(j)}(p)$ will now be evaluated in terms of matrix of coefficients of $\alpha^{(j-1)}(p)$ using eqns. 3.25, 3.29, and 3.18 as follows

$$\begin{aligned}
\alpha^{(j)} &= -F^{(j)}\langle A\rangle_{k+j-1}^{R} \\
&= -\begin{bmatrix} F^{(j-1)} & F_{k+j-1}^{(j)} \end{bmatrix} \begin{bmatrix} A_{k+j} & A_{k+j+1} & \dots & A_n & \dots & 0 \\ \vdots & & \ddots & & \ddots & \vdots \\ A_1 & A_2 & \dots & A_{k+j+1} & \dots & A_n \end{bmatrix} \\
&= -F^{(j-1)} \begin{bmatrix} A_{k+j} & A_{k+j+1} & \dots & A_n & \dots & 0 \\ \vdots & \ddots & & \ddots & & \vdots \\ A_2 & \dots & A_{k+j+1} & \dots & A_n & 0 \end{bmatrix} - F_{k+j-1}^{(j)} \begin{bmatrix} A_1 \dots A_n \end{bmatrix} \\
&= -F^{(j-1)} \begin{bmatrix} A_{k+j-1} & A_{k+j} & \dots & A_n & \dots & 0 \\ \vdots & \ddots & \ddots & & \ddots & \vdots \\ A_1 & \dots & A_{k+j-1} & A_{k+j} & \dots & A_n \end{bmatrix} \begin{bmatrix} 0 & 0 & \dots & 0 \\ I_m & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \\ 0 & \dots & I_m & 0 \end{bmatrix} \\
&\quad -F_{k+j-1}^{(j)} \begin{bmatrix} A_1 & \dots & A_n \end{bmatrix} \\
&= -F^{(j-1)}\langle A\rangle_{k+j-2}^{R} \begin{bmatrix} 0 & 0 & \dots & 0 \\ I_m & 0 & \dots & 0 \\ \vdots & \ddots & \dots & 0 \\ 0 & \dots & I_m & 0 \end{bmatrix} - F_{k+j-1}^{(j)} \begin{bmatrix} A_1 & \dots & A_n \end{bmatrix},
\end{aligned}$$

and using eqns. 3.25 for $j-1$ and eqn. 3.31 one has

$$\alpha^{(j)} = \alpha^{(j-1)} \begin{bmatrix} 0 & 0 & \dots & 0 \\ I_m & 0 & \dots & 0 \\ \vdots & \ddots & \dots & 0 \\ 0 & \dots & I_m & 0 \end{bmatrix} + \alpha_0^{(j-1)} \begin{bmatrix} -A_1 & \dots & -A_n \end{bmatrix}. \tag{3.32}$$

Let us introduce the following $nm \times nm$ block top-companion matrix

$$A_c = \begin{bmatrix} -A_1 & -A_2 & \dots & -A_n \\ I_m & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & I_m & 0 \end{bmatrix}, \tag{3.33}$$

which has the column shifting property [80, A.31, p. 660]. Then the equation for the calculation of the matrix of coefficients of $\alpha^{(j)}(p)$ in terms of matrix of coefficients of $\alpha^{(j-1)}(p)$ can be written using eqns. 3.32 and 3.33 as

$$\alpha^{(j)} = \alpha^{(j-1)} A_c.$$

The matrix of coefficients of the polynomial matrix $\beta^{(j)}(p)$ is now evaluated in terms of the matrix of coefficients of $\beta^{(j-1)}(p)$. Eqns. 3.26, 3.29 and 3.31 imply

$$\beta^{(j)} = F^{(j)} \langle B \rangle^R_{k+j-1} = \begin{bmatrix} F^{(j-1)} & \alpha_0^{(j-1)} \end{bmatrix} \langle B \rangle^R_{k+j-1}.$$

Using eqn. 3.26 for $(j-1)$, and eqn. 3.3 one has

$$\beta^{(j)} = \beta^{(j-1)} (R_0^{n+j-1,n+j} \otimes I_m) + \alpha_0^{(j-1)} B (R_j^{n,n+j} \otimes I_m). \tag{3.34}$$

Although eqn. 3.34 appears complex, it is only a compact way of expressing the following operations: (1) shift the matrix $\beta^{(j-1)}$ to the left by m null columns, (2) multiply $\alpha_0^{(j-1)}$ by B, (3) shift the matrix $\alpha_0^{(j-1)} B$ to the right by jm null columns, (4) add the matrices calculated in steps (1) and (3).

Let us now summarize the above results in the following remark.

Remark 3.2 *The matrices of coefficients of the j-step-ahead predictor polynomial matrices can be calculated recursively from the matrices of coefficients of the $(j-1)$-step-ahead predictor polynomial matrices as follows*

$$\begin{aligned} \alpha^{(j)} &= \alpha^{(j-1)} A_c, & (3.35)\\ \beta^{(j)} &= \begin{bmatrix} \beta^{(j-1)} & \alpha_0^{(j-1)} B \end{bmatrix} \begin{bmatrix} R_0^{n+j-1,n+j} \otimes I_m \\ R_j^{n,n+j} \otimes I_m \end{bmatrix}, & (3.36) \end{aligned}$$

where matrix A_c is defined by eqn. 3.33 $(j > 0)$.

Hence, in order to evaluate the coefficients of the multi-step-ahead predictors for prediction horizons $j = 1, \ldots, P$, one can calculate the matrices of coefficients $\alpha^{(0)}$, $\beta^{(0)}$ using the algorithm given in Remark 3.1, and then calculate $\alpha^{(j)}$ and $\beta^{(j)}$ recursively using the algorithm 3.35 and 3.36 for $j = 1, \ldots, P$. This approach will be employed in the implementation of the long-range predictive self-tuning controller developed in section 5.1. The algorithm of Remark 3.2 is embedded in the macro XAB written in MATLAB commands (see Appendix D).

Example 3.1.1. Calculation of the matrices of coefficients of the polynomial matrices defining the multi-step-ahead predictor.

The matrices of coefficients of the polynomial matrices of the two-step-ahead predictor (i.e. $j = 2$) for the system described by the transfer matrix given by eqn. 2.47 will be evaluated (see example 2.1, p. 44). The corresponding DARMA model is defined by eqns. 3.1 to 3.3 with

$$A(p) = A(P_2 \otimes I_2) = [I_2\, A_1\, A_2](P_2 \otimes I_2) = \begin{bmatrix} 1 & 0 & 3 & 0 & 2 & 0 \\ 0 & 1 & 0 & 7 & 0 & 12 \end{bmatrix} (P_2 \otimes I_2), \quad (3.37)$$

$$B(p) = pB(P_1 \otimes I_2) = p[B_1\; B_2](P_1 \otimes I_2) = p \begin{bmatrix} 1 & 1 & 2 & 1 \\ 1 & 1 & 4 & 3 \end{bmatrix} (P_1 \otimes I_2), \quad (3.38)$$

i.e., $n = 2$. The delay structure is assumed to be characterized by the LNI matrix (i.e., $\overline{K}_L(q) = K_L(q)$) evaluated in example 2.1 and is given by eqn. 2.48, p. 45

$(k = 3)$. The matrix of the coefficients of the LNI matrix is given by (see eqn. 3.12, $\overline{K}_L = K_L$)

$$K_L = [K_0\ K_1\ K_2] = \begin{bmatrix} 0 & 0 & 0.5 & -0.5 & 0.5 & 0.5 \\ -0.5 & 0.5 & 0.5 & 0.5 & 0 & 0 \end{bmatrix}.$$

The matrix A_c (eqn. 3.33) is given by

$$A_c = \begin{bmatrix} -A_1 & -A_2 \\ I_2 & 0 \end{bmatrix}.$$

The recursive algorithm will be initialized with matrices of coefficients of predictor for $j = 0$.

j=0: Calculate matrix $F^{(0)}$ from eqn. 3.28 using algorithm 3.22 to calculate the inverse of matrix $\langle A\rangle_2^L$, as follows

$$E_0 = I_2, \quad E_1 = -E_0 A_1 = \begin{bmatrix} -3 & 0 \\ 0 & -7 \end{bmatrix}, \quad E_2 = -[E_0\, E_1]\begin{bmatrix} A_2 \\ A_1 \end{bmatrix} = \begin{bmatrix} 7 & 0 \\ 0 & 37 \end{bmatrix};$$

and

$$\begin{aligned} F^{(0)} &= [F_0^{(0)}\ F_1^{(0)}\ F_2^{(0)}] = K_L(\langle A\rangle_2^L)^{-1} \\ &= \begin{bmatrix} 0 & 0 & 0.5 & -0.5 & -1 & 4 \\ -0.5 & 0.5 & 2 & -3 & -5 & 15 \end{bmatrix}. \end{aligned}$$

Now matrices of coefficients of the predictor polynomials follow from eqns. 3.25 and 3.26 as

$$\alpha^{(0)} = [\alpha_0^{(0)}\ \alpha_1^{(0)}] = -F^{(0)}\langle A\rangle_2^R = \begin{bmatrix} 2 & -22 & 2 & -48 \\ 11 & -69 & 10 & -180 \end{bmatrix}; \tag{3.39}$$

$$\beta^{(0)} = [\beta_0^{(0)}\ \beta_1^{(0)}] = F^{(0)}\langle B\rangle_2^R = \begin{bmatrix} 2 & 2 & 14 & 11 \\ 2 & 3 & 50 & 40 \end{bmatrix}. \tag{3.40}$$

j=1: Matrices of coefficients of the one-step-ahead predictor polynomials follow from eqns. 3.35 and 3.36 as

$$\alpha^{(1)} = [\alpha_0^{(1)}\ \alpha_1^{(1)}] = \alpha^{(0)} A_c = \begin{bmatrix} -4 & 106 & -4 & 264 \\ -23 & 303 & -22 & 828 \end{bmatrix},$$

$$\begin{aligned}\beta^{(1)} &= [\beta_{-1}^{(1)}\ \beta_0^{(1)}\ \beta_1^{(1)}] = [\beta^{(0)}\ \alpha_0^{(0)}B]\begin{bmatrix} R_0^{2,3}\otimes I_2 \\ R_1^{2,3}\otimes I_2\end{bmatrix} \\ &= \begin{bmatrix} 2 & 2 & -6 & -9 & -84 & -64 \\ 2 & 3 & -8 & -18 & -254 & -196\end{bmatrix}.\end{aligned}$$

j=2: Matrices of coefficients of the two-step-ahead predictor polynomials follow from eqns. 3.35 and 3.36 as

$$\begin{aligned}\alpha^{(2)} &= [\alpha_0^{(2)}\ \alpha_1^{(2)}] = \alpha^{(1)}A_c = \begin{bmatrix} 8 & -478 & 8 & -1272 \\ 47 & -1293 & 46 & -3636\end{bmatrix}, \\ \beta^{(2)} &= [\beta_{-2}^{(2)}\ \beta_{-1}^{(2)}\ \beta_0^{(2)}\ \beta_1^{(2)}] = [\beta^{(1)}\ \alpha_0^{(1)}B]\begin{bmatrix} R_0^{2,3}\otimes I_2 \\ R_1^{2,3}\otimes I_2\end{bmatrix} \\ &= \begin{bmatrix} 2 & 2 & -6 & -9 & 18 & 38 & 416 & 314 \\ 2 & 3 & -8 & -18 & 26 & 84 & 1166 & 886\end{bmatrix}.\end{aligned}$$

Note that $\beta_{-2}^{(2)} = \beta_{-1}^{(1)} = \beta_0^{(0)} = M_L = \mathcal{N}_0^{(3)}$ (see eqn. C.7, p. 289).

3.1.2 The k-step-ahead predictor based on the right interactor matrix for a multi-input, multi-output, square deterministic system.

In this subsection:

- the k-step-ahead predictor based on the right interactor matrix for a MIMOS system is presented;
- an algorithm is derived for the calculation of the matrices of coefficients which represent the polynomial matrices defining the k-step-ahead predictor.

In this subsection approach (iii) in the classification on page 60 is considered. The idea of employing the (unique) right interactor matrix $\xi_R(q)$ for the development of the k-step-ahead output predictor (where k is the degree of the interactor matrix) was originally proposed in [35].

We have the following result.

Lemma 3.2 *Consider a MIMOS system satisfying Assumptions 3.1 to 3.3 (pp. 58-59). For such a system let us define an auxiliary system as follows*

$$u(t) = \tilde{K}_R(p)u_k(t), \tag{3.41}$$

where the $m \times m$ polynomial matrix $\tilde{K}_R(p)$ is given by

$$\tilde{K}_R(p) = p^k \overline{K}_R(q) = \left[\overline{K}_0 \; \overline{K}_1 \; \ldots \; \overline{K}_{k-1}\right](P_{k-1} \otimes I_m) = \overline{K}_R\,(P_{k-1} \otimes I_m), \tag{3.42}$$

and $\overline{K}_R(q)$ is the right interactor matrix of the system 3.4, and k is the degree of $\overline{K}_R(q)$. Then the k-step-ahead prediction, $y(t+k)$, of system outputs $y(t)$ is given by

$$y(t+k) = \left[\begin{array}{cc} \gamma(p) & \delta(p) \end{array}\right] \left[\begin{array}{c} y(t) \\ u_k(t) \end{array}\right]. \tag{3.43}$$

The $m \times m$ polynomial matrix $\gamma(p)$ results from the following Diophantine equation

$$I_m = \left[\begin{array}{cc} F(p) & \gamma(p) \end{array}\right] \left[\begin{array}{c} A(p) \\ p^k I_m \end{array}\right], \tag{3.44}$$

and the $m \times m$ polynomial matrix $\delta(p)$ is given by

$$\delta(p) = F(p)B(p)\overline{K}_R(q), \tag{3.45}$$

where

$$F(p) = [F_0 \; F_1 \; \ldots \; F_{k-1}](P_{k-1} \otimes I_m) = F\,(P_{k-1} \otimes I_m), \tag{3.46}$$

$$\gamma(p) = [\gamma_0 \; \gamma_1 \; \ldots \; \gamma_{n-1}](P_{n-1} \otimes I_m) = \gamma\,(P_{n-1} \otimes I_m), \tag{3.47}$$

$$\delta(p) = [\delta_0 \; \delta_1 \; \ldots \; \delta_{n+k-1}](P_{n+k-1} \otimes I_m) = \delta\,(P_{n+k-1} \otimes I_m), \tag{3.48}$$

and $\delta_0 = M_R$, where the nonsingular matrix M_R is defined by eqn. 2.41 (p. 42).

The proof of the above lemma is similar to that of [9, Theorem 5.2.4, p. 137] and is omitted.

The right interactor matrix in eqn. 3.42 is the unique right interactor matrix $\overline{K}_R(q) = \xi_R(q)$ (as in [35]) or the right nilpotent interactor (RNI) matrix $\overline{K}_R(q) = K_R(q)$ (introduced in subsection 2.3.3).

The k-step-ahead predictor given by eqn. 3.43 with the auxiliary system 3.41 was introduced in [35] using the transfer matrix plant description. The resulting predictor was employed in the development of *direct* self-tuning MPE controller. In this thesis, the k-step-ahead predictor will be employed in the development of an *indirect* self tuner which involves estimation of parameters of the DARMA model (see chapter 4). Therefore, the predictor polynomial matrices $\gamma(p)$ and $\delta(p)$ must be calculated from the DARMA model; the polynomial matrix $\gamma(p)$ is calculated by solving the Diophantine equation 3.44, and $\delta(p)$ from eqn. 3.45. This approach allows us to consider a plant which is subject to deterministic disturbance, the generator model of which is incorporated in the system model.

The *coefficient matrices* of the polynomial matrices $F(p)$ and $\gamma(p)$ can be found by equating coefficients in the polynomial equation 3.44 as in [9, Lemma 4.2.1, p. 107]. Our approach is based on equating *matrices of coefficients* of polynomials in eqn. 3.44, using the numerical representation of polynomial matrices given in Appendix B.

Using eqns. B.4 to B.6 (Appendix B), polynomial equation 3.44 can be represented by

$$R_0^{1,n+k} \otimes I_m = \begin{bmatrix} F & \gamma \end{bmatrix} \begin{bmatrix} \langle A \rangle_{k-1} \\ R_k^{n,n+k} \otimes I_m \end{bmatrix}, \tag{3.49}$$

where matrices F, γ, and A, are defined by eqns. 3.46, 3.47 and 3.2, respectively. The structure of eqn. 3.49 allows it to be separated into two equations as follows

$$\begin{aligned} R_0^{1,k} \otimes I_m &= F\langle A \rangle_{k-1}^{L}, & (3.50)\\ 0 &= F\langle A \rangle_{k-1}^{R} + \gamma, & (3.51) \end{aligned}$$

where $\langle A \rangle_{k-1}^{L}$ is a $km \times km$ upper-triangular block Toeplitz matrix given by eqn. 3.17 (for $j = 0$), and matrix $\langle A \rangle_{k-1}^{R}$ is a $km \times nm$ matrix defined by eqn. 3.18 (for $j = 0$).

The matrix of coefficients of polynomial matrix $F(p)$ follows from eqn. 3.50 as

$$F = (R_0^{1,k} \otimes I_m)((\langle A \rangle_{k-1}^{L})^{-1}, \tag{3.52}$$

in which the matrix to be inverted, given by eqn. 3.17, is nonsingular. Eqn. 3.52 can be rewritten using algorithm 3.22 for the inversion of an upper-triangular Toeplitz matrix as

$$F = [E_0 \ \dots \ E_{k-1}], \tag{3.53}$$

where E_i is defined by eqn. 3.22.

Once matrix F is found, the coefficients of the predictor polynomial $\gamma(p)$ follow from eqn. 3.51 as

$$\gamma = -F\langle A\rangle_{k-1}^{R}. \tag{3.54}$$

Matrix of coefficients of the polynomial $\delta(p)$ can be evaluated from eqn. 3.45. In general, two polynomial multiplications are required. However, if $\overline{K}_R(q) = K_R(q)$ and the algorithm for the calculation of the RNI matrix (subsection 2.3.3) is used, then the second multiplication is performed in steps 2 and 3 of the final iteration $t = i - 1$ of the algorithm for evaluation of the interactor matrix

$$\delta(p) = F(p)B(p)K_R(q) = F(p)\mathcal{B}^{(t)}(p), \tag{3.55}$$

where $\mathcal{B}^{(t)}(p) = B(p)K_R(q)$ (see eqn. 2.66, p. 48) Thus only one polynomial multiplication is required. It can be represented using eqn. B.6 as

$$\delta = F\langle \mathcal{B}^{(t)}\rangle_{k-1}, \tag{3.56}$$

where matrix δ is defined by eqn. 3.48, and $\mathcal{B}^{(t)} = \left[\mathcal{B}_0^{(t)} \ \dots \ \mathcal{B}_n^{(t)}\right]$ is the matrix of coefficients of the polynomial matrix $\mathcal{B}^{(t)}(p)$.

Let us now summarize the above results in the following remark.

Remark 3.3 *The matrices of coefficients of the polynomials of the k-step-ahead predictor 3.43 (for the auxiliary system 3.41 defined with the right nilpotent interactor matrix) are given by*

$$\begin{aligned} \gamma &= -F\langle A\rangle_{k-1}^{R}, & (3.57)\\ \delta &= F\langle \mathcal{B}^{(t)}\rangle_{k-1}, & (3.58) \end{aligned}$$

where matrix F is given by eqn. 3.53.

The macro XPRR written in MATLAB commands and implementing the algorithm 3.57 and 3.58 is given in Appendix D.

Example 3.1.2. Calculation of the matrices of coefficients of the polynomial matrices defining the k-step-ahead predictor.

The matrices of coefficients of the polynomial matrices $\gamma(p)$ and $\delta(p)$ of the k-step-ahead predictor for the system described by the transfer matrix given by eqn. 2.47 will be evaluated (see example 2.1, p. 44). The corresponding DARMA model is defined by eqn. 3.37 and 3.38 (p. 69), and $n = 2$. The delay structure is assumed to be characterized by the RNI matrix, i.e., $\overline{K}_R(q) = K_R(q)$, evaluated in example 2.2 (p. 50) and is given by eqn. 2.74 (p. 53); $k = 3$.

The matrix F is calculated from eqn. 3.53 using algorithm 3.22 as follows

$$F_0 = I_2, \quad F_1 = -F_0 A_1 = \begin{bmatrix} -3 & 0 \\ 0 & -7 \end{bmatrix}, \quad F_2 = -[F_0 \; F_1] \begin{bmatrix} A_2 \\ A_1 \end{bmatrix} = \begin{bmatrix} 7 & 0 \\ 0 & 37 \end{bmatrix}.$$

The matrix of coefficients of $\gamma(p)$ is calculated from eqn. 3.57 as follows

$$\gamma = [\gamma_0 \; \gamma_1] = -F\langle A\rangle_2^R = \begin{bmatrix} -15 & 0 & -14 & 0 \\ 0 & -175 & 0 & -444 \end{bmatrix}.$$

The matrix of coefficients of $\delta(p)$ is calculated from eqn. 3.58 using eqn. 2.73 (p. 53) as follows

$$\delta = [\delta_0 \; \delta_1 \; \delta_2 \; \delta_3 \; \delta_4] = F\langle \mathcal{B}^{(3)}\rangle_2^R$$

$$= \begin{bmatrix} 1.5811 & 0.9487 & -2.8460 & -2.8460 & 5.3759 & 6.6408 & 13.2816 & 0 & 0 & 0 \\ 1.5811 & 2.2136 & -6.6408 & -15.4952 & 27.5118 & 81.9030 & 163.806 & 0 & 0 & 0 \end{bmatrix}.$$

Note that $\delta_0 = M_R = \mathcal{B}_0^{(3)}$ and thus is nonsingular (see eqn. 2.73, p. 53).

3.1.3 Discussion of computational aspects of the multi-step-ahead and k-step-ahead predictors.

For the prediction 3.6 and 3.43 to be correct for all sample instants, the initial conditions for the predictor must be chosen appropriately. However, the initial conditions on $\{y(t)\}$ may be unknown. In order to analyze the effect on the output prediction of arbitrary initial conditions, it is convenient to interpret prediction as an output of a linear filter driven by system outputs $y(t)$ and system inputs $u(t)$. It can be shown from eqns. 3.6 and 3.43 that all poles of the transfer matrices of such filters lie at the origin of z-plane. Therefore, the predictions calculated from eqns. 3.6 and 3.43 with arbitrary initial conditions converge to the predictions which would result if the actual initial conditions were used.

Some computational aspects involved in evaluation of the polynomial matrices which define predictors 3.6 and 3.43 are discussed below. As already mentioned, the predictors will be employed in the development of self-tuning controllers. The polynomial matrices which define predictors will be evaluated at each sample instant from the new estimates of parameters of the DARMA model.

The expressions for *matrices of coefficients* of the polynomial matrices, which define predictors, were introduced in this section (see Remarks 3.1, 3.2, and 3.3). These expressions can be viewed as closed or contracted forms of iterative algorithms which yield *coefficients* of polynomial matrices instead of matrices of coefficients. For example, algorithm given in Remark 3.2 can be written in the iterative form which yields coefficients of polynomial matrices rather than matrices of coefficients. For $j > 0$ the following algorithm results from eqn. 3.35

$$\begin{aligned} \alpha_i^{(j)} &= -\alpha_i^{(j-1)} A_{i+1} + \alpha_{i+1}^{(j-1)}, \qquad i = 0, \ldots, n-2 \\ \alpha_{n-1}^{(j)} &= -\alpha_0^{(j-1)} A_n, \end{aligned} \tag{3.59}$$

and from eqn. 3.36

$$\begin{aligned} \beta_{-i}^{(j)} &= \beta_{-i+1}^{(j-1)}, \qquad i = 1, \ldots, j, \\ \beta_l^{(j)} &= \beta_{l+1}^{(j-1)} + \alpha_0^{(j-1)} B_{l+1}, \qquad l = 0, \ldots, n-2 \\ \beta_{n-1}^{(j)} &= \alpha_0^{(j-1)} B_n. \end{aligned} \tag{3.60}$$

The choice between the contracted and iterative form of the algorithm depends on the application and computational tool. If the interpretive software is used (e.g., MATLAB [110]), then multiplication and addition of matrices (of arbitrary but conformable dimensions) are the elementary operations (i.e., operations which require a single command). The update of polynomial matrices using the above iterative algorithm involves $2n$ multiplications and $2(n-1)$ additions, in comparison with three multiplications required by the algorithm given in Remark 3.2 (not counting Kronecker products, since these operations can be performed in advance). For the interpretive software, the most significant portion of the machine time is required to interpret the commands. Therefore, the contracted forms of the algorithms are likely to reduce the machine time since the amount of code is reduced. If the non-interpretive software is used (as it is usually the case for real-time applications), then the iterative form of the algorithms is appropriate, since the unnecessary operations (such as multiplication by null elements implicit in matrix A_c, eqn. 3.33) are eliminated. The iterative algorithms can be readily obtained from the contracted forms. The expressions given in Remarks 3.1 and 3.3 can be written in an iterative form in a similar way to that presented above for the algorithm of Remark 3.2. For example, eqns. 3.53 and 3.54 are equivalent to eqns. 4.2.9 and 4.2.10 in [9, p. 108] for the coefficients of polynomial matrices $F(p)$ and $\gamma(p)$ in eqn. 3.44.

3.2 The k-step-ahead predictor for one output of a single-input, multi-output, deterministic system having the feedback configuration FD.

In this section:

- the k-step-ahead predictor for one output is derived for a single-input, multi-output, deterministic system which has the feedback configuration FD and is described by the RDO representation;
- the relationship between solutions to the k-step-ahead prediction problem based on the RDO and state-space system representations is established;
- the corresponding adaptive k-step-ahead predictor is introduced.

The problem of prediction of one output of a deterministic system having the feedback configuration FD (see p. 18) is addressed in this section. A special case of such a system is considered in the form of a single-input, multi-output (SIMO) system. A new predictor for the future value of the output $y(t)$ is developed which utilizes available system outputs $y_F(t)$. The output $y(t)$ may or may not be an element of $y_F(t)$. Such a predictor will be referred to as the SIMO-type output predictor. The SIMO-type predictor will be employed in the development of the MPE direct self-tuning controllers for SIMO systems in section 6.1.

The feedback configuration FD implies that there are some *additional system outputs* which can be utilized in the predictor. Although the idea of utilization of additional system variables for the purpose of prediction is not new (see e.g., [9, pp. 232-234]), the problem formulation and its solution is, to the author's knowledge, original.

On the other hand, conventional predictors have been developed for systems having the feedback configuration FI, i.e., $y_F(t) \equiv y(t)$ (see p. 18). The conventional predictor utilizes only the output the future value of which is to be predicted. The problem of prediction of one output of a SIMO system having the feedback configuration FD is considered in this section; therefore, the corresponding problem for a system having the feedback configuration FI involves development of the predictor for a single-input, single-output (SISO) system. The predictor for a SISO system will be referred to as the SISO-type predictor.

The following assumptions are made.

Assumption 3.4 (Feedback configuration assumption) *Given is a single-input, multi-output (SIMO), deterministic system having the feedback configuration FD, such that*

$$\begin{aligned} l &= \dim u(t) = 1, \\ m &= \dim y(t) = 1, \\ f &= \dim y_F(t) \geq 1. \end{aligned}$$

The state-space model of a system having the feedback configuration FD is given by eqn. 2.3, p. 20.

Assumption 3.5 (System assumption) *The system* (A_s, B_s, C_F) *is controllable and observable; the observability index* [6] *is* ν_F.

The RDO representation (in the backward shift operator) of a system satisfying Assumptions 3.4 and 3.5 follows from eqn. 2.22 (p. 28)

$$\begin{bmatrix} u(t) \\ y(t) \\ y_F(t) \end{bmatrix} = \begin{bmatrix} d(p) \\ N(p) \\ N_F(p) \end{bmatrix} x_R(t), \tag{3.61}$$

with initial conditions on $\{x_R(t)\}$. Polynomials $d(p)$ and $N(p)$, and $f \times 1$ polynomial vector $N_F(p)$ are expressed using the polynomial representation of Appendix B (see eqns. B.1 to B.3) as follows

$$\begin{aligned} d(p) &= p^n \det(qI - A_s) = p^n \hat{D}(q) = [1\ d_1\ \ldots\ d_n]P_n = dP_n, && (3.62) \\ N(p) &= p^n \hat{N}(q) = [0\ \ldots\ 0\ n_k\ \ldots\ n_n]P_n = NP_n, && (3.63) \\ N_F(p) &= p^n \hat{N}_F(q) = [0\ N_1\ \ldots\ N_n]P_n = N_F P_n, && (3.64) \end{aligned}$$

where $P_n = [1\ p\ \ldots\ p^n]'$. In eqn. 3.63 it is assumed that $n_k \neq 0$, i.e., k represents the delay from the system input $u(t)$ to output $y(t)$. The pair $(N_F(p),\ d(p))$ is right coprime since the pair $(C_F,\ A_s)$ is observable.

3.2.1 The k-step-ahead SIMO-type predictor for a system with known parameters.

We have the following result.

Lemma 3.3 *Consider a SIMO system satisfying Assumptions 3.4 and 3.5. Then there exist a* $1 \times f$ *polynomial vector* $\alpha(p)$ *and polynomial* $\beta(p)$ *of degree* $n_p \geq \nu_F - 1$, *defined as*

$$\begin{aligned} \alpha(p) &= [\alpha_0\ \alpha_1\ \ldots\ \alpha_{n_p}]\left(P_{n_p} \otimes I_f\right) = \alpha\left(P_{n_p} \otimes I_f\right), && (3.65) \\ \beta(p) &= [\beta_0\ \beta_1\ \ldots\ \beta_{n_p}]P_{n_p} = \beta P_{n_p}, && (3.66) \end{aligned}$$

[6] See [80, p. 357] for the definition of the observability index.

where α_i is of dimension $1 \times f$ for $i = 0, \ldots, n_p$, and α is a $1 \times (n_p + 1)f$ vector, and β is a $1 \times (n_p + 1)$ vector. These polynomials satisfy the following Diophantine equation

$$p^{-k}N(p) = \begin{bmatrix} \alpha(p) & \beta(p) \end{bmatrix} \begin{bmatrix} N_F(p) \\ d(p) \end{bmatrix}. \tag{3.67}$$

The output $y(t+k)$ is given by the following predictor expression

$$y(t+k) = \begin{bmatrix} \alpha(p) & \beta(p) \end{bmatrix} \begin{bmatrix} y_F(t) \\ u(t) \end{bmatrix}. \tag{3.68}$$

Proof.

From eqn. 3.61 one has $y(t+k) = p^{-k}N(p)x_R(t)$. Using eqn. 3.67

$$y(t+k) = \alpha(p)N_F(p)x_R(t) + \beta(p)d(p)x_R(t),$$

and eqn. 3.68 follows using eqn. 3.61. The existence of polynomials $\alpha(p)$ and $\beta(p)$ of degree $n_p \geq \nu_F - 1$ and satisfying Diophantine equation 3.67 follows from [112, Theorem 9-12, p. 466] since the pair $(N_F(p),\, d(p))$ is right coprime.

Q.E.D.

The predictor 3.68 will be referred to as the *k-step-ahead SIMO-type predictor.*

In order to solve the Diophantine equation 3.67 for $\alpha(p)$ and $\beta(p)$, its numerical representation is considered. This translates the problem of solving a polynomial equation into a problem of solving a set of linear algebraic equations. Such an approach to solving the Diophantine equations was considered for example in [107,112].

Using the method of resultants, the polynomial equation 3.67 can be represented by the following set of linear algebraic equations (see eqns. B.4 to B.6, Appendix B)

$$NS^kR_0^{n+1,n+n_p+1} = \begin{bmatrix} \alpha & \beta \end{bmatrix} \begin{bmatrix} \langle N_F \rangle_{n_p} \\ \langle d \rangle_{n_p} \end{bmatrix}, \tag{3.69}$$

where S^k is the k-th power of the $(n+1) \times (n+1)$ shift matrix defined as

$$S = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 1 & 0 & \ddots & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & \dots & 1 & 0 \end{bmatrix}. \qquad (3.70)$$

Eqn. 3.69 is of the form $b = x\mathcal{A}$ where matrix $\mathcal{A}$ is of dimension $(f+1)(n_p+1) \times (n+n_p+1)$. The existence of a solution x for *any* b is guaranteed by coprimeness of the pair $(N_F(p)\ d(p))$ and the choice of $n_p \geq \nu_F - 1$ [112, Theorem 9-12, p. 466] [7]. For a particular $b = NS^kR_0^{n+1,n+n_p+1}$, however, the solution x may exists for a degree smaller than $\nu_F - 1$ (see example 3.2.1, p. 83) [8].

Furthermore,

Comment 3.1 *It follows from eqn. 3.69 that $\beta_0 = n_k \neq 0$ (n_k is defined in eqn. 3.63).*

Comparison of predictors for systems having the feedback configuration FD and FI.

It is desirable to compare the properties of the SIMO-type predictor developed for a system having the feedback configuration FD with those of the SISO-type predictor for the same system within the feedback configuration FI.

The RDO model of the system having the feedback configuration FI is given by eqn. 3.61 with $y_F(t) = y(t)$ $(f = 1)$. Then the SISO-type predictor follows from Lemma 3.3 and is given by the following expression [9]

$$y(t+k) = \begin{bmatrix} \alpha_s(p) & \beta_s(p) \end{bmatrix} \begin{bmatrix} y(t) \\ u(t) \end{bmatrix}, \qquad (3.71)$$

[7] Recall that given a $m \times n$ matrix $\mathcal{A}$, for every $1 \times n$ vector b, there exists a $1 \times m$ vector x such that $b = x\mathcal{A}$ if and only if $\operatorname{rank}\mathcal{A} = n$ [112].

[8] Recall that given a $m \times n$ matrix $\mathcal{A}$ and given a $1 \times n$ vector b, there exists a $1 \times m$ vector x such that $b = x\mathcal{A}$ if and only if $\operatorname{rank}\mathcal{A} = \operatorname{rank}\begin{bmatrix} \mathcal{A} \\ b \end{bmatrix}$ [112].

[9] Alternatively one can develop the SISO-type predictor from the system DARMA model, see [9, Lemma 4.2.1, p. 107].

where the polynomials $\alpha_s(p)$ and $\beta_s(p)$ have degree $n_p \geq \nu_F - 1 = n - 1$ and satisfy Diophantine equation 3.67 for $N_F(p) = N(p)$.

The SISO-type predictor involves measurement of the output $y(t)$ the future value of which is to be predicted, whereas the SIMO-type predictor does not require measurement of $y(t)$ [10]. Hence the SIMO-type predictor offers flexibility of the output measurement configuration; this becomes especially important if the measurement of the output $y(t)$ is costly or inaccurate but some other outputs $y_F(t)$ can be measured at a lower cost or more accurately.

Comment 3.2 *Note that if the degree of polynomials of the SISO- and SIMO-type predictors is chosen as $n_p = \nu_F - 1$, then the degree of the SIMO-type predictor polynomials is smaller than that of the polynomials of the SISO-type predictor provided $f = \operatorname{rank} C_F > 1$ (recall that $\nu_F \leq n - \operatorname{rank} C_F + 1$ [112, p. 199]).*

The number of coefficients of polynomials of the SIMO-type predictor may be smaller than that of the polynomials of the SISO-type predictor (see example 3.2.1). For example, if $f = \operatorname{rank} C_F = n$ then $n_p = 0$ and the number of coefficients of the SIMO-type predictor polynomials is $n + 1$ in comparison with $2n$ for the SISO-type predictor.

Finally, note that if it is required that the output prediction equals to the system output for all sample instants, then the appropriate initial conditions for the predictor must be considered. If, however, arbitrary initial conditions are assumed, then the effect of incorrect initial conditions on the output prediction dies away after n_p sample instants (cf. discussion for MIMOS systems, p. 75).

[10] Possible benefits resulting from the separation of the output $y(t)$ from the outputs $y_F(t)$ are illustrated in example 6.1.2 (p. 210) for the application of the SIMO-type predictor in self-tuning control.

Example 3.2.1.

Let us consider a single-input, two-output system given by the state-space model 2.3 (p. 20) in the controller form [11] [80, p. 434]

$$A_s = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad B_s = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad C_F = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{bmatrix},$$

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \end{bmatrix}, \tag{3.72}$$

where $a_1 = -2.7124$, $a_2 = 2.6636$, $a_3 = -0.9512$, $c_{11} = 0.0008128$, $c_{12} = 0.0031708$, $c_{13} = 0.0007927$, $c_{21} = c_{12}$, $c_{22} = -0.0099728$, and $c_{23} = -0.001377$. The observability index is $\nu_F = 2$. The corresponding RDO representation is given by eqns. 3.61 to 3.64 with $d(p) = dP_3 = [1\ a_1\ a_2\ a_3]\,P_3$, $N_F(p) = N_F P_3 = [0\ \ C_F]\,P_3$, and $N(p) = NP_3 = [0\ \ C]P_3$, i.e., $n = 3$ and $k = 1$.

Let us now determine the $(k = 1)$-step-ahead SIMO-type predictor for the output $y(t) = y_1(t) = Cx(t)$ in terms of outputs $y_F(t) = [y_1(t)\, y_2(t)]' = C_F x(t)$ $(f = 2)$.

Choosing $n_p = \nu_F - 1 = 1$, and solving eqn. 3.69, we have [12] $\alpha_0 = [\alpha_{10}\ 0]$ and $\alpha_1 = [\alpha_{11}\ 0]$. This suggests that a solution may exists for $n_p = 0 < \nu_F - 1$. In this case one has from eqn. 3.69

$$NS = \begin{bmatrix} \alpha_{10} & \alpha_{20} & \beta_0 \end{bmatrix} \begin{bmatrix} N_F \\ d \end{bmatrix}.$$

The above equation is of the form $b = x\mathcal{A}$ and is found to be consistent; a general solution is $x = b\mathcal{A}^\dagger + P(I - \mathcal{A}\mathcal{A}^\dagger)$, where $\mathcal{A}^\dagger$ is the generalized inverse of matrix $\mathcal{A}$, and P is a parameter of the solution [122, chapter 2]. Since the matrix $\mathcal{A}$ has full row rank the solution is unique [112] and is given by

$$\begin{bmatrix} \alpha_{10} & \alpha_{20} & \beta_0 \end{bmatrix} = \begin{bmatrix} 2.7124 & 1 & 0.0008128 \end{bmatrix}. \tag{3.73}$$

[11] There is no loss of generality in assuming that the system is given in the controller form since the pair $(A_s,\ B_s)$ is said to be controllable (see Assumption 3.5, p. 79).

[12] Recall that $\alpha(p)$ is a $1 \times f$ polynomial vector having coefficient vectors α_j for $j = 0, \ldots, n_p$ of dimension $1 \times f$ (see eqn. 3.65); each coefficient vector α_j has f elements α_{ij} (α_{ij} is associated with the system output $y_i(t)$).

If the only output which can be measured is the output the future value of which is to be predicted (i.e., the system has the feedback configuration FI) then the SISO-type predictor 3.71 is employed. The matrices of coefficients of polynomials $\alpha_s(p)$ and $\beta_s(p)$, calculated from eqn. 3.69 for $n_p = n-1 = 2$ and $N_F = N$, are given by

$$\alpha_s = [\alpha_{s0}\ \alpha_{s1}\ \alpha_{s2}] = [-a_1\ \ -a_2\ \ -a_3], \tag{3.74}$$

$$\beta_s = [\beta_{s0}\ \beta_{s1}\ \beta_{s2}] = [c_{11}\ c_{12}\ c_{13}]. \tag{3.75}$$

Note that for the SIMO-type predictor, the polynomials are of degree zero and involve 3 coefficients; the SISO-type predictor polynomials are of second degree and involve 6 coefficients.

Relationship between solutions to the k-step-ahead SIMO-type prediction problem based on the RDO and state-space system representations.

The predictor polynomials $\alpha(p)$ and $\beta(p)$ can be found simultaneously from the Diophantine equation 3.67 which is based on the system RDO representation. It is shown below that coefficients of the polynomial $\alpha(p)$ can be evaluated independently of those of $\beta(p)$ in terms of the system state-space model.

Remark 3.4 *Consider a SIMO system satisfying Assumptions 3.4 and 3.5. Then for the predictor 3.68 and for the system state-space model 2.3 (p. 20)*

(i) the $1 \times (n_p+1)f$ vector α of coefficients of the polynomial vector $\alpha(p)$ satisfies

$$\alpha\vartheta = CA_s^{k+n_p}, \tag{3.76}$$

where

$$\vartheta = \begin{bmatrix} C_F A_s^{n_p} \\ \vdots \\ C_F \end{bmatrix}; \tag{3.77}$$

(ii) the coefficients of the polynomial $\beta(p)$ are given by

$$\beta_0 = CA_s^{k-1}B_s,$$

$$\beta_i \;=\; CA_s^{k-1+i}B_s - \begin{bmatrix} \alpha_0 & \dots & \alpha_{i-1} \end{bmatrix} \begin{bmatrix} C_F A_s^{i-1} B_s \\ \vdots \\ C_F B_s \end{bmatrix}, \tag{3.78}$$

for $i = 1, \dots, n_p$.

Proof.

Considering the property of Markov parameters for a system with delay k one has $CA_s^{j-1}B_s = 0$ for $j = 1, \dots, k-1$ [80]. Then the output $y(t)$ at time $t + k + n_p$ is given in terms of the system state-space model 2.3 by

$$y(t+k+n_p) = CA_s^{k+n_p}x(t) + \sum_{i=0}^{n_p} CA_s^{k+n_p-1-i}B_s u(t+i). \tag{3.79}$$

Alternatively, from the predictor expression 3.68 one has

$$y(t+k+n_p) = \alpha(p)y_F(t+n_p) + \beta(p)u(t+n_p),$$

and substituting for $y_F(t+n_p)$ form the state-space model 2.3

$$y(t+k+n_p) = \sum_{i=0}^{n_p} \alpha_i C_F A_s^{n_p-i}x(t) +$$
$$+ \sum_{j=0}^{n_p-1} \alpha_j [\sum_{i=0}^{n_p-1-j} C_F A_s^{n_p-j-1-i} B_s u(t+i)] + \sum_{i=0}^{n_p} \beta_{n_p-i} u(t+i). \tag{3.80}$$

Eqns. 3.76 and 3.78 follow form eqns. 3.79 and 3.80 by comparison of terms associated with $x(t)$ and $u(t+i)$ for $i = 0, 1, \dots, n_p$.

Q.E.D.

Note that vector α can be evaluated from eqn. 3.76 independently of β. Furthermore, since the pair $(C_F,\ A_s)$ is observable the existence of solution to eqn. 3.76 is guaranteed by the choice $n_p \geq \nu_F - 1$ which implies that matrix ϑ has full (column) rank. If $n_p = \nu_F - 1$, then ϑ is the system *partial observability matrix* [80, p. 357].

3.2.2 Adaptive k-step-ahead SIMO-type predictor.

In subsection 3.2.1 the k-step-ahead SIMO-type predictor was introduced for systems with known parameters. If the system parameters are unknown, then the adaptive predictor can be employed. Such predictors, for systems having the feedback configuration FI, are discussed for instance in [9, chapter 4] for deterministic systems and in [123,124,125,126] for stochastic systems; examples of applications of adaptive predictors are given in [117,118,127].

We will make use of the separation of estimation and prediction to develop the SIMO-type k-step-ahead *direct* adaptive predictor. For this purpose let us rewrite the predictor expression 3.68 as

$$y(t) = \Theta'\phi(t-k), \tag{3.81}$$

where the $(f+1)(n_p+1) \times 1$ parameter vector Θ and regression vector $\phi(t)$ are defined as

$$\begin{aligned} \Theta &= [\alpha \;\; \beta]', && (3.82) \\ \phi(t) &= \begin{bmatrix} y_F(t)' & \ldots & y_F(t-n_p)' & u(t) & \ldots & u(t-n_p) \end{bmatrix}'. && (3.83) \end{aligned}$$

The vector of parameters Θ in model 3.81 is estimated on-line by a recursive parameter estimation algorithm. This involves calculation of the equation error (see eqn. E.1, Appendix E), and therefore knowledge (i.e., measurement) of $y(t)$ is required. Hence, the following assumption is made (cf. discussion on pp. 34-35)

Assumption 3.6 (Output measurement configuration assumption) *For the purpose of estimation of predictor parameters, the present value $y(t)$ of the output which is to be predicted is known.*

Furthermore, it is assumed that the system delay k is known and that $n_p \geq \nu_F - 1$ is chosen. Then the direct *adaptive SIMO-type predictor* consists of the following steps which are performed at every sample instant t:

step 1. calculate the predictor parameter estimates

$$\widehat{\Theta}(t) = \left[\widehat{\alpha}(t) \;\; \widehat{\beta}(t)\right]' \tag{3.84}$$

using, say, the RLS algorithm (see eqns. E.1 to E.4, Appendix E), with the equation error defined as

$$e(t) = y(t) - \widehat{\Theta}(t-1)'\phi(t-k); \tag{3.85}$$

step 2. calculate the prediction of $y(t)$ as follows

$$\widehat{y}(t) = \widehat{\Theta}(t-k)'\phi(t-k). \tag{3.86}$$

We have the following convergence result.

Lemma 3.4 *Let the output sequence $\{y(t)\}$ be generated by the system satisfying Assumptions 3.4, 3.5, and 3.6. If the sequences $\{u(t)\}$ and $\{y_F(t)\}$ are bounded, the delay k is known, and $n_p \geq \nu_F - 1$, then for the SIMO-type adaptive RLS prediction $\widehat{y}(t)$ resulting from eqn. 3.86 one has*

$$\lim_{t\to\infty}[y(t) - \widehat{y}(t)] = 0.$$

(It is assumed that the RLS estimator with no forgetting is employed.)

Proof.

The existence of the predictor 3.81 follows from Lemma 3.3, p. 79. Then the proof of [9, Lemma 4.3.1, p. 112] applies *mutatis mutandis* to Lemma 3.4.

Q.E.D.

The above lemma shows that the adaptive output prediction $\widehat{y}(t)$ converges to the actual value of the system output $y(t)$. No persistent excitation condition is required for the input sequence $\{u(t)\}$. However, nothing is said about the convergence of estimates $\widehat{\alpha}(t)$ and $\widehat{\beta}(t)$ to their true values [9].

Implementation of the adaptive predictor requires selection of n_p based on the observability index ν_F. If the observability index is unknown then the choice of n_p can be based on the upper bound on ν_F, i.e., $n_p = n - \operatorname{rank} C_F$. Alternatively, the methods for the on-line identification of the order of the estimated system model can be applied to search for n_p [128]. The recursive method proposed in [129] involves a model order testing criterion based on the sum of squares of the prediction error $y(t) - \widehat{y}(t)$ and is amenable to predictor applications since the prediction error can be readily obtained.

There are two properties of the SIMO-type predictor which are especially important in its adaptive implementation. Firstly, the degree n_p of predictor polynomials $\alpha(p)$ and $\beta(p)$ is likely to be ***smaller*** than that of the SISO-type predictor polynomials. This implies that ***more recent data*** is used in the regression vector $\phi(t)$ of the estimator which is especially desirable if the system is time-varying. Secondly, the number of estimated parameters is likely to be ***smaller*** for the SIMO-type predictor than for the SISO-type predictor provided sufficient number of outputs $y_F(t)$ is available. This improves the ***convergence rate*** of the estimates (see example 3.2.2).

Example 3.2.2.

Let us consider a single-input, two-output system given by the state-space model 3.72 (p. 83) in example 3.2.1. This is a simplified linearized model of a single joint of the robot manipulator POLAR 6000 of COMAU S.p.A. [130]. The input $u(t)$ represents servovalve excitation current of the hydraulic motor [mA]; the output $y(t)$ is the angular position of the robot arm in the horizontal plane [rad]. [13] The system delay is $k = 1$.

The SIMO- and SISO-type adaptive ($k = 1$)-step-ahead predictors for the angular position of the robot arm were simulated to compare the convergence rate of the estimates of coefficients of predictor polynomials. The true values of those coefficients are given by eqns. 3.73 to 3.75, p. 83.

The input was a pseudo-random binary sequence (PRBS) [131] generated by 10 stage shift register and of amplitude 50 [mA]. The initial parameter estimates of the RLS algorithm were all set to zero; no forgetting was used, and the initial covariance matrix was $P(-1) = 10I$ (see Appendix E).

The coefficient estimates are shown in fig. 3.1. It can be seen that the convergence rate of the estimates of the SIMO-type predictor was faster during the initial phase of estimation than that of the SISO-type predictor, i.e., the estimates of the SIMO-

[13] The values of the model parameters assumed in example 3.2.1 correspond to the following set of parameters of the second order (plus integrator) continuous time transfer function [130]: $K = 2$, $\omega = 50$ rad s^{-1}, $\zeta = 0.05$; the sampling time is $T_s = 0.01$ s.

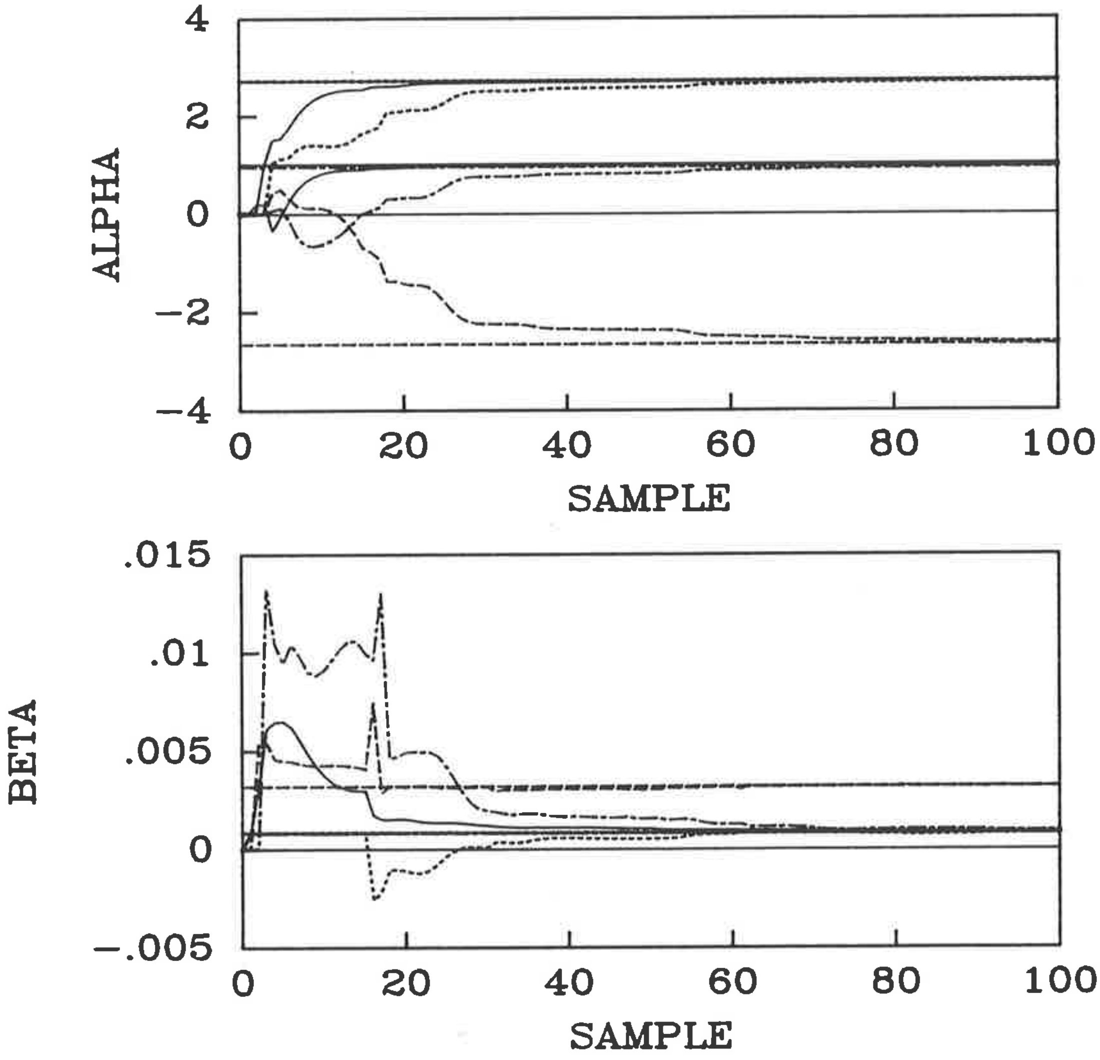

Figure 3.1: Estimates and true values of parameters α and β of the adaptive SIMO-type predictor (solid lines), and α_s and β_s of the SISO-type predictor (broken lines) for the robot manipulator.

type predictor converged closely to the true values more rapidly than those of the SISO-type predictor. This resulted in the performance improvement of the SIMO-type predictor in comparison with that of the SISO-type predictor. As a measure of performance of adaptive predictors, the cumulative sum of squared prediction errors (CSSPE) was calculated as

$$\mathrm{CSSPE}(t) = \sum_{\tau=1}^{t} [y(\tau) - \hat{y}(\tau)]^2$$

for $t = 1, \ldots, 100$. The CSSPE is depicted in fig. 3.2. It may be observed that the CSSPE achieves a constant level more rapidly for the SIMO-type predictor and its

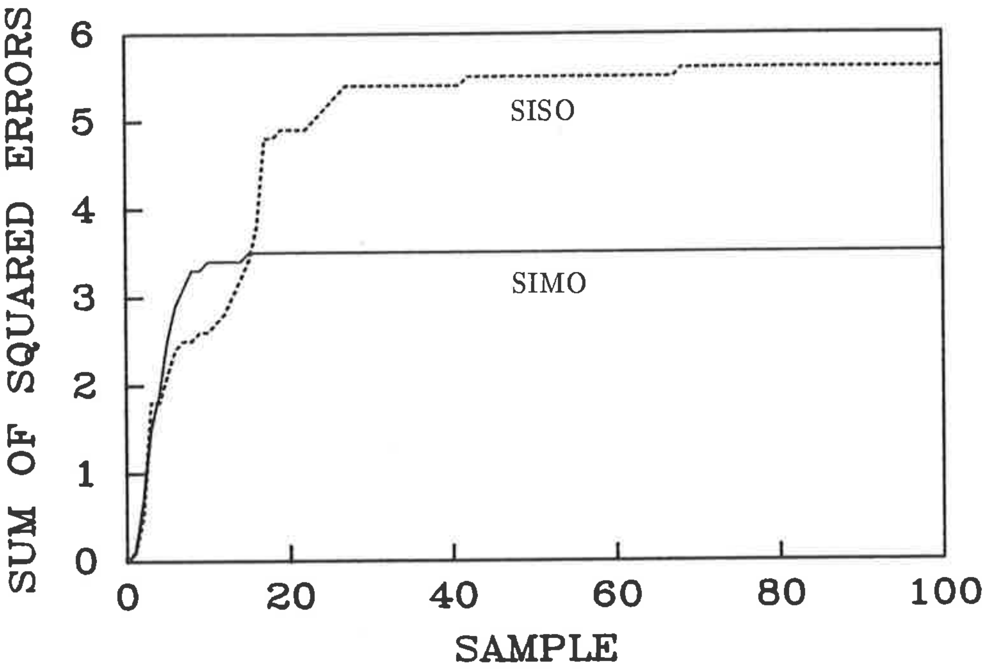

Figure 3.2: The cumulative sum of squared prediction errors for the adaptive SIMO-type predictor (solid line) and SISO-type predictor (broken line) for the robot manipulator.

final value is smaller than that for the SISO-type predictor.

3.3 Optimal predictor for one output of a two-input, multi-output, stochastic system having the feedback configuration FD.

In this section:

- the optimal predictor for one output is derived for a two-input, multi-output, stochastic system which has the feedback configuration FD and is described by the RDO representation;
- the relationship between solutions to the optimal prediction problem based on the RDO and state-space system representations is established;
- the corresponding adaptive predictor is introduced.

The problem of optimal prediction of one output of a multi-output stochastic system having the feedback configuration FD (see p. 18) is addressed in this section. A special case of such a system is considered. Namely, it is assumed that there is one measurable input $u(t)$ and one unmeasurable stochastic input $\omega(t)$. Therefore, we shall refer to such a system as the two-input, multi-output (TIMO), stochastic system. A new optimal predictor for the future value of the output $y(t)$ is developed which utilizes available system outputs $y_F(t)$. The output $y(t)$ may or may not be an element of $y_F(t)$. Such a predictor will be referred to as the TIMO-type output predictor. The TIMO-type predictor will be employed in the development of the MPE self tuner for TIMO systems in section 6.2.

The formulation of the optimal prediction problem for systems having the feedback configuration FD and its solution presented in this section is, to the author's knowledge, original.

The feedback configuration FD implies that there are some *additional system outputs* which can be utilized in the predictor. On the other hand, conventional predictors have been developed for systems having the feedback configuration FI, i.e., $y_F(t) \equiv y(t)$. The conventional predictor utilizes only the output the future value of which is to be predicted. The problem of optimal prediction of one output of a TIMO system having the feedback configuration FD is considered in this section; therefore, the corresponding problem for a system having the feedback configuration FI involves development of the predictor for a two-input ($u(t)$, $\omega(t)$), single-output (TISO) stochastic system. The predictor for a TISO system will be referred to as the TISO-type predictor.

The following assumptions are made.

Assumption 3.7 (Feedback configuration assumption) *Given is a two-input, multi-output (TIMO), stochastic system having the feedback configuration FD, such that*

$$\begin{aligned} l &= \dim u(t) = 1, \\ s &= \dim \omega(t) = 1, \end{aligned}$$

$$\begin{aligned} m &= \dim y(t) = 1, \\ f &= \dim y_F(t) > 1. \end{aligned}$$

The state-space model of a stochastic system having the feedback configuration FD is given by eqn. 2.8 (p. 22).

Assumption 3.8 (System assumption) *The system (A_s, B_s, C_F) is controllable and observable. Furthermore, the pair $(qI - A_s, K)$ is left coprime* [14].

Let us now define the following $(n_p + 1)f \times (n + n_p + 1)$ matrix in terms of the system state-space model

$$\vartheta_\omega = \begin{bmatrix} C_S & C_F K & C_F A_s K & \dots & C_F A_s^{n_p-1} K & C_F A_s^{n_p} \\ 0 & C_S & C_F K & \dots & C_F A_s^{n_p-2} K & C_F A_s^{n_p-1} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & C_S & C_F K & C_F A_s \\ 0 & \dots & 0 & \dots & C_S & C_F \end{bmatrix}, \tag{3.87}$$

where integer $n_p > 0$ (if $n_p = 0$ then $\vartheta_\omega = [C_S \; C_F]$).

Assumption 3.9 *The matrix ϑ_ω has full column rank, i.e.,*

$$\operatorname{rank} \vartheta_\omega = n + n_p + 1. \tag{3.88}$$

The RDO representation (in the backward shift operator) of a system satisfying Assumptions 3.7 and 3.8 follows from eqn. 2.31 (p. 30)

$$\begin{bmatrix} u(t) \\ \omega(t) \\ y(t) \\ y_F(t) \end{bmatrix} = \begin{bmatrix} d(p) & 0 \\ 0 & d(p) \\ N(p) & c_\omega(p) \\ N_F(p) & C_\omega(p) \end{bmatrix} \begin{bmatrix} x_R(t) \\ x_{R\omega}(t) \end{bmatrix}, \tag{3.89}$$

with initial conditions on $\{[x_R(t) \; x_{R\omega}(t)]'\}$. Polynomials $d(p)$, $N(p)$, $c_\omega(p)$, and $f \times 1$ polynomial vectors $N_F(p)$ and $C_\omega(p)$ are expressed using the polynomial representation of Appendix B (see eqns. B.1 to B.3) as follows

$$d(p) \;=\; p^n \det(qI - A_s) = p^n \hat{D}(q) = p^n \hat{D}_\omega(q) = [1 \; d_1 \; \dots \; d_n] P_n = d P_n (3.90)$$

[14] This guarantees that eqn. 2.28 on page 30 holds.

$$
\begin{aligned}
N(p) &= p^n \hat{N}(q) = [0 \; \dots \; 0 \; n_k \; \dots \; n_n] P_n = N P_n, & (3.91)\\
c_\omega(p) &= p^n \hat{c}_\omega(q) = [1 \; c_1 \; \dots \; c_n] P_n = c_\omega P_n, & (3.92)\\
N_F(p) &= p^n \hat{N}_F(q) = [0 \; N_1 \; \dots \; N_n] P_n = N_F P_n, & (3.93)\\
C_\omega(p) &= p^n \hat{C}_\omega(q) = [C_0 \; C_1 \; \dots \; C_n] P_n = C_\omega P_n, & (3.94)
\end{aligned}
$$

where $P_n = [1 \; p \; \dots \; p^n]'$. In eqn. 3.91 it is assumed that $n_k \neq 0$, i.e., k represents the delay from the system input $u(t)$ to output $y(t)$.

For the purpose of analysis presented in this section and in section 6.2, the assumptions about the noise sequence given by eqns. 2.4 and 2.5 on page 21 are formalized in the following way (see [132,133,125,55,57,56,134,9] for similar assumptions).

Assumption 3.10 (Noise assumption) *Let the (scalar) sequence $\{\omega(t)\}$ be a real stochastic process defined in a probability space $(\Omega, \mathcal{F}, P)$ and adapted to the sequence of increasing sub-sigma algebras $(\mathcal{F}_t, t \in N)$. The sequence $\mathcal{F}_t$ is generated by the observations $\{y_F(t), \dots, y_F(0)\}$ and $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \dots \subset \mathcal{F}_t$. $\mathcal{F}_0$ includes initial condition information. Furthermore,*

$$
\begin{aligned}
E\{\omega(t)|\mathcal{F}_{t-1}\} &= 0 \quad a.s., \quad t \geq 1; & (3.95)\\
E\{\omega(t)^2|\mathcal{F}_{t-1}\} &= \sigma^2 \quad a.s., \quad t \geq 1. & (3.96)
\end{aligned}
$$

(Symbol "*a.s.*" means "almost surely", i.e., save on a set having probability measure zero. The sequence $\{\omega(t)\}$ can be thought of as a white noise sequence [9, p. 324].)

3.3.1 Optimal TIMO-type predictor based on the right difference operator system representation.

In this subsection the TIMO-type predictor for an arbitrary prediction horizon $k_p \geq 1$ is developed. Depending on the application of the TIMO-type predictor to predictive control, three relations between the prediction horizon k_p and delay k are of particular interest. Firstly, if $k_p > k$ then the predictor is referred to as the multi-step-ahead predictor [15]. Such a predictor can be employed in the development of

[15] The practical applications of various variants of the multi-step-ahead predictors are given in [118,127,130,38,135] for systems having the feedback configuration FI. For instance, a multi-step-ahead

a long-range predictive control law [16]. Secondly, the ($k_p = k$)-step-ahead predictor will be employed in the development of the MPE controller in section 6.2. Thirdly, the ($k_p = 1$)-step-ahead predictor for systems with delay $k > 1$ will be employed in convergence analysis of the self-tuning MPE controller for TIMO systems (see Theorem 6.2 (p. 230) and Comment F.4, p. 309).

Optimal TIMO-type predictor.

In the derivation of the optimal TIMO-type predictor it is convenient to consider two cases $k_p \geq k$ and $k_p < k$ separately. We have the following result.

Lemma 3.5 *Consider a TIMO system satisfying Assumptions 3.7, 3.8, 3.9, and 3.10. Denote $k_i = k - k_p$ for $k_p \geq k$ and $k_i = 1$ for $k_p < k$, where $k_p \geq 1$ is the prediction horizon. Then there exist a $1 \times f$ polynomial vector $\alpha_\omega(p)$ and polynomial $\beta_\omega(p)$, defined as*

$$\begin{aligned} \alpha_\omega(p) &= \begin{bmatrix}\alpha_0 & \alpha_1 & \dots & \alpha_{n_p}\end{bmatrix}\left(P_{n_p} \otimes I_f\right) = \alpha_\omega\left(P_{n_p} \otimes I_f\right), & (3.97)\\ \beta_\omega(p) &= \begin{bmatrix}\beta_{k_i} & \beta_{k_i+1} & \dots & \beta_{n_p}\end{bmatrix} P_{n_p-k_i} = \beta_\omega P_{n_p-k_i}, & (3.98)\end{aligned}$$

where α_i is of dimension $1 \times f$ for $i = 0, \dots, n_p$, and α_ω is a $1 \times (n_p+1)f$ vector, and β_ω is a $1 \times (n_p - k_i + 1)$ vector. These polynomials satisfy the following composite Diophantine equation

$$p^{-k_p}\begin{bmatrix} N(p) & c_\omega(p)\end{bmatrix} = \begin{bmatrix}\alpha_\omega(p) & \beta_\omega(p) & F(p)\end{bmatrix}\begin{bmatrix} N_F(p) & C_\omega(p) \\ p^{k_i}d(p) & 0 \\ 0 & p^{-k_p}d(p)\end{bmatrix}, \qquad (3.99)$$

where the unique polynomial $F(p)$ has degree $k_p - 1$

$$F(p) = \begin{bmatrix} f_0 & f_1 & \dots & f_{k_p-1}\end{bmatrix} P_{k_p-1} = FP_{k_p-1}; \quad f_0 = 1. \qquad (3.100)$$

predictor is used in the area of decision making, e.g., an operator-based control of a blast-furnace with the multi-step-ahead predictor indicating if the planned control policy will be favourable or not [118].

[16] The long-range predictive controller for systems having the feedback configuration FD is not considered in this thesis. However, the long-range predictive control law is derived for systems having the feedback configuration FI in section 5.2 using the corresponding multi-step-ahead predictor.

The optimal k_p-step-ahead prediction $y^o(t+k_p|t)$ of output $y(t)$ is given by

$$y^o(t+k_p|t) = \begin{bmatrix} \alpha_\omega(p) & \beta_\omega(p) \end{bmatrix} \begin{bmatrix} y_F(t) \\ u(t-k_i) \end{bmatrix}, \tag{3.101}$$

where

$$y^o(t+k_p|t) \overset{\text{def}}{=} E\{y(t+k_p)|\mathcal{F}_t\} = y(t+k_p) - F(p)\omega(t+k_p). \tag{3.102}$$

The predictor given by eqn. 3.101 will be referred to as the *TIMO-type predictor.*

The following convention was chosen to denote coefficients β_j of the polynomial $\beta_\omega(p)$: the subscript j is positive for coefficients associated with past values of the input $\{u(t)\}$ in the predictor expression 3.101, $j=0$ for the coefficient corresponding to the present value of the input, and j is negative for coefficients associated with future values of the input. Note that the constant k_i determines the sample instant, $t-k_i$, up to which the values of the input $\{u(t)\}$ are required to be known.

Comment 3.3 *Knowledge of the values of the input $\{u(t)\}$ up to time instant $t-k_i$ implies, in view of eqn. 3.101, that $y^o(t+k_p|t)$ is $\mathcal{F}_t$ measurable.*

The proof of Lemma 3.5 is divided into three parts. In part 1 it is shown that the prediction calculated from eqn. 3.101 is the optimal prediction of output $y(t)$. In part 2 (p. 98) the existence of the unique polynomial $F(p)$ (eqn. 3.100) is shown. In part 3 (p. 101) the existence of the polynomials $\alpha_\omega(p)$ and $\beta_\omega(p)$ (eqns. 3.97 and 3.98) is established.

Proof of Lemma 3.5 (part 1).

Substituting from eqn. 3.89 for $y(t)$ into eqn. 3.102 one has

$$\begin{aligned} y^o(t+k_p|t) &= p^{-k_p}y(t) - p^{-k_p}F(p)\omega(t) \\ &= p^{-k_p}N(p)x_R(t) + p^{-k_p}c_\omega(p)x_{R\omega}(t) - p^{-k_p}F(p)\omega(t), \end{aligned}$$

and using eqns. 3.99 and 3.89

$$y^o(t+k_p|t) = \left[\alpha_\omega(p)N_F(p) + p^{k_i}\beta_\omega(p)d(p)\right] x_R(t) +$$

$$
\begin{aligned}
&+ \left[\alpha_\omega(p)C_\omega(p) + p^{-k_p}F(p)d(p)\right] x_{R\omega}(t) - p^{-k_p}F(p)\omega(t) \\
=\ & \alpha_\omega(p)\left[N_F(p)x_R(t) + C_\omega(p)x_{R\omega}(t)\right] + p^{k_i}\beta_\omega(p)d(p)x_R(t) \\
=\ & \alpha_\omega(p)y_F(t) + \beta_\omega(p)u(t-k_i).
\end{aligned}
$$

This establishes eqn. 3.101. The optimality of the prediction 3.101 (i.e., that $y^o(t+k_p|t) = E\{y(t+k_p)|\mathcal{F}_t\}$) can be shown in the usual way (see e.g., [9, p. 269]).

Q.E.D.

It follows from eqns. 3.102 and 3.100 that the prediction error is a moving average of order k_p

$$\epsilon(t+k_p) = y(t+k_p) - y^o(t+k_p|t) = F(p)\omega(t+k_p). \tag{3.103}$$

The variance of the prediction error is found using eqns. 3.100, 3.95, and 3.96 as in [9, p. 268]

$$E\{\epsilon(t+k_p)^2\} = FF'\sigma^2. \tag{3.104}$$

Calculation of the polynomials $\alpha_\omega(p)$ and $\beta_\omega(p)$ of the TIMO-type predictor.

The composite Diophantine equation 3.99 can be solved for $\alpha_\omega(p)$, $\beta_\omega(p)$, and $F(p)$ using elementary operations on polynomial matrices [65, chapter 3]. However, in order to avoid the complexity of dealing with polynomial matrices, the numerical representation of the Diophantine equation 3.99 is considered below. This translates the problem of solving a polynomial equation into a problem of solving a set of linear algebraic equations.

The polynomial equation 3.99 can be represented by the set of linear equations depending on the relation between k_p and k as follows (see eqns. B.4 to B.6, Appendix B)

Case $k_p \geq k$:

$$\left[\begin{array}{cc} NS^kR_0^{n+1,n+n_p+k_p-k+1} & c_\omega R_0^{n+1,n+n_p+k_p+1} \end{array}\right] =$$

$$\begin{bmatrix} \alpha_\omega & \beta_\omega & F \end{bmatrix} \begin{bmatrix} \langle N_F R_{k_p-k}^{n+1,n+k_p-k+1} \rangle_{n_p} & \langle C_\omega R_{k_p}^{n+1,n+k_p+1} \rangle_{n_p} \\ \langle d \rangle_{n_p+k_p-k} & 0_{n_p+k_p-k+1,n+n_p+k_p+1} \\ 0_{k_p,n+n_p+k_p-k+1} & \langle dR_0^{n+1,n+n_p+2} \rangle_{k_p-1} \end{bmatrix} \tag{3.105}$$

Case $k_p < k$:

$$\begin{bmatrix} NS^{k_p} R_0^{n+1,n+n_p+1} & c_\omega R_0^{n+1,n+n_p+k_p+1} \end{bmatrix} =$$

$$\begin{bmatrix} \alpha_\omega & \beta_\omega & F \end{bmatrix} \begin{bmatrix} \langle N_F \rangle_{n_p} & \langle C_\omega R_{k_p}^{n+1,n+k_p+1} \rangle_{n_p} \\ \langle dR_1^{n+1,n+2} \rangle_{n_p-1} & 0_{n_p,n+n_p+k_p+1} \\ 0_{k_p,n+n_p+1} & \langle dR_0^{n+1,n+n_p+2} \rangle_{k_p-1} \end{bmatrix}, \tag{3.106}$$

where S^i is the i-th power of the $(n+1) \times (n+1)$ shift matrix defined in eqn. 3.70 (p. 81), and $0_{r,c}$ is a $r \times c$ zero matrix. Eqns. 3.105 and 3.106 are of the form $b = x\mathcal{A}$. The structure of matrix $\mathcal{A}$ is shown in fig. 3.3 for the case $k_p > k$.

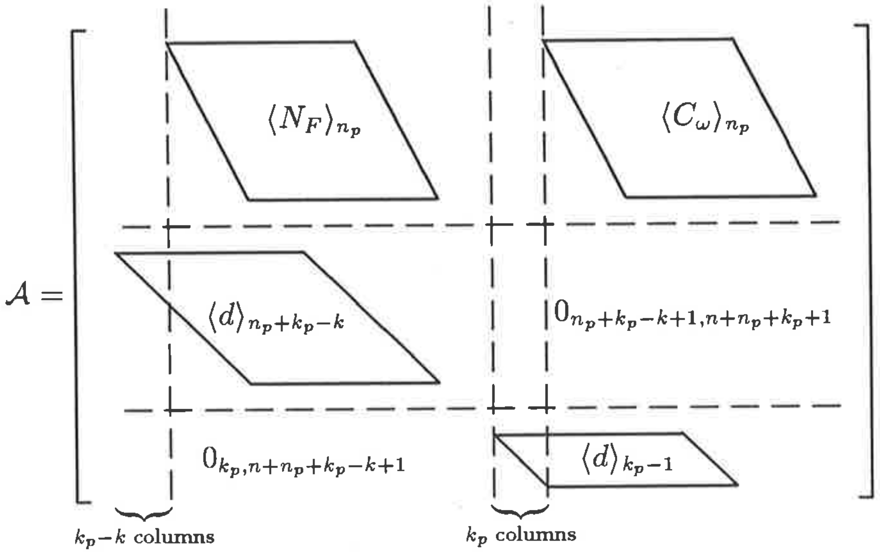
$\mathcal{A} =$
$\langle N_F \rangle_{n_p}$
$\langle C_\omega \rangle_{n_p}$
$\langle d \rangle_{n_p+k_p-k}$
$0_{n_p+k_p-k+1,n+n_p+k_p+1}$
$0_{k_p,n+n_p+k_p-k+1}$
$\langle d \rangle_{k_p-1}$
$k_p - k$ columns
k_p columns

Figure 3.3: Structure of matrix $\mathcal{A}$ for $k_p > k$.

Note from eqns. 3.105 and 3.106 (or from fig. 3.3 for $k_p > k$) that each of them can be split into two equations such that vector F of coefficients of the polynomial $F(p)$ can be calculated independently of α_ω and β_ω. For this purpose let us define the following vector

$$c_\omega^L = \begin{bmatrix} 1 & c_1 & \ldots & c_{k_p-1} \end{bmatrix}, \tag{3.107}$$

(put $c_i = 0$ for $i > n$), and matrix

$$\langle d \rangle_{i-1}^{L} = \begin{bmatrix} 1 & d_1 & \dots & d_{i-1} \\ 0 & 1 & \dots & d_{i-2} \\ 0 & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 \end{bmatrix}. \tag{3.108}$$

The first equation resulting from separation of each of eqns. 3.105 and 3.106 is

$$c_{\omega}^{L} = F \langle d \rangle_{k_p - 1}^{L}. \tag{3.109}$$

Note that vector of coefficients F can be calculated from eqn. 3.109 as

$$F = c_{\omega}^{L} (\langle d \rangle_{k_p - 1}^{L})^{-1}. \tag{3.110}$$

Proof of Lemma 3.5 (part 2).

The existence and uniqueness of the polynomial $F(p)$ such that the polynomial equation 3.99 is satisfied, is guaranteed in view of eqn. 3.110, where matrix $\langle d \rangle_{k_p - 1}^{L}$ is nonsingular (see eqn. 3.108 for $i = k_p$). Furthermore, $f_0 = 1$ (see eqn. 3.100).

Q.E.D.

Let us now define the following vector for $k_p \leq n$

$$c_{\omega}^{R} = \begin{bmatrix} c_{k_p} & \dots & c_n \end{bmatrix}, \tag{3.111}$$

and matrix

$$\langle d \rangle_{i-1}^{R} = \begin{bmatrix} d_i & d_{i+1} & \dots & d_n & 0 & \dots & 0 \\ \vdots & \vdots & & & \ddots & \ddots & \vdots \\ d_2 & d_3 & & \dots & d_{n-1} & d_n & 0 \\ d_1 & d_2 & & \dots & & d_{n-1} & d_n \end{bmatrix}. \tag{3.112}$$

The second equation resulting from the separation of each of eqns. 3.105 and 3.106 is given for two cases $k_p \geq k$ and $k_p < k$.

Case $k_p \geq k$: The second equation resulting from separation of eqn. 3.105 is as follows

$$\begin{bmatrix} N S^k R_0^{n+1, n+n_p+k_p-k+1} & b^R \end{bmatrix}$$

$$= \begin{bmatrix} \alpha_\omega & \beta_\omega \end{bmatrix} \begin{bmatrix} \langle N_F R_{k_p-k}^{n+1,n+k_p-k+1} \rangle_{n_p} & \langle C_\omega \rangle_{n_p} \\ \langle d \rangle_{n_p+k_p-k} & 0_{n_p+k_p-k+1,n+n_p+1} \end{bmatrix}, \tag{3.113}$$

where

$$b^R = \begin{cases} c_\omega^R R_0^{n-k_p+1,n+n_p+1} - F\langle d \rangle_{k_p-1}^R R_0^{n,n+n_p+1} & \text{for } k_p \le n, \\ -F\langle d \rangle_{k_p-1}^R R_0^{n,n+n_p+1} & \text{for } k_p > n. \end{cases} \tag{3.114}$$

Comment 3.4 *It follows from eqn. 3.113 that $\beta_0 = n_k \neq 0$ for $k_p = k$ (n_k is defined in eqn. 3.91).*

The coefficients of predictor polynomials $\alpha_\omega(p)$ and $\beta_\omega(p)$ can be found from eqn. 3.113. However, for the case $k_p > k$ the above equation can be further separated into two equations. For this purpose let us define the following vectors of coefficients

$$\beta_\omega^L = \begin{bmatrix} \beta_{-k_p+k} & \beta_{-k_p+k+1} & \cdots & \beta_{-1} \end{bmatrix}, \tag{3.115}$$

$$\beta_\omega^R = \begin{bmatrix} \beta_0 & \beta_1 & \ldots & \beta_{n_p} \end{bmatrix}. \tag{3.116}$$

The coefficients of the polynomial $\beta_\omega(p)$ which are contained in the vector β_ω^L are associated with future values of the input $\{u(t)\}$; the coefficients contained in the vector β_ω^R are associated with present and past values of the input $\{u(t)\}$. Note from eqn. 3.113 (or from fig. 3.3) that for $k_p > k$ it can be split into the following two equations

$$\begin{bmatrix} n_k & \ldots & n_{k_p-1} \end{bmatrix} = \beta_\omega^L \langle d \rangle_{k_p-k-1}^L, \tag{3.117}$$

(put $n_i = 0$ for $i > n$), and

$$\begin{bmatrix} NS^{k_p} R_0^{n+1,n+n_p+1} - \beta_\omega^L \langle d \rangle_{k_p-k-1}^R R_0^{n,n+n_p+1} & b^R \end{bmatrix}$$

$$= \begin{bmatrix} \alpha_\omega & \beta_\omega^R \end{bmatrix} \begin{bmatrix} \langle N_F \rangle_{n_p} & \langle C_\omega \rangle_{n_p} \\ \langle d \rangle_{n_p} & 0_{n_p+1,n+n_p+1} \end{bmatrix}, \tag{3.118}$$

where b^R is defined by eqn. 3.114. The vector β_ω^L can be calculated from eqn. 3.117 as

$$\beta_\omega^L = \begin{bmatrix} n_k & \ldots & n_{k_p-1} \end{bmatrix} \left(\langle d \rangle_{k_p-k-1}^L \right)^{-1}, \tag{3.119}$$

where matrix $\langle d \rangle_{k_p-k-1}^L$ is nonsingular (see eqn. 3.108 for $i = k_p - k$).

Case $k_p < k$: The second equation resulting from separation of eqn. 3.106 is as follows

$$\begin{bmatrix} NS^{k_p}R_0^{n+1,n+n_p+1} & (c_\omega^R R_0^{n-k_p+1,n+n_p+1} - F\langle d\rangle_{k_p-1}^R R_0^{n,n+n_p+1}) \end{bmatrix}$$
$$= \begin{bmatrix} \alpha_\omega & \beta_\omega \end{bmatrix} \begin{bmatrix} \langle N_F\rangle_{n_p} & \langle C_\omega\rangle_{n_p} \\ \langle dR_1^{n+1,n+2}\rangle_{n_p-1} & 0_{n_p,n+n_p+1} \end{bmatrix}. \tag{3.120}$$

Recall that our task was to calculate the TIMO-type predictor polynomials satisfying the composite-matrix polynomial equation 3.99. This was translated into a problem of solving a set of linear algebraic equations 3.105 or 3.106. These equations were partitioned leading to reduced dimension equations.

Summarizing, in order to find the coefficients of polynomials $\alpha_\omega(p)$ and $\beta_\omega(p)$ of the k_p-step-ahead TIMO-type predictor, one needs to calculate the vector of coefficients F of the polynomial $F(p)$ from eqn. 3.110 and

if $\mathbf{k_p = k}$ then solve eqn. 3.113 for the vectors of coefficients α_ω and β_ω using F;

if $\mathbf{k_p > k}$ then calculate the vector of coefficients β_ω^L from eqn. 3.119 and solve eqn. 3.118 for the vectors of coefficients α_ω and β_ω^R using F and β_ω^L;

if $\mathbf{k_p < k}$ then solve eqn. 3.120 for the vectors of coefficients α_ω and β_ω using F.

Eqns. 3.113, 3.118, and 3.120 have the form $b = x\mathcal{A}$. The general solution is $x = b\mathcal{A}^\dagger + P(I - \mathcal{A}\mathcal{A}^\dagger)$, where $\mathcal{A}^\dagger$ is the generalized inverse of matrix $\mathcal{A}$, and P is a parameter of the solution [122, chapter 2] (see example 3.3.1, p. 105). Existence of the solution x is investigated in the following subsection.

3.3.2 Relationship between solutions to the optimal prediction problem based on the right difference operator and state-space system representations.

The existence of the TIMO-type predictor polynomials $\alpha_\omega(p)$ and $\beta_\omega(p)$ is examined below. For this purpose the optimal predictor 3.101 will be interpreted in terms of the system state-space model. It is also shown that coefficients of the polynomial vector $\alpha_\omega(p)$ can be found independently of those of the polynomial $\beta_\omega(p)$ in terms of the system state-space model.

Proof of Lemma 3.5 (part 3).

Let us establish the following relation between coefficients of the polynomial $F(p)$ (eqn. 3.100) and the system state-space model 2.8 (p. 22)

$$f_i = CA_s^{i-1}K, \quad \text{for} \quad i = 1, \ldots, k_p - 1. \tag{3.121}$$

In order to prove the above relation note at first from eqn. 3.89 that the transfer function relating the output $y(t)$ to the noise input $\omega(t)$ is

$$H_{y,\omega}(z) = c_\omega(z^{-1})/d(z^{-1}) = 1 + \sum_{i=1}^{\infty} h_i z^{-i},$$

where the Markov parameters $h_i = CA_s^{i-1}K$ for $i = 1, 2, \ldots$ are defined in terms of the system state-space model 2.8 [80]. Using the relation between the Markov parameters and coefficients of the polynomials of the corresponding transfer function [112, eqns. 6-27 to 6-30, p. 245] we can conclude that

$$\begin{aligned} c_\omega(p) &= d(p) + CC_R(p) = 1 + [d_1 + CK]p + [d_2 + C(A_s + d_1 I_n)K]p^2 + \\ &+ \cdots + [d_n + C(A_s^{n-1} + d_1 A_s^{n-2} + \cdots + d_{n-1} I_n)K]p^n. \end{aligned}$$

Hence, the vector of coefficients c_ω^L (see eqn. 3.107) can be expressed as

$$c_\omega^L = \begin{bmatrix} 1 & d_1 + CK & \ldots & d_{k_p-1} + C(A_s^{k_p-2} + d_1 A_s^{k_p-3} + \cdots + d_{k_p-2} I_n)K \end{bmatrix}. \tag{3.122}$$

The inverse of the upper triangular Toeplitz matrix $\langle d \rangle_{k_p-1}^L$ is given by the eqns. 3.21 and 3.22, p. 64. Thus it follows from eqns. 3.110 and 3.122 that coefficients of $F(p)$ are

$$\begin{aligned} f_0 &= 1, \\ f_1 &= \begin{bmatrix} 1 & d_1 + CK \end{bmatrix} \begin{bmatrix} -d_1 \\ 1 \end{bmatrix} = CK, \end{aligned}$$

$$f_2 = \begin{bmatrix} 1 & d_1 + CK & d_2 + C(A_s + d_1 I_n)K \end{bmatrix} \begin{bmatrix} d_1^2 - d_2 \\ -d_1 \\ 1 \end{bmatrix} = CA_s K,$$

and so on. This establishes the relation 3.121.

Let us now express the output $y(t + k_p + n_p)$ in terms of the state $x(t)$, and system inputs $u(t)$ and $\omega(t)$ of the state-space model 2.8 as

$$\begin{aligned} & y(t+k_p+n_p) = Cx(t+k_p+n_p) + \omega(t+k_p+n_p) = \\ = \; & CA_s^{k_p+n_p} x(t) + \sum_{i=0}^{k_p+n_p-1} CA_s^{k_p+n_p-1-i} B_s u(t+i) + \sum_{i=0}^{n_p} CA_s^{k_p+n_p-1-i} K\omega(t+i) + \\ & + \sum_{i=1}^{k_p-1} CA_s^{k_p-1-i} K\omega(t+n_p+i) + \omega(t+k_p+n_p). \end{aligned} \tag{3.123}$$

Eqn. 3.123 can be written using eqn. 3.121 as follows

$$\begin{aligned} y(t+k_p+n_p) = CA_s^{k_p+n_p} x(t) + \sum_{i=0}^{k_p+n_p-1} CA_s^{k_p+n_p-1-i} B_s u(t+i) + \\ + \sum_{i=0}^{n_p} CA_s^{k_p+n_p-1-i} K\omega(t+i) + \sum_{i=1}^{k_p} f_{k_p-i}\omega(t+n_p+i). \end{aligned} \tag{3.124}$$

On the other hand, the output $y(t + k_p + n_p)$ can be expressed using the predictor equation 3.101 and eqn. 3.102 as

$$y(t+k_p+n_p) = \alpha_\omega(p) y_F(t+n_p) + \beta_\omega(p) u(t-k_i+n_p) + F(p)\omega(t+k_p+n_p).$$

Using the state-space model one has

$$\begin{aligned} & y(t+k_p+n_p) = \alpha_0 \left[C_F A_s^{n_p} x(t) + \right. \\ & \left. + \sum_{i=0}^{n_p-1} C_F A_s^{n_p-1-i} B_s u(t+i) + \sum_{i=0}^{n_p-1} C_F A_s^{n_p-1-i} K\omega(t+i) + C_S\omega(t+n_p) \right] \\ & + \cdots + \alpha_{n_p-1} \left[C_F A_s x(t) + C_F B_s u(t) + C_F K\omega(t) + C_S \omega(t+1) \right] + \\ & + \alpha_{n_p} \left[C_F x(t) + C_S \omega(t) \right] + \sum_{i=0}^{n_p-k_i} \beta_{n_p-i} u(t+i) + \sum_{i=1}^{k_p} f_{k_p-i}\omega(t+n_p+i) \\ & = \sum_{i=0}^{n_p} \alpha_i \left[C_F A_s^{n_p-i} x(t) \right] + \sum_{j=0}^{n_p-1} \alpha_j \left[\sum_{i=0}^{n_p-1-j} C_F A_s^{n_p-j-1-i} B_s u(t+i) \right] + \\ & + \sum_{i=0}^{n_p-k_i} \beta_{n_p-i} u(t+i) + \sum_{j=0}^{n_p-1} \alpha_j \left[\sum_{i=0}^{n_p-j-1} C_F A_s^{n_p-j-1-i} K\omega(t+i) \right] + \\ & + \sum_{i=0}^{n_p} \alpha_{n_p-i} C_S \omega(t+i) + \sum_{i=1}^{k_p} f_{k_p-i}\omega(t+n_p+i). \end{aligned} \tag{3.125}$$

The left hand side of eqns. 3.124 and 3.125 is the output $y(t+k_p+n_p)$. We can thus equate the right hand side of these equations. The comparison of the terms associated with $x(t)$ on both sides of such equation leads to

$$CA_s^{k_p+n_p} = \sum_{i=0}^{n_p} \alpha_i C_F A_s^{n_p-i}. \tag{3.126}$$

Next the noise terms are compared in eqns. 3.124 and 3.125

$$\begin{aligned}&\sum_{i=0}^{n_p} CA_s^{k_p+n_p-1-i} K\omega(t+i) + \sum_{i=1}^{k_p} f_{k_p-i}\omega(t+n_p+i) = \\ &\sum_{j=0}^{n_p-1} \alpha_j \left[\sum_{i=0}^{n_p-j-1} C_F A_s^{n_p-j-1-i} K\omega(t+i) \right] + \sum_{i=0}^{n_p} \alpha_{n_p-i} C_S \omega(t+i) + \\ &+ \sum_{i=1}^{k_p} f_{k_p-i}\omega(t+n_p+i).\end{aligned}$$

This results in the following relation for $i = 0, \ldots, n_p$

$$CA_s^{k_p+i-1}K = \sum_{j=0}^{i-1} \alpha_j C_F A_s^{i-1-j} K + \alpha_i C_S. \tag{3.127}$$

Combining eqns. 3.126 and 3.127 into a single equation, one has the following expression which involves the vector of coefficients α_ω of the polynomial vector $\alpha_\omega(p)$ defined by eqn. 3.97

$$\alpha_\omega \vartheta_\omega = v_{k_p}, \tag{3.128}$$

where matrix ϑ_ω is defined by eqn. 3.87 and

$$v_{k_p} = \left[\begin{array}{cccc} CA_s^{k_p-1}K & \ldots & CA_s^{k_p+n_p-1}K & CA_s^{k_p+n_p} \end{array} \right]. \tag{3.129}$$

The existence of solution α_ω to eqn. 3.128 for each value of the prediction horizon k_p is guaranteed by Assumption 3.9.

Now we shall find an algorithm for the calculation of coefficients of the polynomial $\beta_\omega(p)$ defined by eqn. 3.98. For this purpose, the terms associated with the input $u(t)$ are compared on both sides of the equation resulting from equating the two expressions 3.124 and 3.125 for the output $y(t+k_p+n_p)$. Thus

$$\begin{aligned}&\sum_{i=0}^{k_p+n_p-1} CA_s^{k_p+n_p-1-i} B_s u(t+i) = \\ &= \sum_{j=0}^{n_p-1} \alpha_j \left[\sum_{i=0}^{n_p-1-j} C_F A_s^{n_p-j-1-i} B_s u(t+i) \right] + \sum_{i=0}^{n_p-k_i} \beta_{n_p-i} u(t+i).\end{aligned} \tag{3.130}$$

Two cases are considered depending on the relation of the prediction horizon to the delay. Furthermore, we make use of the property of Markov parameters for a system with delay of k sample intervals, i.e., $CA_s^{l-1}B_s = 0$ for $l = 1, \ldots, k-1$ [80].

Case $\mathbf{k_p \geq k}$: Using the property of Markov parameters mentioned above, and substituting $k_i = k - k_p$ one has from eqn. 3.130

$$\sum_{i=0}^{n_p+k_p-k} CA_s^{k_p+n_p-1-i}B_s u(t+i) =$$
$$= \sum_{j=0}^{n_p-1} \alpha_j \left[\sum_{i=0}^{n_p-1-j} C_F A_s^{n_p-j-1-i} B_s u(t+i) \right] + \sum_{i=0}^{n_p+k_p-k} \beta_{n_p-i} u(t+i).$$

The following algorithm for calculation of coefficients of the polynomial $\beta_\omega(p)$ results from the above equation

$$\beta_i = CA_s^{k_p-1+i}B_s, \quad \text{for} \quad i = k-k_p, k-k_p+1, \ldots, -1, 0; \tag{3.131}$$

$$\beta_j = CA_s^{k_p-1+j}B_s - \begin{bmatrix} \alpha_0 & \ldots & \alpha_{j-1} \end{bmatrix} \begin{bmatrix} C_F A_s^{j-1} B_s \\ \vdots \\ C_F B_s \end{bmatrix}, \quad \text{for} \quad j = 1, \ldots, n_p. \tag{3.132}$$

Case $\mathbf{k_p < k}$: Using the property of Markov parameters mentioned above, and substituting $k_i = 1$ one has from eqn. 3.130

$$\sum_{i=0}^{n_p-1} CA_s^{k_p+n_p-1-i}B_s u(t+i) =$$
$$= \sum_{j=0}^{n_p-1} \alpha_j \left[\sum_{i=0}^{n_p-1-j} C_F A_s^{n_p-j-1-i} B_s u(t+i) \right] + \sum_{i=0}^{n_p-1} \beta_{n_p-i} u(t+i).$$

It follows from the above equation, that the algorithm for calculation of coefficients of the polynomial $\beta_\omega(p)$ is given by eqn. 3.132. [17]

[17] Eqn. 3.132 explains the choice of $k_i = 1$ for $k_p < k$ (see Lemma 3.5). The coefficient $\beta_1 = CA_s^{k_p}B_s - \alpha_0 C_F B_s$ of the polynomial $\beta_\omega(p)$ may be nonzero even if $k_p < k-1$ (in which case it follows from the property of Markov parameters for a system with delay k that $CA_s^{k_p}B_s = 0$) since $\alpha_0 C_F B_s$ may be nonzero.

Therefore, the existence of the polynomials of the TIMO-type predictor depends on the existence of solution to eqn. 3.128 which is guaranteed for an arbitrary value of k_p by Assumption 3.9. This completes proof of Lemma 3.5.

Q.E.D.

Comment 3.5 *Condition 3.88 of Assumption 3.9 requires that the number of rows of the matrix ϑ_ω is not less that the number of columns, which can be satisfied only if $f = \dim y_F(t) > \dim \omega(t) = 1$ (see Assumption 3.7).*

Note also that condition 3.88 leads to the following requirement on the degree n_p of the TIMO-type predictor polynomials

$$n_p \geq \left\lceil \frac{n - f + 1}{f - 1} \right\rceil ,$$

where for a real x, $\lceil x \rceil$ denotes minimal integer such that $\lceil x \rceil \geq x$. Furthermore, implicit in condition 3.88 is the requirement that the matrix ϑ, consisting of the last n columns of matrix ϑ_ω, has full column rank. The latter requirement will be satisfied if $n_p \geq \nu_F - 1$, where ν_F is the observability index of the pair (C_F, A_s). Hence, in order to satisfy Assumption 3.9, the degree of the predictor should be chosen according to

$$n_p \geq \max\left(\nu_F - 1, \left\lceil \frac{n - f + 1}{f - 1} \right\rceil\right) . \tag{3.133}$$

Condition 3.88 requires also that the upper triangular block Toeplitz matrix, consisting of the first $n_p + 1$ columns of matrix ϑ_ω, has full column rank. This implies that rank $C_S \neq 0$.

Example 3.3.1.

This example illustrates calculation of the coefficients of polynomials $\alpha_\omega(p)$ and $\beta_\omega(p)$ of the TIMO-type predictor 3.101.

Let us consider a third order system given by the state-space model 2.8 (p. 22), where $y(t) = y_1(t)$, $y_F(t) = [y_2(t)\ y_3(t)]'$, and

$$A_s = \begin{bmatrix} 1.5 & -0.75 & 0.1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad B_s = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad K = \begin{bmatrix} 0.2 \\ -0.1 \\ 0.3 \end{bmatrix}, \tag{3.134}$$

$$C = \begin{bmatrix} 1 & -0.2 & 0.5 \end{bmatrix}, \qquad C_F = \begin{bmatrix} 0.5 & 0.75 & 0.1 \\ 0 & 1 & -0.5 \end{bmatrix}, \qquad C_S = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

The corresponding RDO model 3.89 is given by

$$\begin{aligned}
d(p) &= dP_3 = [1 \ \ -1.5 \ \ 0.75 \ \ -0.1]\, P_3, \\
N(p) &= NP_3 = [0 \ \ C]\, P_3, \\
c_\omega(p) &= c_\omega P_3 = [1 \ \ -1.13 \ \ 0.51 \ \ 0.1715]\, P_3, \\
N_F(p) &= N_F P_3 = [0 \ \ C_F]\, P_3, \\
C_\omega(p) &= C_\omega P_3 = \begin{bmatrix} 1 & -1.445 & 1.010 & -0.0250 \\ 1 & -1.750 & 1.375 & -0.3575 \end{bmatrix} P_3.
\end{aligned} \tag{3.135}$$

The output the future value of which is to be predicted is $y(t) = y_1(t)$. The outputs which are available for utilization in the predictor are $y_F(t) = [y_2(t)\ y_3(t)]'$.

Let us find the ($k_p = 3$)-step-ahead TIMO-type predictor for the above system. The observability index is $\nu_F = 2$, $f = 2$, and $n = 3$; the degree of the predictor polynomials is chosen according to condition 3.133 as $n_p = \max(1, 2) = 2$.

First, the coefficients of the polynomial vector $\alpha_\omega(p)$ are calculated from the system state-space model 3.134. From eqns. 3.129 and 3.87 one has

$$\begin{aligned}
v_3 &= [0.4665 \ \ 0.5005 \ \ 0.4324 \ \ 0.9650 \ \ -0.8581 \ \ 0.1368], \\
\vartheta_\omega &= \begin{bmatrix} 1 & 0.055 & 0.3425 & 1.975 & -1.075 & 0.15 \\ 1 & -0.250 & 0.2500 & 1.000 & -0.750 & 0.10 \\ 0 & 1.000 & 0.0550 & 1.500 & -0.275 & 0.05 \\ 0 & 1.000 & -0.2500 & 1.000 & -0.500 & 0 \\ 0 & 0 & 1.000 & 0.500 & 0.750 & 0.10 \\ 0 & 0 & 1.000 & 0 & 1.000 & -0.50 \end{bmatrix},
\end{aligned}$$

and rank $\vartheta_\omega = 6$, i.e., Assumption 3.9 is satisfied. The solution [18] to eqn. 3.128 yields [19]

$$\begin{aligned}\alpha_\omega &= [\alpha_0 \;\; \alpha_1 \;\; \alpha_2] = [\alpha_{20} \;\; \alpha_{30} \;\; \alpha_{21} \;\; \alpha_{31} \;\; \alpha_{22} \;\; \alpha_{32}] \\ &= [1.8081 \;\; -1.3416 \;\; -4.2219 \;\; 4.2876 \;\; 1.5618 \;\; -0.1092]\,. \qquad (3.136)\end{aligned}$$

Since $k_p > k = 1$, the coefficients of the $\beta_\omega(p)$ polynomial are calculated from eqns. 3.131 and 3.132

$$\beta_\omega = [\beta_{-2} \;\; \beta_{-1} \;\; \beta_0 \;\; \beta_1 \;\; \beta_2] = [1 \;\; 1.3 \;\; 1.7 \;\; 0.7709 \;\; 2.1079]\,.$$

Alternatively, the coefficients of both polynomials $\alpha_\omega(p)$ and $\beta_\omega(p)$ can be found from the system RDO representation 3.135. Note from eqn. 3.107 that vector $c^L_\omega = [1 \;\; -1.13 \;\; 0.51]$, and vector of coefficients of the polynomial $F(p)$ follows from eqn. 3.110 as $F = [1 \;\; 0.37 \;\; 0.315]$. The vector β^L_ω (eqn. 3.115) is found from eqn. 3.119 as $\beta^L_\omega = [1 \;\; 1.3]$. Now it remains to solve eqn. 3.118; $k_p = n$, $c^R_\omega = 0.1715$ (eqn. 3.111), and the left hand side of eqn. 3.118 is

$$\begin{aligned}&\left[\left(NS^3R_0^{4,6} - \beta^L_\omega \langle d\rangle_1^R R_0^{3,6}\right) \quad \left(c^R_\omega R_0^{1,6} - F\langle d\rangle_2^R R_0^{3,6}\right)\right] \\ = \;& [1.7 \;\; -0.875 \;\; 0.13 \;\; 0 \;\; 0 \;\; 0 \;\; 0.4665 \;\; -0.1992 \;\; 0.0315 \;\; 0 \;\; 0 \;\; 0]\,.\end{aligned}$$

The matrix on the right hand side of eqn. 3.118 is of dimension 11 × 12 and is of full row rank. Therefore, the solution is unique and yields α_ω given by eqn. 3.136 and $\beta^R_\omega = [1.7 \;\; 0.7709 \;\; 2.1079]$ (see eqn. 3.116).

3.3.3 Discussion of properties of the TIMO-type predictor.

It is desirable to compare the properties of the TIMO-type predictor developed for a system having the feedback configuration FD with those of the TISO-type predictor for the same system within the feedback configuration FI.

[18] All computations were performed using MATLAB package [110].

[19] Recall that $\alpha_\omega(p)$ is a $1 \times f$ polynomial vector having coefficient vectors α_j for $j = 0, \ldots, n_p$ of dimension $1 \times f$ (see eqn. 3.97); each coefficient vector α_j has f elements α_{ij} (α_{ij} is associated with the system output $y_i(t)$).

The RDO model of the system having the feedback configuration FI is given by eqn. 3.89 with $y_F(t) = y(t)$. The corresponding state-space model is given by eqn. 2.8 (p. 22) where $C_F = C$, $C_S = 1$, and $f = 1$; hence, Assumption 3.9 is not satisfied and Lemma 3.5 cannot be used. However, it can be shown following the derivation of the optimal predictor for a TISO system given by the ARMAX model [83, chapter 6] [9, Lemma 7.4.1, p. 268], that if all roots of the noise polynomial $c_\omega(z^{-1})$ lie *strictly inside the unit circle*, then the optimal k_p-step-ahead prediction $y^o(t+k_p|t)$ of output $y(t)$ satisfies

$$c_\omega(p)y^o(t+k_p|t) = \left[\begin{array}{cc} \alpha_{s\omega}(p) & \beta_{s\omega}(p) \end{array}\right] \left[\begin{array}{r} y(t) \\ u(t+k_p-k) \end{array}\right]. \tag{3.137}$$

The polynomials $\alpha_{s\omega}(p) = \alpha_{s\omega}P_{n-1}$ and $F_s(p) = F_s P_{k_p-1}$ satisfy the following polynomial equation

$$c_\omega(p) = \left[\begin{array}{cc} F_s(p) & \alpha_{s\omega}(p) \end{array}\right] \left[\begin{array}{c} d(p) \\ p^{k_p} \end{array}\right], \tag{3.138}$$

and

$$\beta_{s\omega}(p) = p^{-k}F_s(p)N(p). \tag{3.139}$$

The predictor 3.137 will be referred to as the *TISO-type predictor.*

The first aspect to be compared is the "quality" of the TISO- and TIMO-type predictions.

The relation between the prediction errors of the TISO- and TIMO-type predictions.

The polynomial equation 3.138 implies that $F_s = c_\omega^L \left(\langle d\rangle_{k_p-1}^L\right)^{-1}$, where c_ω^L and $\langle d\rangle_{k_p-1}^L$ are defined by eqns. 3.107 and 3.108, respectively. Note from eqn. 3.110 that $F = F_s$. Hence, from eqns. 3.103 and 3.104 it follows that

Comment 3.6 *The prediction errors and their variances are identical for the k_p-step-ahead TISO- and TIMO-type optimal predictions.*

This result can be explained as follows. Firstly, both predictors minimize the mean-square prediction error. Secondly, the output data sequences, i.e., $\{y_F(t)\}$, although different for both predictors, are not statistically independent. For the TISO-type predictor the output data sequence is the sequence the future values of which are to be predicted since $y_F(t) \equiv y(t)$. For the TIMO-type predictor the output data sequence $\{y_F(t)\}$ is statistically related via model 3.89 to the sequence $\{y(t)\}$ the future values of which are to be predicted.

Similarity between the predictors for stochastic and deterministic multi-output systems.

Note that calculation of the TISO-type prediction $y^o(t + k_p|t)$ from eqn. 3.137 involves output prediction filtered by the polynomial $1 - c_\omega(p)$. On the other hand, the TIMO-type prediction 3.101 does *not* involve past output predictions. In fact, the same input-output sequences are involved in the TIMO-type prediction for *stochastic* systems and the SIMO-type prediction 3.68 (p. 80) for *deterministic* systems. This similarity between the predictors for stochastic and deterministic multi-output systems implies that a number of features which are associated with predictors for deterministic systems extends to predictors for stochastic systems. These features are significant in both nonadaptive (see discussion below) and adaptive (see subsection 3.3.4) applications.

Comparison of assumptions concerning the noise polynomial $c_\omega(z^{-1})$ required for the TISO- and TIMO-type predictors.

Considering the *Spectral Factorization Theorem* (SFT) [82, Theorem 10.1, p. 47] [136], it can be assumed without loss of generality that the roots of the noise polynomial $c_\omega(z^{-1})$ lie *inside or on* the unit circle. However, it is commonly assumed in the derivation of optimal time-invariant predictors (such as the TISO-type predictor 3.137) [83,124,125,137] [59, chapter 12] and optimal predictive controllers [83,25,31,125,99,137,138] [9, chapter 11] [59, chapter 12] that the roots of $c_\omega(z^{-1})$ lie *strictly* inside the unit circle, thus excluding the possibility of roots lying on the unit circle. Note that such an assumption is not required for the TIMO-type predictor because the noise polynomial $c_\omega(p)$ is not involved in eqn. 3.101, i.e.,

Comment 3.7 *The TIMO-type predictor permits arbitrary location of the roots of the noise polynomial $c_\omega(z^{-1})$ (i.e.,* inside or on *the unit circle in the light of the SFT), in contrast to the TISO-type predictor (which constrains the roots to lie* inside *the unit circle).*

If there are roots of the noise polynomial $c_\omega(z^{-1})$ lying on the unit circle, then it is possible to use *time-varying* or *restricted complexity* TISO-type predictors [9, chapter 7]. The time-varying predictor will converge asymptotically to the time-invariant predictor (under some assumptions); on the other hand, the restricted complexity (time-invariant) predictor will yield a *suboptimal* prediction [20] [9, p. 272].

It is known that there are roots of the noise polynomial in the ARMAX model lying on the unit circle if the system has uncontrollable modes lying on the unit circle [9, chapter 7]. It should be noted, however, that roots of $c_\omega(z^{-1})$ lying on the unit circle cannot arise due to the uncontrollable systems modes since the system is assumed to be controllable (see Assumption 3.8, p. 92).

The effect of arbitrary initial conditions on the TISO- and TIMO-type predictions.

If it is required that $y^o(t+k_p|t)$ is an optimal prediction for all sample instants, then appropriate initial conditions must be chosen for the predictors 3.101 and 3.137.

It is well known that if arbitrary initial conditions are chosen for the TISO-type predictor, then the effect of incorrect initial conditions diminishes exponentially provided $c_\omega(z^{-1})$ has all roots inside the unit circle [9, Remark 7.4.3, p. 269]. This is because $z^n c_\omega(z^{-1})$ is the characteristic polynomial of a filter, having inputs $y(t)$ and $u(t)$ and output $y^o(t+k_p|t)$, which represents the TISO-type predictor [83, Remark 3, p. 170]. Hence, the rate of decay of the effect of incorrect initial conditions depends on the location of the roots of $c_\omega(z^{-1})$.

[20] This property is illustrated in example 6.2.3, (p. 255) for the application of the restricted complexity TISO-type predictor to self-tuning MPE control of systems having the noise polynomial with roots on the unit circle.

On the other hand, note from eqn. 3.101 that the characteristic polynomial of a filter representing the TIMO-type predictor is z^{n_p}. Hence,

Comment 3.8 *If arbitrary initial conditions are assumed, then the effect of incorrect initial conditions on the TIMO-type prediction $y^o(t+k_p|t)$ dies away after n_p sample instants regardless of the location of the roots of the noise polynomial $c_\omega(z^{-1})$.*

Output measurement configuration.

The next property of the TIMO-type predictor is associated with the flexibility of the output measurement configuration. The optimal TIMO-type predictor does not require measurement of the output $y(t)$, the future value of which is to be predicted [21], in contrast to the TISO-type predictor. Note, however, that measurement of more than one output ($f > 1$) is required by the TIMO-type predictor (see Assumption 3.7).

The optimal TIMO-type predictor for a system with the stochastic input only.

The system generating sequence $\{y(t)\}$, the future values of which are to be predicted, was assumed to be driven by two input sequences: the known (i.e., measurable) sequence $\{u(t)\}$ and (unmeasurable) white noise sequence $\{\omega(t)\}$ (see Assumption 3.7). However, it is often assumed that the system generating the output sequence $\{y(t)\}$ is driven only by the white noise sequence $\{\omega(t)\}$ [139, chapter 6] [116, chapter 3] [123,137]. Such a system can be alternatively thought of as a TIMO system with zero input $u(t)$. The corresponding predictor results from the TIMO-type predictor by setting $u(t) = 0$, i.e.,

$$y^o(t+k_p|t) = \alpha_\omega(p)y_F(t).$$

[21] This property is illustrated for the application of the TIMO-type predictor to MPE control of TIMO systems in example 6.2.1, p. 224.

The composite Diophantine equation 3.99 reduces to the following polynomial equation

$$p^{-k_p} c_\omega(p) = \begin{bmatrix} \alpha_\omega(p) & F(p) \end{bmatrix} \begin{bmatrix} C_\omega(p) \\ p^{-k_p} d(p) \end{bmatrix},$$

from which the polynomial $\alpha_\omega(p)$ can be determined.

3.3.4 Adaptive TIMO-type predictor.

The problem of adaptive prediction for stochastic systems having the feedback configuration FI has been considered in [123,124,125,126,137,9]. In this subsection the *direct* adaptive TIMO-type predictor is introduced. For this purpose let us rewrite the predictor eqn. 3.101 as

$$y^o(t|t-k_p) = \Theta' \phi(t-k_p), \tag{3.140}$$

where the $((f+1)(n_p+1)-k_i) \times 1$ parameter vector Θ and the regression vector $\phi(t)$ are defined as

$$\begin{aligned} \Theta &= [\alpha_\omega \ \ \beta_\omega]', & (3.141) \\ \phi(t) &= \begin{bmatrix} y_F(t)' & \dots & y_F(t-n_p)' & u(t-k_i) & \dots & u(t-n_p) \end{bmatrix}'. & (3.142) \end{aligned}$$

The estimation algorithm employed to estimate parameters of the TIMO-type predictor is the recursive least squares with condition number monitoring (RLS-CNM). The condition number monitoring (CNM) technique was introduced in [57] for the recursive extended least squares (RELS) estimator and will be discussed in subsection 6.2.2.

In order to calculate the equation error required by the RLS-CNM estimator we need Assumption 3.6 (p. 86), i.e., $y(t)$ is known.

The direct *adaptive TIMO-type predictor* consists of the following steps which are performed at every sample instant t:

step 1. calculate the predictor parameter estimates

$$\widehat{\Theta}(t) = \left[\widehat{\alpha}_\omega(t) \ \ \widehat{\beta}_\omega(t)\right]' \tag{3.143}$$

using the recursive least squares algorithm with condition number monitoring RLS-CNM (see eqns. E.5 to E.7, Appendix E), with the equation error defined as (see eqn. E.1, Appendix E)

$$e(t) = y(t) - \widehat{\Theta}(t-1)'\phi(t-k_p); \tag{3.144}$$

step 2. calculate the prediction of $y(t)$ as follows

$$\widehat{y}(t) = \widehat{\Theta}(t-k_p)'\phi(t-k_p). \tag{3.145}$$

For the purpose of establishing the convergence properties of the adaptive prediction 3.145 we need the standard assumptions (cf. [125,126] [9, p. 372]).

Assumption 3.11 (Noise assumption) *The noise sequence $\{\omega(t)\}$ is sample mean-square bounded almost surely, i.e.,*

$$\limsup_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} \omega(t)^2 < \infty \quad a.s. \tag{3.146}$$

Furthermore,

Assumption 3.12 (Input-output data assumption) *The input sequence $\{u(t)\}$, and output sequences $\{y_F(t)\}$ and $\{y(t)\}$, are sample mean-square bounded almost surely, i.e.,*

$$\begin{aligned} \limsup_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} u(t)^2 &< \infty \quad a.s., \\ \limsup_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} \|y_F(t)\|^2 &< \infty \quad a.s., \\ \limsup_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} y(t)^2 &< \infty \quad a.s. \end{aligned}$$

We have the following convergence result.

Lemma 3.6 *Let the output sequence $\{y(t)\}$ be generated by the system satisfying Assumptions 3.7 to 3.10 (p. 91), 3.6 (p. 86), 3.11 and 3.12. Then the adaptive TIMO-type prediction $\widehat{y}(t)$ calculated from eqn. 3.145 with the RLS-CNM estimator (implemented with no forgetting) converges to the optimal linear prediction $y^o(t|t-k_p)$ of output $y(t)$ in the following sense*

$$\lim_{N\to\infty} \frac{1}{N}\sum_{t=k_p}^{N} \left[\widehat{y}(t) - y^o(t|t-k_p)\right]^2 = 0 \quad a.s. \tag{3.147}$$

Proof. This lemma can be proved following [125,126,9] and using a similar technique to this employed in the proof of Theorem 6.2, p. 230.

Q.E.D.

Note that no persistent excitation condition is needed for the convergence result 3.147. Nothing is said, however, about the convergence of the predictor parameter estimates to the true values of parameters of the optimal predictor.

For the completeness of exposition, let us mention adaptive predictors, and associated convergence results, developed for TISO systems. In [123] the direct adaptive predictor employing the RLS estimator was proposed for a system driven only by the white noise sequence; it is shown that *if* the predictor parameter estimates converge to some values, then the adaptive prediction converges to the optimal prediction. However, the problem of parameter estimates convergence was not addressed in [123].

The convergence in the sense of eqn. 3.147, which does not depend on the estimates convergence, was firstly established for the adaptive predictor employing the stochastic gradient estimator for TISO systems in [125]. Subsequently, corresponding convergence results were established for adaptive predictors based on the variants of the recursive extended least squares (RELS) estimator. In [126] the adaptive predictor involving a bank of k_p interlaced RELS estimators was introduced. The adaptive predictor proposed in [56] employs the *weighted* RELS estimator. The CNM technique was applied to the RELS estimator employed in the adaptive predictor of [57].

Discussion of properties of the adaptive TIMO-type predictor.

The properties of the TIMO-type predictor which are relevant to the adaptive case are briefly discussed below. We shall return to these properties in detail in subsection 6.2.2 (pp. 231-235) for the application of the predictor to self-tuning control of TIMO systems.

Firstly, note that the parameters of the TIMO-type predictor are estimated by the ordinary *linear regression* estimator, such as the RLS. On the other hand, the corresponding adaptive TISO-type predictor involves *pseudo-linear regression* estimator, such as the stochastic gradient or RELS [125,126,56] [9, section 9.3] (cf. discussion on pages 231-234). (The RLS estimator can be employed in the adaptive predictor for a TISO system driven only by the white noise sequence [123].) A significant improvement in the *convergence rate* of the TIMO-type predictor parameter estimates, and enhancement of the *numerical robustness* of the estimator, can be expected in comparison with those of the TISO-type adaptive predictor; this is a consequence of using the linear regression estimator instead of the pseudo-linear regression estimator (cf. Comments 6.8, (p. 234), 6.9 (p. 234) and example 6.2.2 (p. 240)).

Secondly, the assumptions concerning the noise polynomial $c_\omega(z^{-1})$ and required by the adaptive TISO-type predictor are modified for the TIMO-type predictor. In particular, the adaptive TIMO-type prediction converges to the optimal prediction (in the sense of eqn. 3.147) regardless of the location of the roots of the noise polynomial $c_\omega(z^{-1})$ and for arbitrary initial conditions (cf. Comment 3.8, p. 111). On the other hand, the adaptive TISO-type predictor requires roots of the noise polynomial to lie *inside* the unit circle; alternatively, the restricted complexity predictor can be used to yield an asymptotically *suboptimal* prediction [9, Remark 9.3.2, p. 377]. Furthermore, the convergence of the adaptive TISO-type predictor is subject to the so-called *strictly positive real* (SPR) condition [53] imposed on a transfer function involving the noise polynomial $c_\omega(z^{-1})$ [125,126,56] [9, section 9.3]. No SPR condition is required for the convergence of the adaptive TIMO-type predictor (cf. Comment 6.10 on page 235 and example 6.2.2, p. 240).

Summarizing, the adaptive TIMO-type predictor has a number of desirable properties. However, some of the assumptions which were made might be difficult to verify (e.g., Assumption 3.9). Note also that Assumption 3.8 excludes uncontrollable systems, which is not the case of the TISO-type predictor based on the system ARMAX model [9, chapter 7].

3.4 Concluding remarks.

In this chapter, the problem of prediction of the future values of outputs of deterministic and stochastic systems having the feedback configuration FI and FD is considered.

In section 3.1, two types of predictors for a MIMOS deterministic system having the feedback configuration FI are presented. They differ in the prediction horizon, which is related to the degree of the interactor matrix associated with the MIMOS system, and in the choice of the left or right interactor matrix, embedded in the design. The idea of employing the interactor matrix in the development of predictors was proposed elsewhere. However, it is shown here that this idea can be generalized to predictors with the *multi-step-ahead* prediction horizon. (The left interactor matrix is employed in the present work for the multi-step-ahead predictor, but a parallel analysis for the right interactor matrix could be performed.) Moreover, the *nilpotent interactor matrices*, being nonunique characterizations of a multivariable system delay structure, are employed instead of the unique interactor matrices. Furthermore, the new algorithms are derived for the calculation of the *matrices of coefficients* of the polynomial matrices which define predictors. These algorithms are amenable to *computer-based calculations* using any matrix-oriented software. In particular, a *recursive* algorithm is developed for the calculation of matrices of coefficients of the predictor polynomials for subsequent values of the prediction horizon. The predictors will be employed in the development of the MPE self-tuning controller in chapter 4 and the long-range predictive self tuner in section 5.1.

In the remaining two sections of this chapter, the problem of prediction for systems having the *feedback configuration FD* is addressed.

In section 3.2, the *new SIMO-type predictor* for a SIMO deterministic system having the feedback configuration FD is developed. The polynomials which define the SIMO-type predictor are likely to be of degree smaller, and to involve less coefficients, than those of the SISO-type predictor. For systems with known parameters the method for the *off-line* calculation of the coefficients of the polynomials defining the predictor is presented. This method is based on a *numerical representation* of the Diophantine equation. For systems with unknown parameters the *adaptive* SIMO-type predictor is proposed. The *convergence* of the adaptive prediction to the actual value of the system output is established without any persistent excitation condition. An example illustrates improvement in the *convergence rate* of parameter estimates of the adaptive SIMO-type predictor in comparison with that of the SISO-type predictor in application to a robot manipulator. Such an improvement is due to the reduction in the number of estimated parameters of the SIMO-type predictor in comparison with the SISO-type predictor. Faster convergence rate of parameter estimates improves the performance of the adaptive predictor. The SIMO-type predictor will be employed in the development of the minimum prediction error self-tuning controller in section 6.1.

In section 3.3, the *new optimal TIMO-type predictor* for a TIMO stochastic system having the feedback configuration FD is developed. A number of properties of the TIMO-type predictor which are desirable in adaptive applications motivated development of the predictor for the *arbitrary* prediction horizon. For systems with known parameters the method for the *off-line* calculation of the coefficients of the polynomials defining the predictor is proposed. This method is based on a *numerical representation* of the Diophantine equation. For systems with unknown parameters the *adaptive* TIMO-type predictor is proposed. The adaptive TIMO-type predictor employs the *linear regression* estimator (the RLS) in contrast to the adaptive TISO-type predictors for stochastic systems having the feedback configuration FI which use *pseudo-linear* regression estimators. Consequently, superior convergence rate of predictor parameter estimates, and enhancement of the numerical robustness of the estimator, can be expected for the adaptive TIMO-type predictor in comparison with the TISO-type predictor. The *convergence* of the adaptive TIMO-type prediction is established *without the SPR condition* imposed on the noise polynomial in contrast to the adaptive TISO-type prediction. The TIMO-type predictor, with the prediction horizon set to the value of delay from the control input to the controlled output, will be employed in the development of the minimum prediction error self-tuning controller in section 6.2.

Chapter 4

Self-tuning minimum prediction error control based on the right nilpotent interactor matrix for a multi-input, multi-output, deterministic system.

In this chapter a new self-tuning minimum prediction error (MPE) controller is developed for a linear multivariable square (MIMOS) deterministic system having the feedback configuration FI (see page 18). The design of this self-tuner involves characterization of the multivariable system delay structure by the right nilpotent interactor (RNI) matrix, introduced in subsection 2.3.3.

In section 4.1 the MPE controller, which minimizes the k-step-ahead single-stage cost function for a system with known parameters, is derived (see approach (iii) on p. 60). The design of this controller is based on the approach proposed in [35]. However, the RNI matrix is employed in place of the unique right interactor matrix. Furthermore, the system DARMA model is used instead of the system transfer matrix. These features facilitate development of the *indirect* self-tuning MPE controller possessing properties desirable in adaptive control (see section 4.2).

Firstly, the resulting self tuner does *not* require prior knowledge of the interactor matrix (but does involve more complex algorithms). Instead of assuming prior knowledge of the right interactor matrix, the algorithm introduced in subsection 2.3.3 is employed for the *on-line* calculation of the RNI matrix from the estimates of the DARMA model (see approach 5 (b), p. 39). Knowledge of the upper bound on the degree of polynomials in the DARMA model is the only prior system knowledge required by the self tuner. This is a significant relaxation of the assumptions concerning prior system knowledge in comparison with the existing solutions to self-tuning predictive control (see subsection 2.3.1). Furthermore, the only parameters to be estimated are those of the DARMA model.

Secondly, a unified treatment of the MPE control of plants subject to *deterministic disturbances* is facilitated by representation of the system by the DARMA model.

The application of the self tuner to control a robot manipulator is considered in section 4.3.

In this chapter we shall use the results of subsections 2.3.3 and 3.1.2.

4.1 Minimum prediction error control based on the right nilpotent interactor matrix for a system with known parameters.

In this section:

- the MPE controller which minimizes a single-stage cost function is derived (see approach (iii) on page 60);
- stability conditions for the closed-loop system are presented.

The developments of this section form a generalization of results presented in [35], where the application of the (unique) right interactor matrix $\xi_R(z)$ in the design of MPE controllers was originally proposed. In the derivation of the MPE controller we employ a nonunique characterization of the plant delay structure, namely the RNI matrix $K_R(z)$, rather than the unique right interactor matrix. The MPE controller is designed for the DARMA model rather than for the transfer matrix representation employed in [35]. The DARMA model describes, in general, the controlled plant with the generator of deterministic disturbances incorporated in the system model (see discussion on p. 19 and p. 28). The resulting controller eliminates the effect of deterministic disturbances affecting the plant.

Let us consider a MIMOS deterministic system satisfying Assumptions 3.1, 3.2, and 3.3 (see pp. 58-59). Consider the following quadratic cost function proposed in [35]

$$J_k = \|y(t+k) - r(t+k)\|^2_\Upsilon + \|S(p)u_k(t)\|^2_\Lambda, \tag{4.1}$$

where k is the degree of the right interactor matrix of the system, Υ and Λ are positive definite and positive semidefinite matrices, respectively; the $m \times m$ weighting polynomial matrix $S(p)$ is defined as

$$S(p) = S_0 + S_1 p + \cdots + S_{n_s} p^{n_s}, \qquad \det S_0 \neq 0 \quad \text{and} \quad n_s \geq 0.$$

Signal $u_k(t)$, defined in Lemma 3.2 (eqn. 3.41, p. 72), will be called an auxiliary control signal.

The first term in cost function 4.1 penalizes the tracking error; the second term penalizes the auxiliary control signal (i.e., the control signal filtered by the inverse of the right interactor matrix).

We have the following result corresponding to that obtained for the system transfer matrix description with the unique interactor matrix in [35].

Lemma 4.1 *Consider a MIMOS system satisfying Assumptions 3.1, 3.2, and 3.3 (see pp. 58-59). For such a system the auxiliary control signal $u_k(t)$, which minimizes cost function 4.1, is given by*

$$V(p)u_k(t) = \begin{bmatrix} M(p) & -K(p) \end{bmatrix} \begin{bmatrix} r(t+k) \\ y(t) \end{bmatrix}, \tag{4.2}$$

where the $m \times m$ polynomial matrices $V(p)$, $M(p)$, and $K(p)$ are defined as

$$V(p) \;=\; V_0 + V_1 p + \cdots = \delta_0' \Upsilon \delta(p) + S_0' \Lambda S(p), \tag{4.3}$$

$$
\begin{aligned}
M(p) &= \delta_0'\Upsilon, && (4.4)\\
K(p) &= \delta_0'\Upsilon\gamma(p), && (4.5)
\end{aligned}
$$

and the $m \times m$ polynomial matrices $\gamma(p)$ and $\delta(p)$ are defined in eqns. 3.47 and 3.48, p. 72. The control signal is then calculated from eqn. 3.41 (p. 72), i.e.,

$$
u(t) = \tilde{K}_R(p)u_k(t), \tag{4.6}
$$

where $\tilde{K}_R(p) = p^k K_R(q)$ (see eqn. 3.42, p. 72), and $K_R(q)$ is the RNI matrix of the MIMOS system.

Proof.

Let us substitute the k-step-ahead predictor expression 3.43 (see Lemma 3.2, p. 72) for $y(t+k)$ in the cost function 4.1, which leads to

$$
\begin{aligned}
J_k &= \|y(t+k) - r(t+k)\|_\Upsilon^2 + \|S(p)u_k(t)\|_\Lambda^2\\
&= \|\gamma(p)y(t) + \delta(p)u_k(t) - r(t+k)\|_\Upsilon^2 + \|S(p)u_k(t)\|_\Lambda^2\\
&= \left[y(t)'\gamma(p)' + u_k(t)'\delta_0' + u_k(t-1)'p^{-1}[\delta(p)' - \delta_0'] - r(t+k)'\right]\Upsilon \times\\
&\quad \left[\gamma(p)y(t) + \delta_0 u_k(t) + p^{-1}[\delta(p) - \delta_0]u_k(t-1) - r(t+k)\right] +\\
&\quad + \left[u_k(t)'S_0' + u_k(t-1)'p^{-1}[S(p)' - S_0']\right]\Lambda\left[S_0u_k(t) + p^{-1}[S(p) - S_0]u_k(t-1)\right].
\end{aligned}
$$

Then eqn. 4.2 follows from differentiation of the above cost function with respect to $u_k(t)$ and setting the result to zero. Eqn. 4.6 follows from eqns. 3.41 and 3.42 (p. 72) for $\overline{K}_R(q) = K_R(q)$.

Q.E.D.

Note from eqn. 4.3 that the coefficient matrix of $V(p)$ associated with the value of the auxiliary control signal at time t, $u_k(t)$, is $V_0 = [\delta_0'\Upsilon\delta_0 + S_0'\Lambda S_0]$. This coefficient matrix is nonsingular because δ_0 is nonsingular (see Lemma 3.2, p. 72). Hence the controller 4.2 is causal and the auxiliary control signal $u_k(t)$ can be determined uniquely (for a given RNI matrix). The control signal $u(t)$ is a function of past values of the auxiliary control signal (see eqn. 4.6).

Now we shall comment on the choice of the parameters of the MPE controller.

The second term in the cost function 4.1 is introduced to penalize the auxiliary control signal to avoid excessive control effort. This leads to a well-known problem of λ-offset [84]: if the controlled plant does not have integrating properties and $S(1) \neq 0$, then the steady-state auxiliary control signal u_{kss}, which minimizes J_k, introduces a nonzero steady-state error between the constant reference sequence r_{ss} and steady-state output y_{ss}. The λ-offset problem can be avoided by the choice of the weighting polynomial matrix $S(p)$ such that $S(1) = 0$, e.g.,

$$S(p) = I_m - pI_m. \tag{4.7}$$

In this case the cost function J_k is independent of the auxiliary control signal u_{kss} in the steady-state. Then it follows from eqn. 4.2 that in the steady-state $\delta(1)u_{kss} = r_{ss} - \gamma(1)y_{ss}$, and from eqn. 4.6 $u_{ss} = K_R(1)u_{kss}$. Substituting for $\delta(1)$ from eqn. 3.45, and for $\gamma(1)$ from eqn. 3.44, and using $A(1)y_{ss} = B(1)u_{ss}$ we have $y_{ss} = r_{ss}$, i.e., there is zero steady-state error. Alternatively, deviations of $u_k(t)$ from the steady-state auxiliary control signal value can be penalized in the cost function to eliminate λ-offset, as in [35].

The choice of the weighting matrix Λ may be crucial to ensure stability of the closed-loop system. This is revealed by the following stability condition.

Lemma 4.2 *For the closed-loop system resulting from the control law given by eqns. 4.2 and 4.6, the sequences of $\{y(t)\}$, $\{u(t)\}$, and $\{u_k(t)\}$ are bounded provided that*

(a) for polynomial matrices $A(p)$ and $B(p)$ which are left coprime

$$\det \begin{bmatrix} A(z^{-1}) & -B(z^{-1}) & 0 \\ 0 & I_m & -\tilde{K}_R(z^{-1}) \\ \delta_0' \Upsilon z^k & 0 & S_0' \Lambda S(z^{-1}) \end{bmatrix} \neq 0, \qquad |z| \geq 1; \tag{4.8}$$

(b) for polynomial matrices $A(p)$ and $B(p)$ which are not left coprime

(i)

$$\det\begin{bmatrix} A_{con}(z^{-1}) & -B_{con}(z^{-1}) & 0 \\ 0 & I_m & -\tilde{K}_R(z^{-1}) \\ \delta_0'\Upsilon z^k & 0 & S_0'\Lambda S(z^{-1}) \end{bmatrix} \neq 0, \qquad |z| \geq 1; \tag{4.9}$$

where $[A_{con}(p),\ B_{con}(p)]$ *is the controllable part of* $[A(p),\ B(p)]$*, and*

(ii) all uncontrollable modes of $[A(p),\ B(p)]$ *lie inside or on the unit circle and those on the unit circle have Jordan block of size 1.*

Proof.

It follows from eqns. 3.1, 4.6, 4.2 and 3.43 (p. 72) that the closed-loop system is characterized by the following LDO model

$$\begin{bmatrix} A(p) & -B(p) & 0 \\ 0 & I_m & -\tilde{K}_R(p) \\ \delta_0'\Upsilon q^k & 0 & S_0'\Lambda S(p) \end{bmatrix}\begin{bmatrix} y(t) \\ u(t) \\ u_k(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \delta_0'\Upsilon q^k \end{bmatrix} r(t). \tag{4.10}$$

The LDO model is equivalent to an observable state-space model [75, chapter 4]. The common (polynomial matrix) factors of the pair $A(p)$ and $B(p)$ represent the uncontrollable subsystem. The above stability conditions follow then from [9, Lemma B.3.3, p. 486] which assumes boundedness of the reference sequence $\{r(t)\}$ (see also [140, Theorem 1.2, p. 49]).

Q.E.D.

The stability conditions of this type were introduced for the MPE controller based on the left interactor matrix $\xi_L(z)$ in [102,9] and are sometimes referred to as the "generalized minimum phase" conditions. Note that if $\Lambda = 0$, then condition 4.8 reduces to

$$\det\left[K_R(z)B(z^{-1})\right] \neq 0 \quad \text{for} \quad |z| \geq 1. \tag{4.11}$$

Since $\det K_R(z) = c_r z^{k_r}$ (see eqn. 2.51, p. 45), condition 4.11 becomes $\det B(z^{-1}) \neq 0$ for $|z| \geq 1$. We shall call systems satisfying this condition minimum phase systems.

Note that in order to guarantee stability of the closed-loop system for a nonminimum phase system, a nonzero weighting matrix Λ is required.

Finally, an interpretation of the MPE control law 4.2 in terms of the GLCL defined by eqn. 2.35 (p. 33) is given. Note from eqn. 3.42 (p. 72) for $\overline{K}_R(q) = K_R(q)$, that eqn. 4.6 describes a FIR filter, i.e., the filter transfer matrix is of the from

$$\tilde{K}_R(z^{-1}) = z^{-k}K_R(z) = K_0 + K_1 z^{-1} + \cdots + K_{k-1}z^{-k+1}.$$

The control law 4.2 can be interpreted as the GLCL for a system augmented by the filter $\tilde{K}_R(z^{-1})$ which introduces poles at the origin of z plane into the open-loop system (see fig. 4.1).

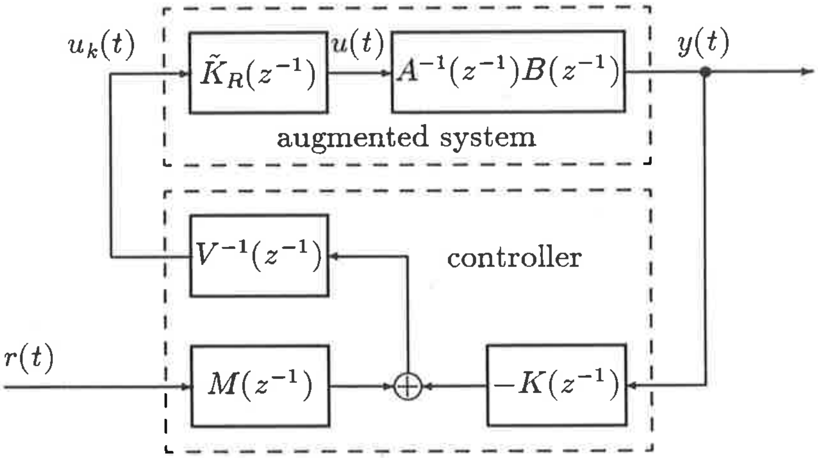

Figure 4.1: The MPE controller for the augmented system.

Comment 4.1 *The properties which characterize an interactor matrix (see Remark 2.1, p. 42) were embedded in the definition of the RNI matrix (see Definition 2.4, p. 45).*

The first property, given by eqn. 2.50 (p. 45), ensures that the coefficient matrix V_0 of $V(p)$ in eqn. 4.2 is nonsingular. Hence, in view of [77, Corollary, p. 343], the controller 4.2 is causal and the control signal $u(t)$ can be determined uniquely (for a given RNI matrix) as a function of past values of the auxiliary control signal.

The second property, given by eqn. 2.51 (p. 45), ensures that the poles introduced into the augmented system are stable, i.e., that the control sequence $\{u(t)\}$ is bounded.

4.2 Self-tuning minimum prediction error control with the on-line calculation of the right nilpotent interactor matrix.

In this section:

- the motivation for development of indirect self-tuning controllers is presented;
- the indirect self-tuning MPE controller, employing the algorithm for the on-line calculation of the RNI matrix, is introduced;
- the self tuner is compared with other self-tuning controllers;
- simulation studies illustrate the performance of the self tuner.

The *indirect self-tuning MPE controller*, involving the *on-line* calculation of the RNI matrix from the estimates of the DARMA model, is introduced in this section. This self tuner will be referred to as ST-MPE-RNI controller.

The reasons for development of the indirect self tuner are as follows.

Firstly, the algorithms developed for the calculation of the RNI matrix from the DARMA model in section 2.3 can be employed for the on-line calculation of the interactor matrix from the estimates of the DARMA model. Therefore, the assumption of the prior knowledge (complete or incomplete) of the interactor matrix is removed. Such assumptions are usually required in direct adaptive multivariable strategies based on minimization of the single-stage cost function [67] (see approaches 1, 2 and 3 in classification on p. 38). The development of the ST-MPE-RNI controller extends the classification of assumptions concerning prior knowledge of the interactor matrices in self-tuning control (see approach 5 (b), p. 39).

Secondly [39]:

> ... indirect strategies are more likely to lead to a reduction in the number of parameters to be estimated. ... Convergence properties for typical

> estimation schemes such as sequential least squares deteriorate rapidly when more than 10-12 parameters are estimated.

The only parameters to be estimated in the ST-MPE-RNI controller are those of the DARMA model.

Thirdly [39]:

> It is natural to expect prior knowledge regarding physical quantities in a system to be more easily mapped into prior knowledge of parameters in an input-output model [1] of the process than into prior knowledge regarding controller parameters. Simplification of the estimation problem by use of the prior knowledge is critical to the practical application of multivariable adaptive control.

The ST-MPE-RNI strategy is introduced below. For this purpose let us make the following assumptions: Assumption 3.1 (p. 58), 3.3 (p. 59), and

Assumption 4.1 *The stability conditions of Lemma 4.2 (p. 122) are satisfied.*

Prior knowledge of the system is restricted to

Assumption 4.2 *An upper bound $\hat{n}$ for the degree n of the polynomial matrices $A(p)$ and $B(p)$ of the system DARMA model is known (see eqns. 3.1 to 3.3, p. 59).*

Then the indirect **ST-MPE-RNI controller** involving a recursive parameter estimator consists of the following steps which are performed at every sample instant t:

step 1. calculate estimates $\widehat{A}_i$, $\widehat{B}_i$ of the coefficient matrices A_i, B_i of the DARMA model using, say, the simplified multivariable RLS algorithm given by eqns. E.18 to E.22, Appendix E ($i = 1, \ldots, \hat{n}$);

[1] i.e., DARMA model

step 2. calculate the RNI matrix $\widehat{K}_R(q)$ [2] from the estimate $\widehat{B}(p)$ of $B(p)$, using the algorithm for calculation of the RNI matrix from the DARMA model (see Theorem 2.1, p. 47);

step 3. calculate polynomial matrices $\widehat{F}(p)$ (defined in eqn. 3.46) and $\widehat{\gamma}(p)$ (defined in eqn. 3.47) from the polynomial equation 3.44, and $\widehat{\delta}(p)$ (defined by eqn. 3.48) from eqn. 3.45; this is equivalent to calculation of the corresponding matrices of coefficients $\widehat{F}$, $\widehat{\gamma}$, and $\widehat{\delta}$ using the algorithm given in Remark 3.3 (p. 74);

step 4. calculate the auxiliary control signal $u_k(t)$ from eqn. 4.2;

step 5. calculate the control signal $u(t)$ from eqn. 4.6.

The macro XSMP, which is written in MATLAB commands and implements the ST-MPE-RNI controller, is given in Appendix D.

In step 2 of the ST-MPE-RNI strategy, the RNI matrix is calculated using the algorithm presented in subsection 2.3.3.

Aside 4.1 *Alternatively, the algorithm for the calculation of the LNI matrix can be used to determine the RNI matrix, as discussed in subsection 2.3.4. This approach requires, however, additional polynomial matrix transpositions (see fig. 2.8 and discussion on pp. 55-56). Furthermore, transposition of the matrix $\mathcal{N}^{(t)}(p)$ resulting from the t-th iteration of the algorithm calculating $K_L(q)$, is needed to obtain matrix $\mathcal{B}^{(t)}(p) = (\mathcal{N}^{(t)}(p))'$ required for evaluation of the polynomial matrix $\widehat{\delta}(p)$ from eqn. 3.55 (p. 74).*

Calculation of the auxiliary control signal $u_k(t)$ in step 4 involves inversion of the matrix $\widehat{V}_0 = (\widehat{\delta}_0)'\Upsilon\widehat{\delta}_0 + (S_0)'\Lambda S_0$ (see eqn. 4.3). This matrix is nonsingular for all sample instants t, since matrix $\widehat{\delta}_0$ is nonsingular (see Comment 4.1).

Assumption 3.3 (p. 59) implies that the polynomial matrix $B(p)$ of the DARMA model is nonsingular. This, in turn, implies that the matrix $B = [B_1\ B_2\ \ldots\ B_n]$ of coefficients of $B(p)$ has full rank. However, the estimate $\widehat{B}$ of B resulting from

[2] The symbol "$\widehat{\ }$" is used to differentiate results of computations based on the estimates of parameters of the DARMA model from results which are based on true parameters.

step 1 of the ST-MPE-RNI strategy is not necessarily of full rank. If rank $\widehat{B} < m$ then the corresponding polynomial matrix $\widehat{B}(z^{-1}) = \widehat{B}_1 z^{-1} + \widehat{B}_2 z^{-2} + \cdots + \widehat{B}_n z^{-n}$ is singular [112, Lemma G-2, p. 615]. Hence, the estimate of the system transfer matrix $\widehat{H}_{y,u}(z) = \widehat{A}^{-1}(z^{-1})\widehat{B}(z^{-1})$ is singular and the RNI matrix is not defined (see Definition 2.4, p. 45). Consequently, the algorithm for the calculation of the RNI matrix will not terminate (see proof of Theorem 2.1, part 2). In order to guarantee nonsingularity of $\widehat{B}$ for all sample instants t, the constrained parameter estimation algorithms can be used [9]. As an alternative approach, if the upper bound on the degree of the RNI matrix is known, the algorithm evaluating the RNI matrix could be terminated after number of steps equal to this bound, and the RNI matrix from the previous sample instant could be used. (Note also that the ST-MPE-RNI scheme must be started with the full-rank initial estimate $\widehat{B}(0)$ of the matrix of coefficients B of the polynomial matrix $B(p)$.)

The ST-MPE-RNI strategy involves estimation of parameters of the DARMA model. On the other hand, the self-tuning controllers proposed in [35,32,34] [9, section 6.3.3]) involve estimation of parameters of the k-step-ahead predictor. The latter strategies require some modifications of the estimation algorithm to guarantee that the estimate of the *coefficient matrix* associated with the value of the control signal (or the auxiliary control signal) at time t, is nonsingular. (This means that the estimate $\widehat{\delta}_0$ must be kept nonsingular for the strategy of [35].) Possible modifications involve constrained parameter estimation algorithms [9] or the modified projection algorithm [32] [9, Lemma 6.3.1, p. 204]. Heuristically, however, the reduction in rank of the $m \times mn$ real matrix $\widehat{B}$ for the ST-MPE-RNI strategy at any sample instant t is less likely than a singularity of the $m \times m$ real coefficient matrix associated the current value of the control signal (i.e., $\widehat{\delta}_0$ for the strategy of [35]).

Some further aspects of the ST-MPE-RNI strategy will be now compared with two other schemes.

The MPE self tuner proposed in [35] requires prior knowledge of the (unique) right interactor matrix $\xi_R(q)$. This restrictive assumption is eliminated in the ST-MPE-RNI scheme but more involved computations are necessary to calculate the RNI matrix. The algorithm of [35] estimates parameters of the k-step-ahead predictor; the number of estimated parameters is greater than that of the ST-MPE-RNI scheme by km^2, which possibly leads to a slower convergence rate than that for the ST-MPE-RNI strategy.

The ST-MPE-RNI strategy requires the same prior knowledge of the system (Assumption 4.2) as the dead-beat self tuner introduced in [39]. Both strategies are indirect, hence the number of the estimated parameters, and therefore the expected convergence rate, are the same. The implementation of the dead-beat self tuner [39] requires two *polynomial matrix divisions* at every sample instant for calculation of the interactor matrix and control law. On the other hand, the ST-MPE-RNI scheme involves *multiplication of polynomial matrices* in determination of the RNI matrix and the control law. However, factorization of real $m \times m$ matrices is required for calculation of the RNI matrix. The operations on polynomial matrices are represented in terms of operations on the corresponding matrices of coefficients.

The dead-beat controller of [39] is based on the left interactor matrix; the closed-loop system behaviour is described by $\xi_L(q)y(t) = r(t)$. In terms of the cost function it means that the squared norm of the tracking error defined as $\xi_L(q)y(t) - r(t)$ is minimized (cf. approach (i), p. 60). The ST-MPE-RNI scheme implies that $y(t+k) = r(t+k)$ for $\Lambda = 0$ (see eqn. 4.10) and minimizes the squared norm of the tracking error defined as $y(t+k)-r(t+k)$ (see eqn. 4.1). The difference between the two approaches is not significant since the reference sequence could be prefiltered by the interactor matrix $\xi_L(q)$ for the dead-beat controller [39] leading to perfect output tracking for both strategies after the transient due to the dynamics of interactor matrix dies away. Note that future values of the reference sequence are required for such a prefiltering, i.e., $\xi_L(q)r(t) = \xi_0 r(t+k) + \cdots + \xi_{k-1}r(t+1)$. The future value $r(t+k)$ of the reference sequence is required also by the control law 4.2 embedded in the ST-MPE-RNI strategy. However, future values of $\{r(t)\}$ are not always known. In the latter case, their predictions can be used, e.g., future values of the reference sequence can be set equal to the current value $r(t)$. Then the output responses resulting from both controllers might differ significantly. The ST-MPE-RNI controller introduces the delay of k sample intervals in tracking (see example 4.1). On the other hand, changes in the reference sequence produce not only delay in tracking, but also variations of system outputs if the self tuner of [39] is applied to a system with nondiagonal interactor matrix (see example 5.1.1 (p. 163) for the illustration of a similar effect). The latter effect is due to the coupling introduced by the triangular factor of $\xi_L(q)$.

Finally, the effect of deterministic disturbances, for which the model of the generator is incorporated in the system DARMA model, is eliminated by the ST-MPE-RNI controller. On the other hand, both strategies [35,39] were designed for systems without disturbances.

The performance of the ST-MPE-RNI controller is demonstrated in the following example.

Example 4.1.

The purpose of this example is to illustrate the performance of the ST-MPE-RNI strategy applied to a system for which the interactor matrix is nondiagonal.

Let us consider a two-input, two-output ($m = 2$) plant given by the following transfer matrix:

$$H_{y,u}(z) = \begin{bmatrix} \frac{1}{z+0.1} & \frac{1}{z+0.2} \\ \frac{1}{z+0.3} & \frac{1}{z+0.4} \end{bmatrix}.$$

The corresponding unique right interactor matrix (see Definition 2.2, p. 37) has a general upper triangular form:

$$\xi_R(z) = \begin{bmatrix} z & -z^3 + 0.1z^2 \\ 0 & z^3 \end{bmatrix}.$$

Prior knowledge of this interactor matrix would be required to implement the MPE strategy of [35].

The DARMA model corresponding to the above transfer matrix $H_{y,u}(z)$ is given by (see eqns. 3.1 to 3.3, p. 59)

$$A(p) \;=\; [I_2 \; A_1 \; A_2](P_2 \otimes I_2) = \begin{bmatrix} 1 & 0 & 0.3 & 0 & 0.02 & 0 \\ 0 & 1 & 0 & 0.7 & 0 & 0.12 \end{bmatrix} (P_2 \otimes I_2),$$

$$B(p) \;=\; p\,[B_1\;B_2]\,(P_1 \otimes I_2) = p\begin{bmatrix} 1 & 1 & 0.2 & 0.1 \\ 1 & 1 & 0.4 & 0.3 \end{bmatrix}(P_1 \otimes I_2),$$

i.e., $n = 2$. The RNI matrix, calculated for this plant from the above DARMA model using the algorithm implemented with QR factorization (see Theorem 2.1, p. 47), is

$$\begin{aligned} K_R(z) &= [K_0\;K_1\;K_2]\left(\left[z^3\;z^2\;z\right]' \otimes I_2\right) \\ &= \begin{bmatrix} 0 & -0.7062 & 0.0353 & 0.0353 & 0.7062 & 0 \\ 0 & 0.7062 & -0.0353 & 0.0353 & 0.7062 & 0 \end{bmatrix}\left(\left[z^3\;z^2\;z\right]' \otimes I_2\right), \end{aligned} \qquad (4.12)$$

i.e., the degree of the RNI matrix is $k = 3$.

The reference signal $r(t)$ was chosen as square wave with amplitude 10 units and period equal 20 sample intervals. The calculation of $u_k(t)$ from the control law 4.2 requires knowledge of the value of the reference sequence $k = 3$ samples ahead. However, it was assumed that future values of $\{r(t)\}$ are unknown. The *fixed reference sequence model* was employed instead [3], i.e., the future values of $\{r(t)\}$ were set equal to the current value $r(t)$. Limits on the amplitudes for each component i of the control and auxiliary control signal vectors were chosen as $|u(t)|_i \leq 175$, $|u_k(t)|_i \leq 225$ $(i = 1,2)$, to avoid excessive control effort during the initial estimation phase. The weighting matrices of the control law were set to $\Upsilon = I_2$, $\Lambda = 0$, which satisfies Assumption 4.1 (p. 126). The simplified RLS algorithm was used (see Appendix E) with the initial covariance matrix $P(-1) = 10^4 I_8$ and constant forgetting factor $\alpha(t) = 0.95$. The initial parameter estimates were:

$$\begin{aligned} \widehat{\Theta}(0) &= \begin{bmatrix} \widehat{A}_1(0) & \widehat{A}_2(0) & \widehat{B}_1(0) & \widehat{B}_2(0) \end{bmatrix}' = \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}', \end{aligned}$$

i.e., the upper bound on the degree of polynomials of the estimated DARMA model was chosen equal to the true degree, $\widehat{n} = n = 2$.

The ST-MPE-RNI controller was implemented with the algorithm for the on-line

[3] For other approaches to describe future values of the reference sequence see [27].

calculation of the RNI matrix from the estimates of the DARMA model with QR matrix decomposition (see macro XSMP in Appendix D).

The input-output behaviour of the closed-loop system is shown in fig. 4.2. It can be observed that tracking of the reference sequence with delay of three sample intervals is achieved after about 30 samples. The delay in tracking (equal to the degree of the RNI matrix) results from the use of fixed reference sequence model.

The trajectories of the estimates $\widehat{A}_i$ and $\widehat{B}_i$ $(i = 1,2)$ of the coefficient matrices of the DARMA model are shown in figs. 4.3 and 4.4, respectively.

The RNI matrix $\widehat{K}_R(z)$ was calculated on-line from the estimate $\widehat{B}$. In the implementation of the algorithm calculating the RNI matrix, the first iteration was omitted and the algorithm was initialized with $K_R^{(1)}(z) = zI_2$ (see Comment 2.2, p. 55). The coefficient matrices $\widehat{K}_0$, $\widehat{K}_1$, and $\widehat{K}_2$ of $\widehat{K}_R(z)$ are shown [4] in fig. 4.5 (see eqn. 4.12 for their true values).

The estimator was initialized with a nonsingular coefficient matrix $\widehat{B}_1(0) = I_2$. This resulted in the RNI matrix $\widehat{K}_R(z) = zI_2$ during the initial estimation phase (up to sample instant 15). During this phase the estimates were not accurate enough to lead to the actual RNI matrix 4.12; in fact, the estimates differed significantly from their true values and as a result the RNI matrix remained fixed at its initial value $\widehat{K}_R(z) = zI_2$. This, in turn, resulted in poor tracking, but without excessive excursions of system inputs and outputs from their desired values.

The system estimates converged closely to the true values after 15 sample instants. During the following period of samples 15 to 22, the estimate $\widehat{B}_1$ became singular leading to the RNI matrix of degree greater than 1. The form of $\widehat{K}_R(z)$ depended on the estimate $\widehat{B}_2$ which was now involved in calculation of the RNI matrix. The degree of the RNI matrix changed as $\widehat{k} = 1,3,2,3,1,3$, reaching the true value $\widehat{k} = k = 3$ after 22 samples. These switchings of the degree of the RNI matrix resulted in the undesirable performance of the closed-loop system between sample instants 15 to 25. Hence, the phase of the worst input-output behaviour was associated with the

[4] Note that only the values plotted at the integer sample instants are relevant.

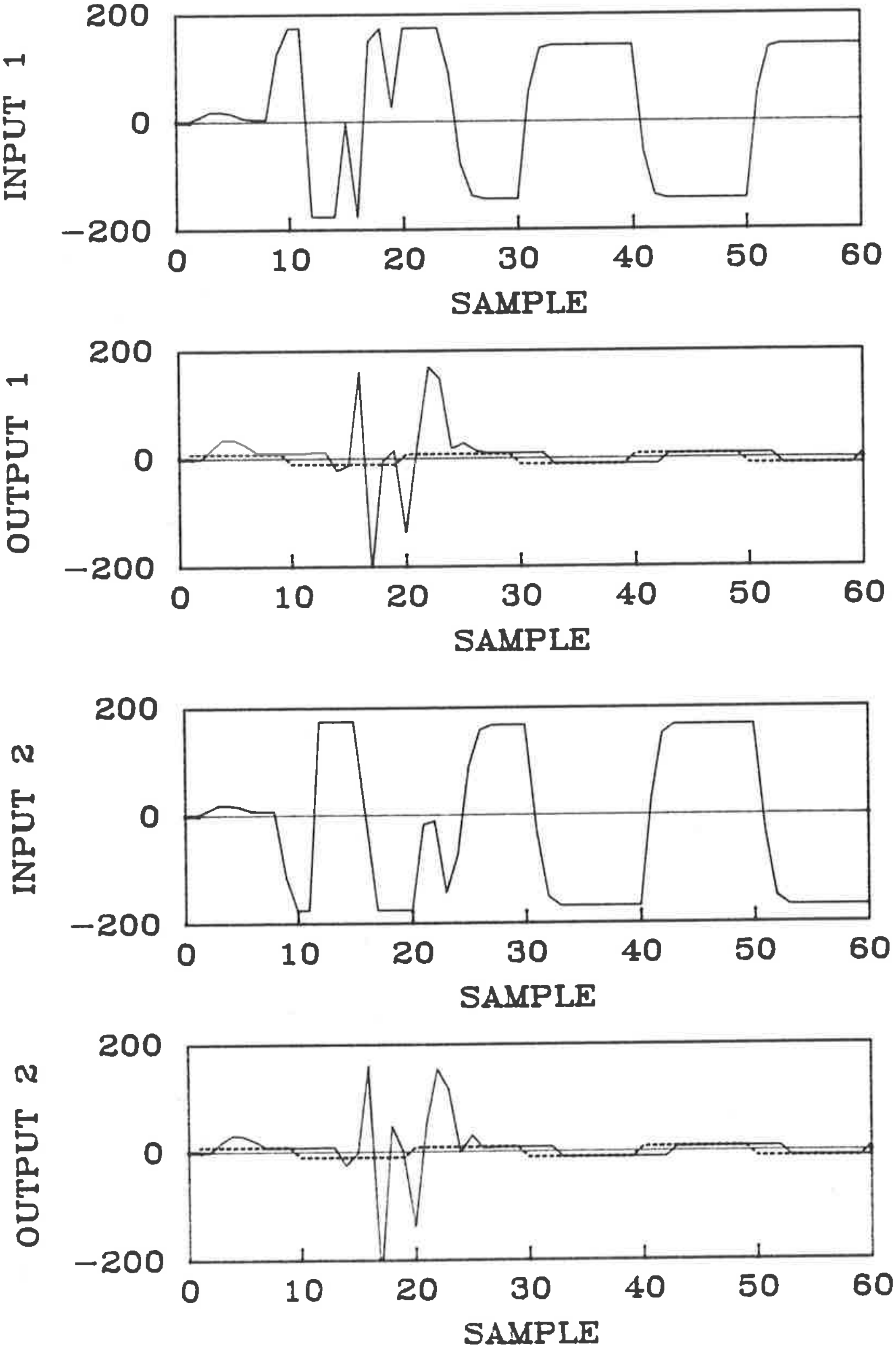

Figure 4.2: ST-MPE-RNI controller: inputs, outputs and reference sequences.

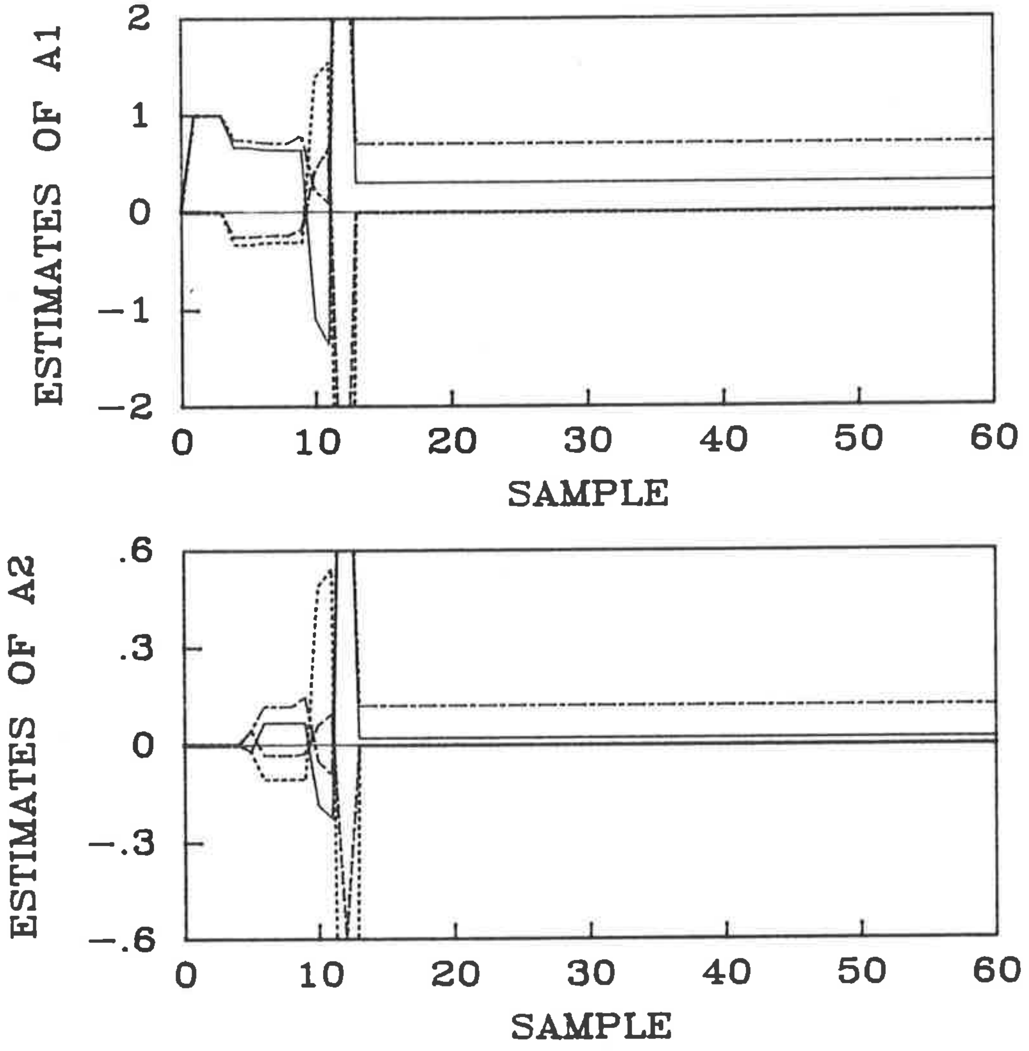

Figure 4.3: ST-MPE-RNI controller: estimates of the coefficient matrices A_1 and A_2.

estimates of parameters of the DARMA model being close to the true values, but not sufficiently accurate to lead to the actual RNI matrix.

The algorithms developed in section 2.3 for the calculation of nilpotent interactor matrices involve testing of the rank of a real (coefficient) matrix in step 1. In practical self-tuning applications, it is unlikely that the singularity of the tested matrix would ever occur, even if the true coefficient matrix were singular. This is because an estimate of the true coefficient matrix is used in calculations. Therefore, the question arises whether to treat the near-singularity as nonsingularity or singularity.

The answer is provided by the analysis of robustness of controllers developed for

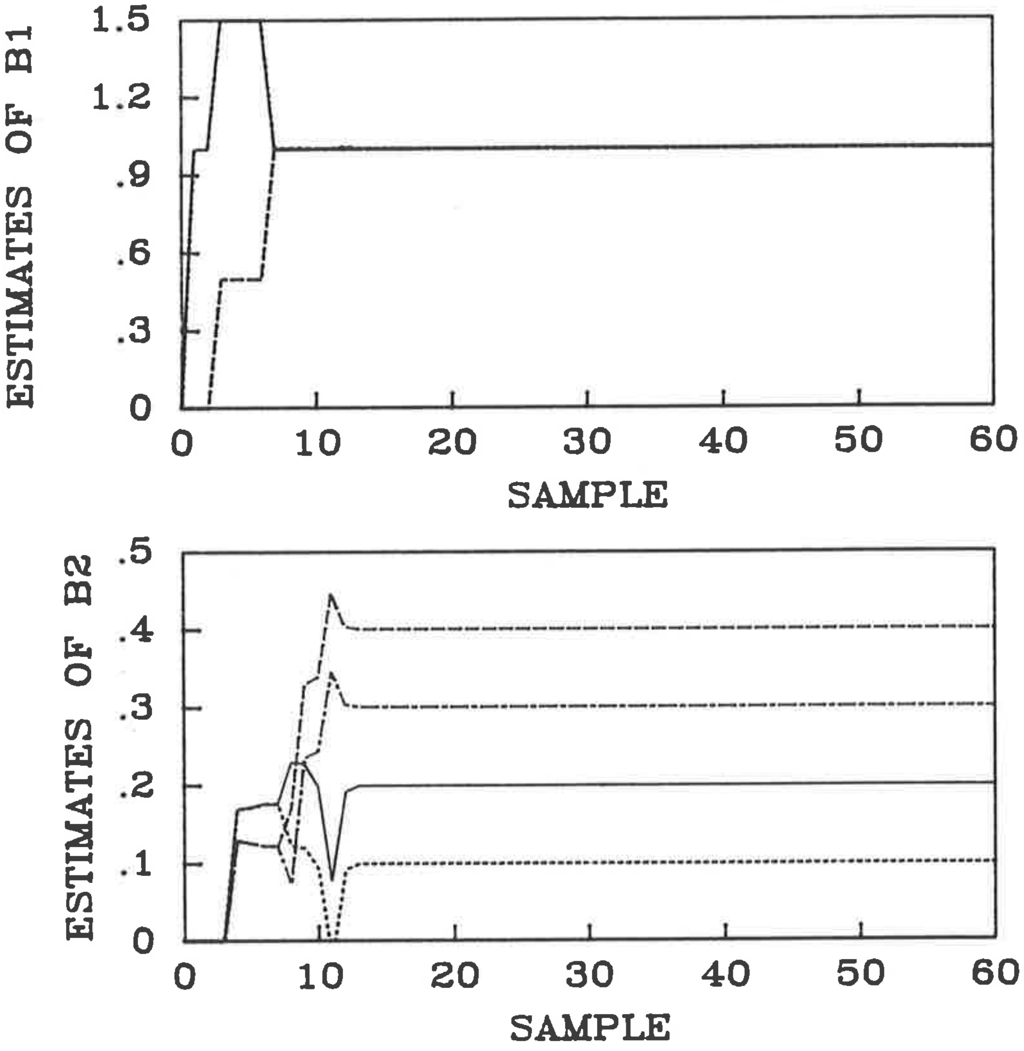

Figure 4.4: ST-MPE-RNI controller: estimates of the coefficient matrices B_1 and B_2.

systems having a nondiagonal unique interactor but assuming that the interactor is diagonal, presented in [98] (see Comment 2.1, p. 38) [5]. Namely, for the design of robust controllers, the near-singularity of the estimate of the coefficient matrix tested in step 1 of the algorithm calculating the RNI matrix should be treated as singularity. This was achieved here by truncating the computer arithmetic with the MATLAB command CHOP(P) with P=8, which sets to zero the P least significant hexadecimal digits in the result of each floating point operation [110]. Thus the singularity of a

[5] This is a very similar problem to that considered here since the question is whether the near-linear dependency of certain vectors, which is tested in the algorithm [33] for the calculation of the unique interactor matrix, should be treated as exact linear dependency or ignored.

matrix which would be considered nonsingular if full precision arithmetic were used, is detected. (Furthermore, this command simulates a computer with a shorter word length.)

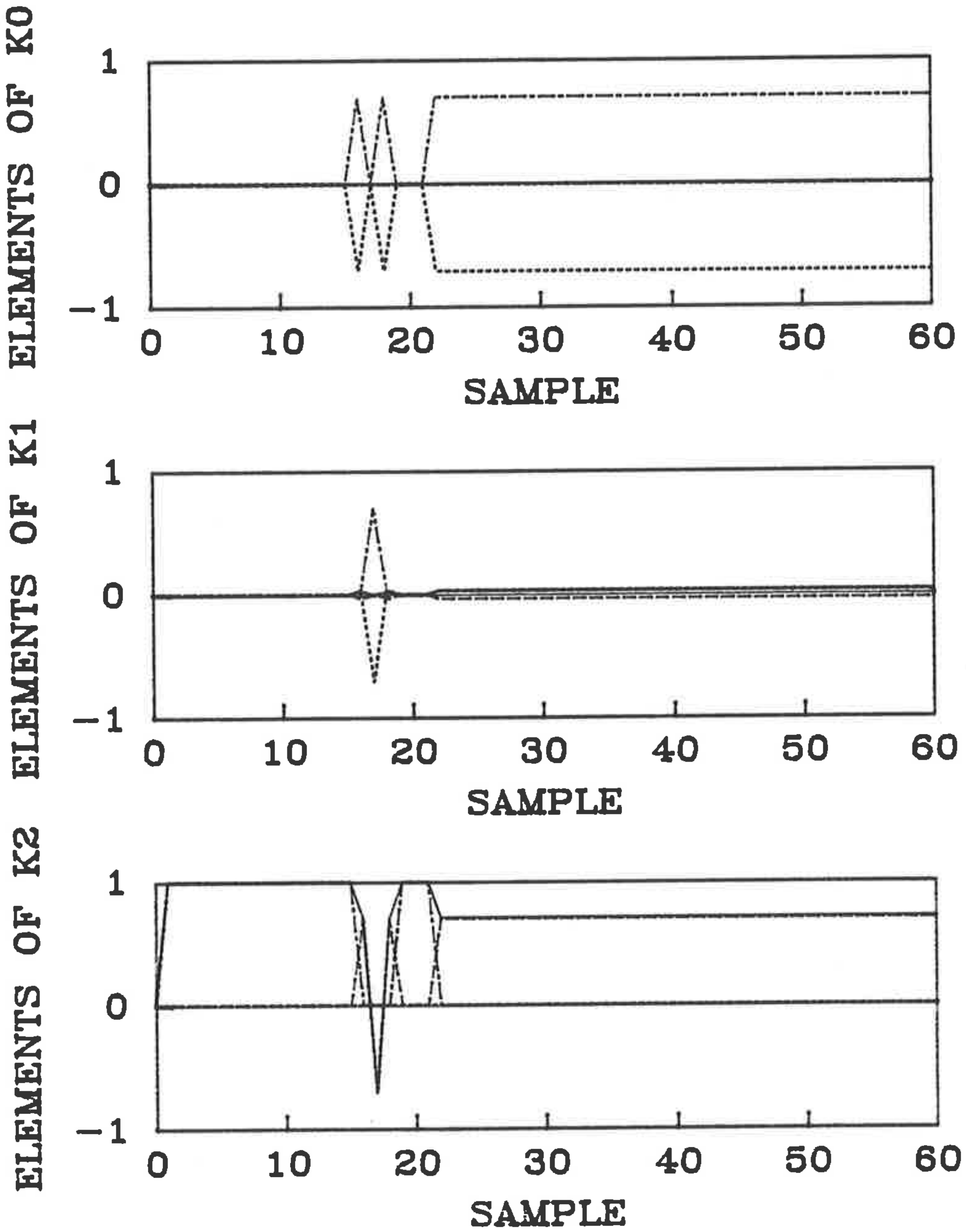

Figure 4.5: ST-MPE-RNI controller: coefficient matrices of the RNI matrix $\widehat{K}_R(z) = \widehat{K}_0 z^3 + \widehat{K}_1 z^2 + \widehat{K}_2 z$ calculated on-line.

4.3 Application of self-tuning minimum prediction error control to robotics.

In this section:

- some background to the control of robot manipulators is presented;
- the ST-MPE-RNI scheme is applied to control a two-link planar robot.

With the growing interest in robotics, the problem of controlling mechanical linkage systems has recently received considerable attention [141]. The dynamic control of industrial manipulators involves determination of the input torques (or forces) such that the (angular) positions and velocities of the manipulator joints track a desired trajectory.

The dynamic behaviour of the n-link, n-joint robot manipulator is characterized by a system of coupled nonlinear differential equations, such as the Euler-Lagrange dynamic equations of motion [141]

$$M(\theta)\ddot{\theta} + N(\theta, \dot{\theta}) + G(\theta) = T,$$

where θ, $\dot{\theta}$, $\ddot{\theta}$ are the $n \times 1$ vectors of joint angles, velocities, and accelerations, respectively; furthermore, $G(\theta)$ is the $n \times 1$ gravitational torque vector, $N(\theta, \dot{\theta})$ is the $n \times 1$ Coriolis and centrifugal torque vector, $M(\theta)$ is the $n \times n$ inertia matrix, and T is the $n \times 1$ vector of control torque. The elements of G, N, and M are highly complex nonlinear functions of θ and $\dot{\theta}$.

The problem of controlling such nonlinear dynamics (relating joint angles to motor torques), has been tackled using optimal control theory and leads to complicated nonlinear strategies (see [142,143,50] for references). Alternatively, simpler linear multivariable strategies were proposed [144]. Such strategies often yield satisfactory performance for stabilizing robot manipulators, but for the case of reference following, they are not robust enough to ensure good tracking properties over the operating range and variable loads. In this case, the linearized robot model and the corresponding linear controller must be updated for various operating points [144].

Complexity of nonlinear controllers and unsatisfactory performance of linear control led to the application of adaptive techniques to enhance the performance, robustness and decrease computational effort of digital controllers in robotics [142,143,50] [145] (see also bibliography in [5]). One of the possible approaches to self-tuning control of robot manipulators is based on the approximation of the nonlinear robot model by a linear discrete-time model, assumed to be of the ARMAX or DARMA form [142,94]. This approach relies on the ability of the adaptive controller to track time-varying process parameters. The variations of the process parameters correspond to the changing operating points along the reference trajectory, thus constituting an approximation to the nonlinear model at various operating points. The following simulation of robot control with the ST-MPE-RNI scheme is based on the same approach.

We shall consider the problem of controlling a two-link planar robot, the schematic diagram of which is shown in fig. 4.6. It will be assumed that the robot joints

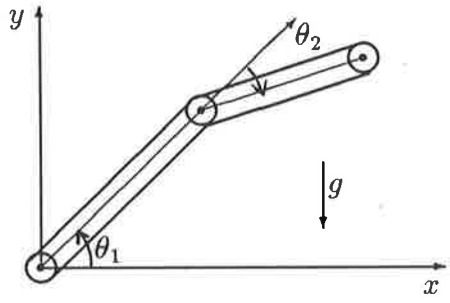

Figure 4.6: Two-link planar manipulator in a vertical plane.

are frictionless and the robot links are uniform with centers of gravity located at midlength. Furthermore, let us assume that the links depicted in fig. 4.6 correspond to links 2 and 3 of the Unimation PUMA 560 robot. The lengths of the links are $l_1 = l_2 = 0.432$ m, and masses of the links are $m_1 = 15.91$ kg, $m_2 = 11.36$ kg.

In order to determine the DARMA model representing the linearized discrete-time model of the PUMA manipulator, let us consider the continuous-time linearized state-space model derived in [144]. The state vector $x(t)$ consists of the angular joint positions $\theta(t) = [\theta_1(t)\ \theta_2(t)]'$ (in radians), and velocities $\dot{\theta}(t)$, the vector of inputs

$u(t)$ consists of the joint torques [Nm], and the vector of outputs $y(t)$ consists of the angular positions of joints [rad]

$$x(t) = \begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \end{bmatrix}, \quad u(t) = \begin{bmatrix} \text{torque}_1(t) \\ \text{torque}_2(t) \end{bmatrix}, \quad y(t) = \theta(t).$$

We choose the sampling period $T_s = 0.25$ s, and two operating points, for which the linearization is performed, as

$$Q_1 = \{\theta_1 = -\frac{\pi}{2},\ \theta_2 = \dot{\theta}_1 = \dot{\theta}_2 = 0\}, \tag{4.13}$$

$$Q_2 = \{\theta_1 = -0.2 - \frac{\pi}{2},\ \theta_2 = 0.5,\ \dot{\theta}_1 = \dot{\theta}_2 = 0\}. \tag{4.14}$$

(Note that Q_1 corresponds to the "arm-down" position.) Discretization of the state-space models (see [112, Appendix D]) linearized at the operating points Q_1 and Q_2 leads to the discrete-time state-space model 2.2 (p. 20), where matrices A_s, B_s and C are given in table 4.1. The corresponding DARMA model 3.1 (p. 59) is found using algebraic methods for transformation from the state-space model to the DARMA model developed in [79, chapter 5]. The coefficient matrices of the DARMA models corresponding to the operating points Q_1 and Q_2 are given in table 4.1.

The two-link manipulator was simulated using the DARMA model augmented by the noise $\omega(t)$ representing modeling errors (i.e., ARMAX model with $C(p) = I$) [142]

$$A(p)y(t) = B(p)u(t) + \omega(t). \tag{4.15}$$

The term $\omega(t)$ is a (vector) zero mean white gaussian noise with covariance matrix $\Sigma = 10^{-6} I_2$ rad^2 (see eqns. 2.4 to 2.6, p. 21). The manipulator was simulated at the operating point Q_1 for the period $0 \leq t < 9.8$ seconds, and at Q_2 for $9.8 \leq t \leq 37.5$ seconds.

Model	at the operating point Q_1	at the operating point Q_2
A_s	$\begin{matrix} .3126 & .3609 & .1850 & .0415 \\ .6257 & -.6497 & .0720 & .0743 \\ -3.8577 & .5158 & .3126 & .3609 \\ .8960 & -5.2357 & .6257 & -.6497 \end{matrix}$	$\begin{matrix} .3381 & .2882 & .1896 & .0305 \\ .5211 & -.4345 & .0551 & .1080 \\ -4.1436 & .9121 & .3381 & .2882 \\ 1.6491 & -6.5886 & .5211 & -.4345 \end{matrix}$
B_s	$\begin{matrix} .0129 & -.0277 \\ -.0277 & .0964 \\ .0540 & -.0739 \\ -.0744 & .2931 \end{matrix}$	$\begin{matrix} .0118 & -.0244 \\ -.0244 & .0868 \\ .0630 & -.1027 \\ -.1027 & .3894 \end{matrix}$
C	$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$	$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$
$[A_1\ A_2]$	$\begin{matrix} -.6251 & -.7218 & 1 & 0 \\ -1.2515 & 1.2995 & 0 & 1 \end{matrix}$	$\begin{matrix} -.6761 & -.5764 & 1 & 0 \\ -1.0422 & .8690 & 0 & 1 \end{matrix}$
$[B_1\ B_2]$	$\begin{matrix} .0129 & -.0277 & .0129 & -.0277 \\ -.0277 & .0964 & -.0277 & .0964 \end{matrix}$	$\begin{matrix} .0118 & -.0244 & .0118 & -.0244 \\ -.0244 & .0868 & -.0244 & .0868 \end{matrix}$
z_{pi}	$0.5082 \pm 0.8612i$, $-0.8414 \pm 0.5403i$	$0.4989 \pm 0.8666i$, $-0.5954 \pm 0.8034i$
z_{zi}	$-1, -1$	$-1, -1$

Table 4.1: Matrices of the state-space and DARMA models of the robot manipulator linearized at the operating points Q_1 and Q_2 ($T_s = 0.25$ s).

The reference signal $r(t) = [r_1(t)\ r_2(t)]'$ (in radians) was defined as follows

$$r_1(t) = \begin{cases} 0.2\sin(\frac{2\pi t}{12.5}) & \text{for } 0 \leq t < 12.5, \text{ and } 25.0 \leq t \leq 37.5 \text{ s} \\ 0.4\sin(\frac{2\pi t}{12.5}) & \text{for } 12.5 \leq t < 25.0 \text{ s} \end{cases} \tag{4.16}$$

$$r_2(t) = \begin{cases} 0.08t & \text{for } 0 \leq t < 6.25, \text{ and } 25.0 \leq t \leq 37.5 \text{ s} \\ 0.5 & \text{for } 6.25 \leq t < 12.5 \text{ s} \\ -0.16t & \text{for } 12.5 \leq t < 18.75 \text{ s} \\ -0.5 & \text{for } 18.75 \leq t < 25.0 \text{ s.} \end{cases} \tag{4.17}$$

Assuming perfect tracking of the above reference sequence, the robot is at the operating point Q_1 at time $t = 0$ and at Q_2 at time $t = 9.8$ s.

Note that the system poles (denoted by z_{pi} in table 4.1) lie on the unit circle (see fig. 4.7), hence the robot is not stable at the operating points Q_1 and Q_2. Furthermore, roots of the equation $\det B(z^{-1}) = 0$ (denoted by z_{zi} in table 4.1) lie on the unit circle as well (see fig. 4.7). Hence, in order to satisfy the stability conditions of the closed-loop system with the ST-MPE-RNI controller, nonzero weighting of the auxiliary control signal must be used (see Assumption 4.1, p. 126). Otherwise, if $\Lambda = 0$, the stability condition given by eqn. 4.11 (p. 123) is not satisfied.

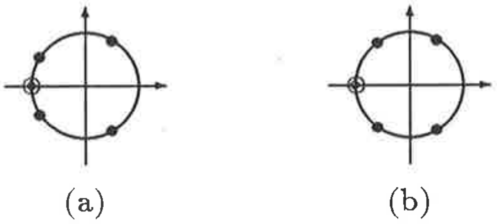

Figure 4.7: Poles (denoted by discs) and roots of $\det B(z^{-1}) = 0$ (denoted by circles) of the DARMA model of the two-link manipulator at the operating point: (a) Q_1, and (b) Q_2.

The parameters of the ST-MPE-RNI control law were chosen as $\Upsilon = I_2$, $S(p) = I_2 - pI_2$ and the auxiliary control signal weighting matrix

$$\Lambda = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.5 \end{bmatrix} 10^{-4}.$$

The simplified RLS algorithm was used with the initial covariance matrix $P(-1) = 10^4 I_8$ and constant forgetting factor $\alpha(t) = 0.95$. The initial estimates of the coefficient matrices of the DARMA model were $\widehat{A}_1(0) = \widehat{B}_1(0) = I_2$ and $\widehat{A}_2(0) = \widehat{B}_2(0) = 0$.

The input-output behaviour of the closed-loop system is shown in fig. 4.8. (The first joint angle $\theta_1(t)$ is plotted around constant level of $-\frac{\pi}{2}$ rad taken as a zero level in fig. 4.8.) It can be observed that the system performance is satisfactory after the initial tuning phase of about 30 sample periods. The second output $\theta_2(t)$ is disturbed slightly after sample 38, i.e., after the change of operating points. Afterwards both outputs follow the reference sequence closely. The future values of the reference sequence are approximated by their current values in the calculation of controls, i.e., the fixed reference sequence model is assumed. This explains the delay between outputs and reference sequence. Note that the λ-offset is eliminated: there is zero steady state error for $\theta_2(t)$ for sample instants 75 to 100.

The trajectories of the estimates $\widehat{A}_i$ and $\widehat{B}_i$ $(i = 1, 2)$ are shown in figs. 4.9 and 4.10, respectively. The parameter estimates converged to their true values after about 15 iterations. After the jump of parameters occured at sample instant 38 (due to the transition form Q_1 to Q_2), the estimates followed the change and then drifted around new true values. The most significant variations of parameter estimates were observed for $\widehat{A}_2$, for the true values of which there was no change at sample 38. The input-output performance was not affected by those drifts. The estimates converged again after about 100 iterations.

The forgetting factor was constant in this simulation. There is, however, a number of methods which improve the estimator capability of tracking time-varying parameters by covariance matrix modification or variable forgetting factors [146,11,147,148] [13] [9, chapters 3 and 6].

The RNI matrix $\widehat{K}_R(z)$ was calculated on-line from the estimate $\widehat{B}$ of the matrix B. In the implementation of the algorithm calculating the RNI matrix, the first iteration was omitted and the algorithm was initialized with $\widehat{K}_R^{(1)}(z) = zI_2$ (see Comment 2.2, p. 55). The estimate $\widehat{B}_1$ of the nonsingular coefficient matrix B_1 was nonsingular during the whole simulation run. Hence, a correct RNI matrix

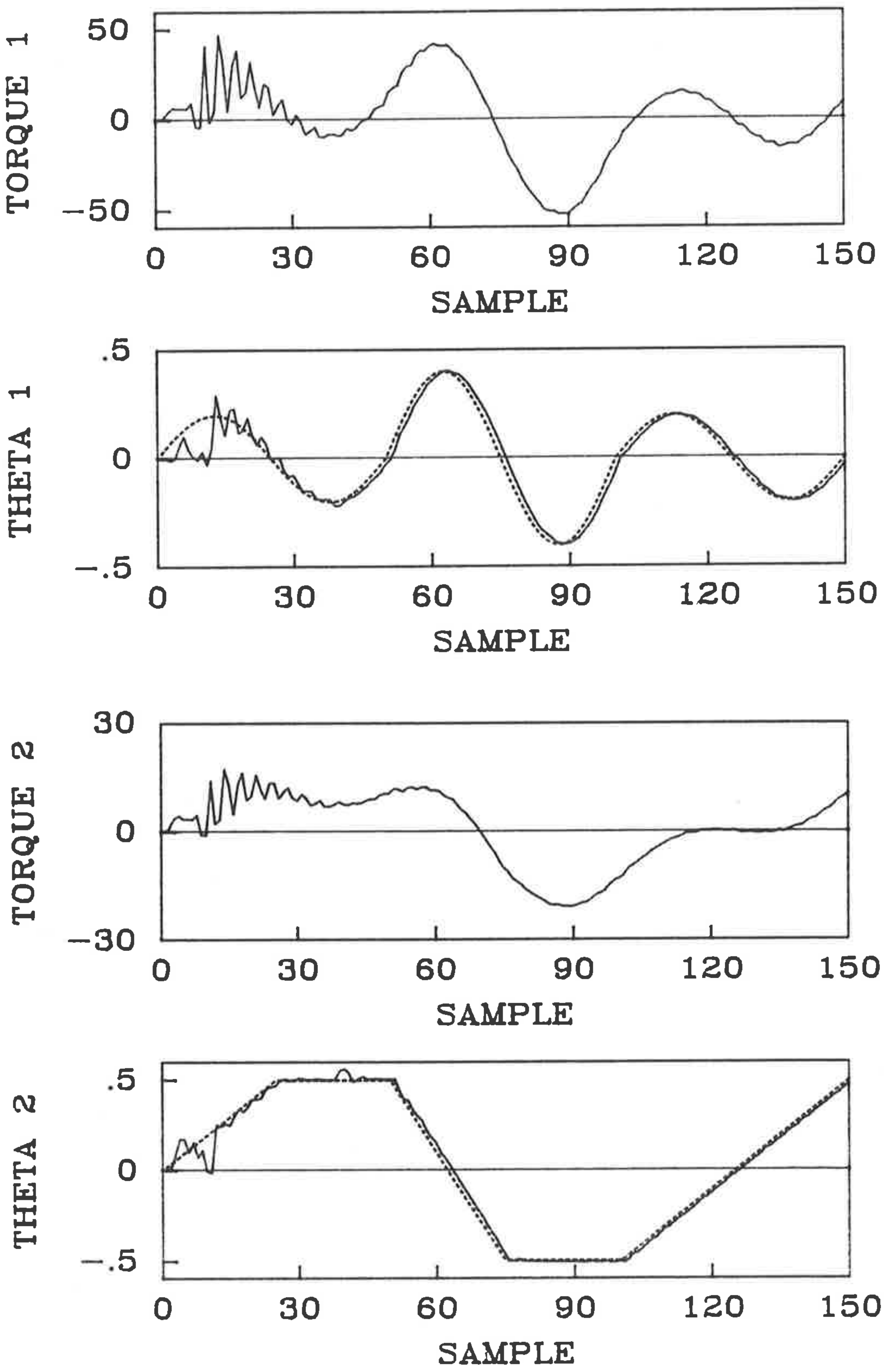

Figure 4.8: ST-MPE-RNI controller applied to the two-link robot manipulator: torques, angular joint positions $\theta_1(t)$ and $\theta_2(t)$, and reference sequences (shown with dotted lines).

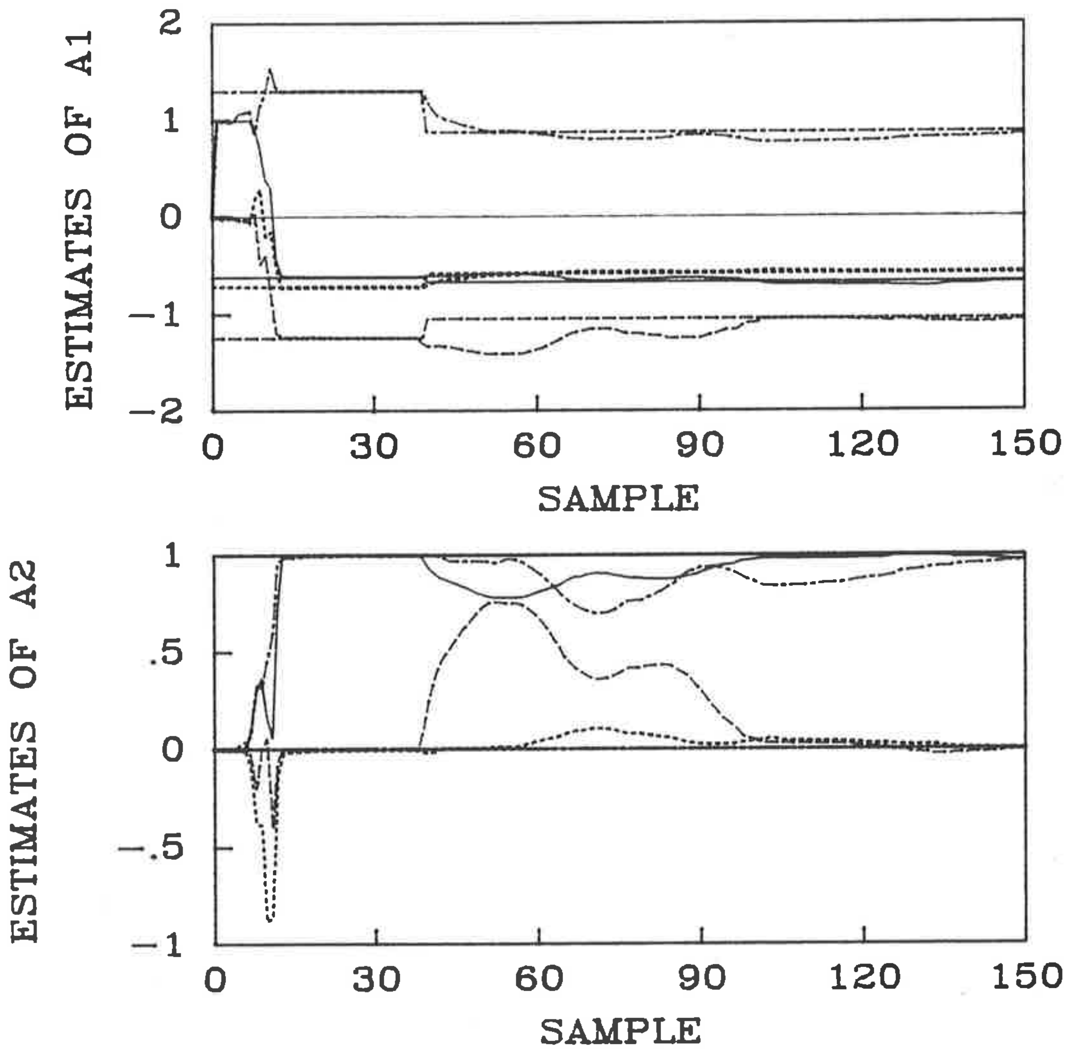

Figure 4.9: ST-MPE-RNI controller applied to the two-link robot manipulator: estimates and true values of the coefficient matrices A_1 and A_2.

$\widehat{K}_R(z) = zI_2$ was used during the simulation. The calculation of $\widehat{K}_R(z)$ was not affected either by the change of parameters or by the output noise.

It was pointed out in [6,5] that in order to avoid abuse of adaptive techniques employed to cope with the variations of the open-loop dynamics over the operating range, the possibility of robust nonadaptive control must be examined at first. For this purpose the nonadaptive MPE-RNI controller was simulated to assess its performance in application to control of the manipulator. The controller was determined for the known robot model linearized at the operating point Q_1. The parameters of the control law were chosen to be the same as for the self-tuning scheme. The

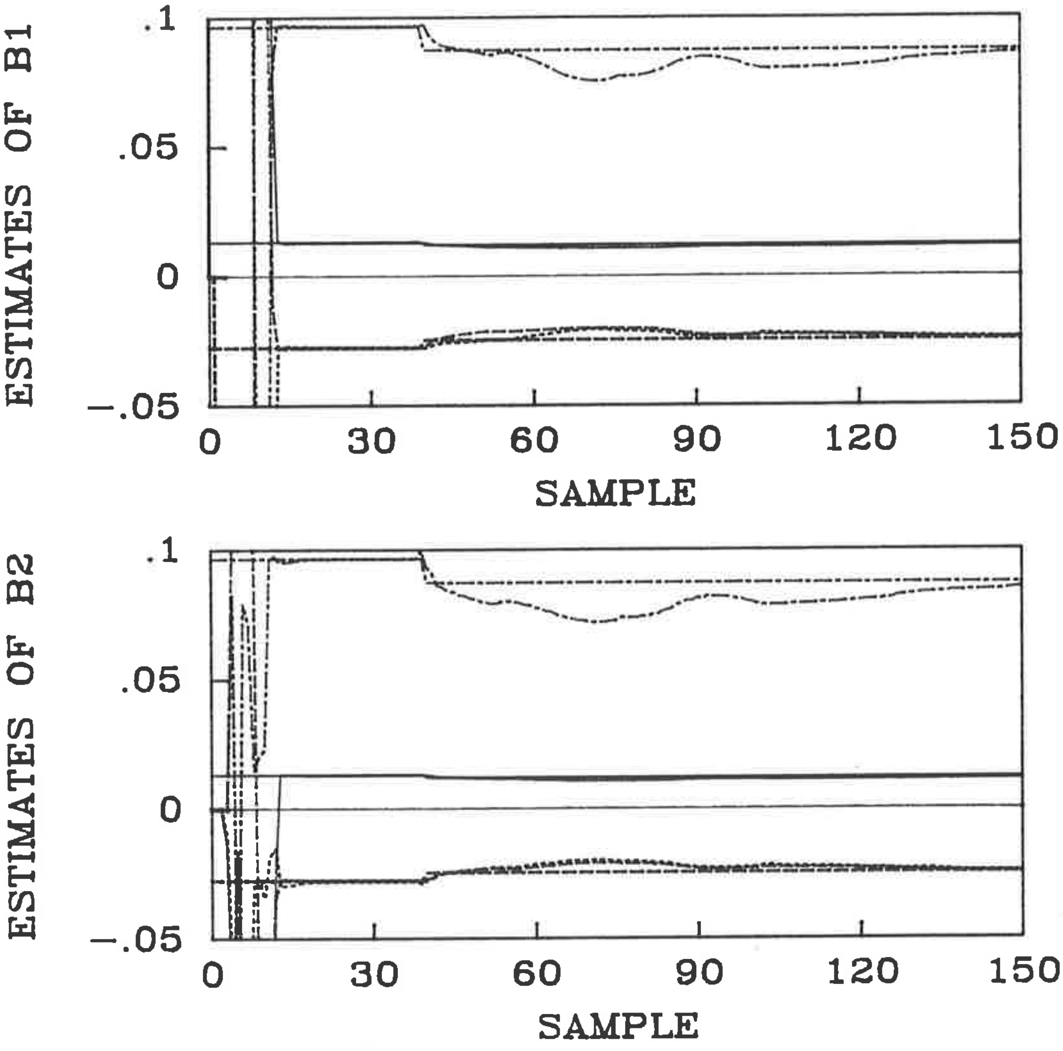

Figure 4.10: ST-MPE-RNI controller applied to the two-link robot manipulator: estimates and true values of the coefficient matrices B_1 and B_2.

coefficient matrices of polynomials $\gamma(p)$ and $\delta(p)$ resulting form eqns. 3.57 and 3.58 (p. 74) are $\gamma = [-A_1 \ -A_2]$, and $\delta = [B_1 \ B_2]$. Then the MPE-RNI control law is given by eqn. 4.2 with $V(p) = [B_1'\delta + \Lambda S]\,(P_1 \otimes I_2)$ where $S = [I_2 \ \ -I_2]$, $M(p) = B_1'$, and $K(p) = B_1'\gamma(P_1 \otimes I_2)$, together with eqn. 4.6 with $\tilde{K}_R(p) = I_2$.

The input-output behaviour of the simulated closed-loop system is shown in fig. 4.11. It may be observed that after the change of operating points from Q_1 to Q_2 at sample instant 38, the performance of the system deteriorates to a level which might be unacceptable in many robot applications. This confirms results on the application of linear (nonadaptive) controllers to the PUMA manipulator reported in

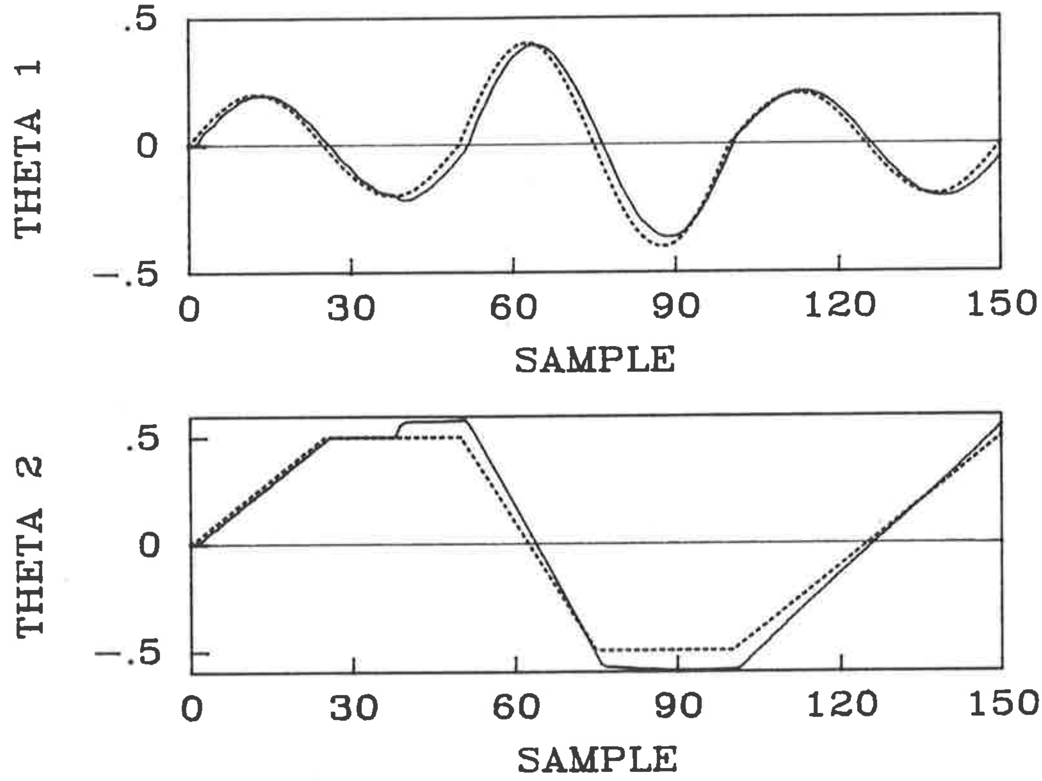

Figure 4.11: Nonadaptive MPE controller based on the RNI matrix applied to the two-link robot manipulator: $\theta_1(t)$, $\theta_2(t)$ and reference sequences (shown with dotted lines).

[144] which indicate the need to update the linearized system model and the corresponding controller to ensure satisfactory performance of the closed-loop system under variety of operating conditions. Note that the self-tuning strategy yields a significantly better performance under the same operating conditions (see fig. 4.8), despite drifts of parameter estimates after transition from Q_1 to Q_2.

4.4 Concluding remarks.

In this chapter, the minimum prediction error (MPE) controller which minimizes the k-step-ahead single-stage cost function is considered for the development of the new *indirect* self tuner.

The self-tuning MPE controller involves the *on-line* calculation of the right nilpotent interactor (RNI) matrix from the estimates of the DARMA model. This increases the computational burden in comparison with strategies based on the assumption of prior knowledge of the interactor matrix. However, the prior system knowledge required by the self tuner is reduced to the *upper bound* on the degree of polynomial matrices of the DARMA model.

For the purpose of comparison, the strategy of [35], which minimizes the same cost function, requires prior knowledge of the unique right *interactor matrix* and involves estimation of *more* parameters. The pole-placement (indirect) self tuner of [39] involves estimation of the same number of parameters but requires prior knowledge of the system *observability indexes* and an upper bound on the system *controllability index*. Furthermore, the above strategies assume that the system is *not* affected by a deterministic disturbance, in contrast to the new MPE self tuner.

The only MPE strategy which requires the same prior system knowledge is that of [39]. The cost function minimized by this strategy penalizes the tracking error filtered by the left interactor matrix. On the other hand, the new MPE self tuner minimizes the actual tracking error and the control signal filtered by the inverse of the RNI matrix. Therefore, one would need to examine which performance criterion is more meaningful for a particular application. Furthermore, the strategy of [39] requires two divisions of polynomial matrices, in contrast to the new self tuner which involves multiplication of polynomial matrices.

In comparison with *direct* self-tuning controllers, the self tuner based on the RNI matrix estimates less parameters; therefore, superior convergence rate can be expected for the latter strategy.

The MPE self tuner has three tuning knobs: Υ, Λ and $S(p)$. Appropriate choice of $\Lambda \neq 0$ may be crucial to guarantee closed-loop system stability for nonminimum phase systems. Such a choice is not straight-forward for systems with unknown parameters. In order to overcome the above-mentioned drawback of the MPE self tuner, we shall consider long-range predictive control for application in self-tuning control.

Chapter 5

Self-tuning long-range predictive control.

In this chapter, the self-tuning long-range predictive control of systems having the feedback configuration FI (see page 18), is considered.

The MPE control laws derived in chapters 4 and 6 minimize the k-step-ahead single-stage performance criteria (see eqn. 4.1 (p. 120), 6.1 (p. 197), 6.38 (p. 217)). Although such controllers are simple, it may be difficult to choose the parameters of the control law which yield a stable closed-loop system for nonminimum phase plants, especially in self-tuning applications (see Lemma 4.2 (p. 122), Lemma 6.1 (iii) on p. 197, and Theorem 6.1 (iii) on p. 218). Furthermore, the MPE self tuners are sensitive to variable system delay [29].

Alternatively, a multi-stage performance criterion can be considered to overcome some of the drawbacks of the single-stage approach. In this case, the optimization horizon extends over a significant part of the plant response to the present control signal (i.e., beyond the k-step-ahead horizon), thus explaining the term "long-range" [149]. The resulting long-range predictive (LRP) controllers appear to possess properties desirable in self-tuning control at the cost of increased computational burden in comparison with the MPE strategies. In particular, the LRP controller developed in this chapter has a number of tuning knobs that give the user a wider choice in the design of the closed-loop system performance than the parameters of the MPE control law. Simulation studies have illustrated desirable robustness properties of the LRP controller against violation of modeling assumptions for a scalar plant [29,150]. Another advantage of the LRP approach is that it is possible to use known future reference sequence values (the so-called programmed control [27]), a feature desirable in robot control [94], for example.

There are many approaches to LRP control depending on the plant representation and performance criterion [30,151,118,152,74,127,27,130,36,38,149]. In this chapter, however, we consider the approach based on the LDO system representation and involving a cost function related to those of the control strategies known as *Dynamic Matrix Control* (DMC) and *Generalized Predictive Control* (GPC).

The DMC strategy was originally developed for multivariable deterministic systems using the step-response model [30]. The DMC was successfully applied to petrochemical process control [30,151]. The comparison of the DMC with other similar long-range predictive control techniques based on the weighting-sequence plant representation, revealed superior properties of the DMC in terms of the resulting closed-loop system performance, and indicated promising properties for self-tuning applications [152,149]. However, the number of the coefficients of the truncated weighting-sequence model required for accurate plant representation may be large, especially for poorly damped systems with high sampling rate [149]. This is undesirable in self-tuning applications due to a large number of parameters which are to be estimated. Furthermore, it is known that only stable plants can be represented by the weighting-sequence models, thus excluding systems which are unstable or possess integral action [149]. In order to overcome drawbacks resulting from the use of the step-response model, the CARIMA model (see eqns. 2.33 and 2.34, p. 32) of a scalar system was chosen for the development of the GPC strategy extending the DMC approach [29,149]. The self-tuning GPC was found to possess superior robustness properties in comparison not only with the LRP control methods based on the weighting-sequence models (including DMC), but also the self-tuning MPE controllers resulting from minimization of a single-stage performance criterion and pole-placement self tuners [29].

In this chapter the DMC approach is considered for the development of a LRP controller for (i) multivariable square (MIMOS) deterministic systems described by the DARMA model (see section 5.1), and (ii) scalar stochastic systems described by the ARMAX model (see section 5.2).

With respect to the original DMC, a contribution of this chapter is in the development of strategies based on the DARMA and ARMAX models. In comparison with the DMC, the resulting LRP controllers possess the following properties (i) the LRP control can be applied to unstable plants, (ii) in general, fewer parameters are required for adequate process representation, hence the resulting control law is more amenable to self-tuning, (iii) the LRP self tuner developed for the ARMAX model possesses similar robustness properties to those of the GPC developed for the CARIMA model. Furthermore, (iv) the incorporation of the generator of deterministic disturbances into the model of a deterministic system (see discussion on p. 19 and p. 28) allows for a unified treatment of elimination of the effect of deterministic disturbances on system outputs. This is in contrast to the LRP techniques based on the weighting-sequence models which involve some *ad hoc* methods for eliminating output offset [149]. Furthermore, the DMC does not guarantee solvability for the control signal for systems with arbitrary delay structure. In the following analysis, the design of the controller involves characterization of the multivariable system delay structure by an interactor matrix which overcomes the latter drawback of the DMC. In particular, the left nilpotent interactor matrix, introduced in subsection 2.3.2, is employed in the design. The LRP controller is also developed for a scalar system subject to stochastic disturbance, which was not considered for the DMC.

With respect to the GPC, a contribution of this chapter involves the following aspects. Firstly, the LRP controller is developed for multivariable deterministic systems, in contrast to scalar systems considered for the GPC. Secondly, the LRP control law is derived for scalar systems subject to stationary stochastic disturbances, in contrast to nonstationary disturbances assumed for the GPC. It has been found in simulation studies that if the system is subject to stationary stochastic disturbance modeled by the ARMAX model, then the closed-loop system performance deteriorates significantly if the GPC strategy developed for the CARIMA model (i.e., for a system with nonstationary random disturbance) is applied [49] (see also example 5.2.1, p. 185). As pointed out in [94], in servomechanism self-tuning, such as robot control, the predominant stochastic disturbance is sensor noise. Such a disturbance is stationary and the GPC results in unacceptable system performance [94]. Furthermore, the model of a closed-loop system involving the LRP controller is developed for scalar stochastic systems. The model facilitates (i) a rigorous stability analysis of the closed-loop system, (ii) investigation of the influence of the parameters of the LRP controller on the location of the closed-loop system poles, (iii) calculation of the variance of the output regulated with the LRP controller. To the author's best knowledge such results have not been presented either for the DMC or for the GPC strategies.

5.1 Self-tuning long-range predictive control based on the left interactor matrix for a multi-input, multi-output deterministic system.

In this section:

- the long-range predictive control law, which minimizes multi-stage cost function for a MIMOS system, is derived (see approach (ii) on page 60);
- the self-tuning strategy, based on the long-range predictive control law, is introduced;
- the application of this self tuner to the control of a two-link robot manipulator is described.

The results of subsections 2.3.2 and 3.1.1 are used in this section.

5.1.1 Long-range predictive control of a deterministic system with known parameters.

Let us consider a MIMOS deterministic system satisfying Assumptions 3.1, 3.2 and 3.3 (see pp. 58-59). Consider the following quadratic multi-stage cost function [1]

$$\begin{aligned} J_{P,C} &= \sum_{j=0}^{P} \|\overline{K}_L(q)[y(t+j) - r(t+j)]\|_{\Upsilon}^2 + \sum_{j=0}^{C} \|S(p)u(t+j)\|_{\Lambda}^2 + \\ &+ \sum_{j=C+1}^{P} \|\Delta(p)u(t+j)\|_{\Lambda_\Delta}^2, \end{aligned} \quad (5.1)$$

where Υ and Λ_Δ are positive definite $m \times m$ matrices, and Λ is a positive semidefinite $m \times m$ matrix; the $m \times m$ weighting polynomial matrices $S(p)$ and $\Delta(p)$ are defined as

$$\begin{aligned} S(p) &= S_0 + S_1 p + \cdots + S_{n_s} p^{n_s}, \qquad \det S_0 \neq 0 \quad \text{and} \quad n_s \geq 0, \\ \Delta(p) &= I_m - I_m p. \end{aligned}$$

The first term in cost function 5.1 penalizes the squared norm of the error between future filtered system outputs $\overline{K}_L(q)y(t+j)$ and future filtered reference signals $\overline{K}_L(q)r(t+j)$ over the *prediction horizon* P. The left interactor matrix $\overline{K}_L(q)$ is either the unique interactor matrix $\xi_L(q)$, or the left nilpotent interactor (LNI) matrix $K_L(q)$ introduced in subsection 2.3.2.

The second term penalizes the control effort over the *control horizon* C, thus allowing for a compromise between minimization of the tracking error (first term in criterion 5.1) and cost of the required control effort.

The third term is introduced to ensure boundedness of the control sequence for nonminimum phase systems when $\Lambda = 0$. The control horizon C defines the sample instant $t + C$ after which increments of the control signal are penalized in the performance criterion. Such an approach to guarantee boundedness of the control signal

[1] The original DMC was developed as a least squares control problem without postulating any specific cost function [30]. The corresponding cost functions were introduced for the purpose of comparison of the DMC with other LRP control techniques for scalar systems in [152,149]. The cost function postulated here is a generalization of the latter cost functions for multivariable systems.

was originally proposed in [153,27] to control scalar nonminimum phase plants, in contrast to the minimum variance control without weighting the control effort and based on the single-stage performance criterion [83].

The cost function 5.1 involves future values of the reference sequence $r(t+k+j)$ for $j = 0, \ldots, P$ (k is the degree of the interactor matrix). If the future values of $\{r(t)\}$ are known, then the optimal control law results form minimization of the cost function. This approach is known as ***programmed control*** [27] (see examples in subsection 5.1.2). If the future values of $\{r(t)\}$ are unknown, then their predictions are used [27]. The simplest approach is to set future values equal to the current value $r(t)$. This method is referred to as the *fixed reference sequence model.*

The cost function 5.1 can be minimized with respect to the sequence of present and future control signals U, defined as

$$U = [u(t)' \;\; u(t+1)' \;\; \ldots \;\; u(t+P)']', \tag{5.2}$$

using for example the Lagrange multipliers method [154]. The resulting control law involves inversion of a $(P+1)m \times (P+1)m$ matrix in order to determine U (see [30] for a similar result obtained for the DMC). This may be unacceptable for the self-tuning implementation of the LRP controller, which involves the on-line inversion of such a matrix. However, the computational burden can be significantly reduced if it is assumed that the infinite cost is placed on the changes in the control signal beyond the control horizon C, i.e., $\Lambda_\Delta \to \infty$. This implies that there will be no change in the control sequence after time $t+C$. Therefore, the cost function 5.1 can be reduced to the following performance criterion

$$J_{P,C} = \sum_{j=0}^{P} \|\overline{K}_L(q)[y(t+j) - r(t+j)]\|_\Upsilon^2 + \sum_{j=0}^{C} \|S(p)u(t+j)\|_\Lambda^2, \tag{5.3}$$

with the constraint on changes in the control sequence after time $t+C$, i.e.,

$$\Delta(p)u(t+C+i) = 0 \quad \text{for} \quad i = 1, \ldots, P-C.$$

The minimization of cost 5.3 yields the control sequence with no variations beyond the control horizon. Hence, it is sufficient to calculate the truncated control sequence vector U_T consisting of the first $C+1$ controls of U

$$U_T = [u(t)' \ldots u(t+C)']'. \tag{5.4}$$

This is an important idea adopted here from the DMC approach which results in a simplification of the LRP control law and reduction of the computational burden.

Before considering the minimization of the cost function 5.3, let us define the following vectors

$$\begin{aligned} U_P &= \left[\, u(t-1)' \ldots u(t-n+1)' \,\right]', & (5.5)\\ U_S &= \left[\, u(t-1)' \ldots u(t-n_s)' \,\right]', & (5.6)\\ Y_P &= \left[\, y(t)' \ldots y(t-n+1)' \,\right]', & (5.7)\\ R &= \left[\, (\overline{K}_L(q)r(t))' \ldots (\overline{K}_L(q)r(t+P))' \,\right]', & (5.8)\end{aligned}$$

where U_P and U_S are $(n-1)m \times 1$ and $n_s m \times 1$ vectors of past control signals (n is the degree of the polynomial matrices in the DARMA model 3.1, p. 59), Y_P is $nm \times 1$ vector of present and past outputs and R is $(P+1)m \times 1$ vector of future reference signals filtered by the left interactor matrix $\overline{K}_L(q)$.

Note that the cost function 5.3 involves future values of the system outputs filtered by the interactor matrix, i.e., $y_k(t+j) = \overline{K}_L(q)y(t+j)$. In the derivation of the LRP control law we shall employ the multi-step-ahead predictor expression for $y_k(t+j)$ introduced in Lemma 3.1 (eqn. 3.6, p. 61). Let us define the $(P+1)m \times (P+1)m$ and $(P+1)m \times (n-1)m$ matrices consisting of coefficient matrices of the polynomial matrix $\beta^{(j)}(p)$ of the multi-step-ahead predictor for $j = 0, \ldots, P$ (see eqn. 3.11, p. 62)

$$B_F = \begin{bmatrix} \beta_0^{(0)} & 0 & 0 & \ldots & 0 \\ \beta_0^{(1)} & \beta_{-1}^{(1)} & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \beta_0^{(P-1)} & \beta_{-1}^{(P-1)} & \ldots & \beta_{-P+1}^{(P-1)} & 0 \\ \beta_0^{(P)} & \beta_{-1}^{(P)} & \ldots & \beta_{-P+1}^{(P)} & \beta_{-P}^{(P)} \end{bmatrix} = \begin{bmatrix} \beta_0^{(0)} & 0 & 0 & \ldots & 0 \\ \beta_0^{(1)} & \beta_0^{(0)} & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \beta_0^{(P-1)} & \beta_0^{(P-2)} & \ldots & \beta_0^{(0)} & 0 \\ \beta_0^{(P)} & \beta_0^{(P-1)} & \ldots & \beta_0^{(1)} & \beta_0^{(0)} \end{bmatrix}, \tag{5.9}$$

$$B_P = \begin{bmatrix} \beta_1^{(0)} & \beta_2^{(0)} & \dots & \beta_{n-1}^{(0)} \\ \beta_1^{(1)} & \beta_2^{(1)} & \dots & \beta_{n-1}^{(1)} \\ \vdots & \vdots & \dots & \vdots \\ \beta_1^{(P)} & \beta_2^{(P)} & \dots & \beta_{n-1}^{(P)} \end{bmatrix}. \tag{5.10}$$

The block Toeplitz structure of matrix B_F results from application of eqn. 3.60, p. 76. Next define the $(P+1)m \times nm$ matrix consisting of coefficient matrices of the polynomial matrix $\alpha^{(j)}(p)$ of the multi-step-ahead predictor for $j = 0, \dots, P$ (see eqn. 3.10, p. 62)

$$A_P = \begin{bmatrix} \alpha_0^{(0)} & \alpha_1^{(0)} & \dots & \alpha_{n-1}^{(0)} \\ \alpha_0^{(1)} & \alpha_1^{(1)} & \dots & \alpha_{n-1}^{(1)} \\ \vdots & \vdots & & \vdots \\ \alpha_0^{(P)} & \alpha_1^{(P)} & \dots & \alpha_{n-1}^{(P)} \end{bmatrix}. \tag{5.11}$$

Next we introduce the $(C+1)m \times (C+1)m$ and $(C+1)m \times n_s m$ matrices of coefficients of the filter $S(p)$ (put $S_i = 0$ if $i > n_s$)

$$S_F = \begin{bmatrix} S_0 & 0 & 0 & \dots & 0 \\ S_1 & S_0 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ S_{C-1} & S_{C-2} & \dots & S_0 & 0 \\ S_C & S_{C-1} & \dots & S_1 & S_0 \end{bmatrix}, \tag{5.12}$$

$$S_P = \begin{bmatrix} S_1 & S_2 & S_3 & \dots & S_{n_s-1} & S_{n_s} \\ S_2 & S_3 & S_4 & \dots & S_{n_s} & 0 \\ \vdots & & & & & \vdots \\ S_{C+1} & \dots & S_{n_s} & 0 & \dots & 0 \end{bmatrix}. \tag{5.13}$$

Finally, let us define the following $(P+1)m \times (C+1)m$ matrix

$$T = \begin{bmatrix} & I_{C+1} & \\ 0 & \dots & 0 & 1 \\ \vdots & & \vdots & \vdots \\ 0 & \dots & 0 & 1 \end{bmatrix} \otimes I_m, \tag{5.14}$$

(I_i is the $i \times i$ identity matrix).

Now the LRP control law is given in the following result.

Lemma 5.1 *Consider a MIMOS system satisfying Assumptions 3.1, 3.2 and 3.3 (see pp. 58-59). For such a system the control signal sequence $u(t+i)$, $i = 0, \ldots, C$, which minimizes the cost function 5.3 given the constraints $\Delta(p)u(t+C+i) = 0$ for $i = 1, \ldots, P-C$, is given by*

$$U_T = B_{rT}^{-1} \left[\begin{array}{cccc} B'_{FT}\Upsilon_P & -B'_{FT}\Upsilon_P A_P & -B'_{FT}\Upsilon_P B_P & -S'_F\Lambda_C S_P \end{array} \right] \left[\begin{array}{c} R \\ Y_P \\ U_P \\ U_S \end{array} \right], \tag{5.15}$$

where the $(P+1)m \times (C+1)m$ matrix $B_{FT} = B_F T$, and the $(C+1)m \times (C+1)m$ matrix B_{rT} is defined as

$$B_{rT} = B'_{FT}\Upsilon_P B_{FT} + S'_F\Lambda_C S_F, \tag{5.16}$$

and $\Upsilon_P = I_{P+1} \otimes \Upsilon$ and $\Lambda_C = I_{C+1} \otimes \Lambda$.

Proof.

Let us define the (filtered) tracking error as $e(t+j) = \overline{K}_L(q)\,[y(t+j) - r(t+j)]$, and the $(P+1)m \times 1$ vector tracking error as $E = [e(t)' \;\ldots\; e(t+P)']'$. The vector (filtered) tracking error can be expressed in the matrix form using eqns. 3.6 (p. 61), 5.2, and 5.5 to 5.11 as follows

$$E = A_P Y_P + B_F U + B_P U_P - R.$$

Note from eqns. 5.2, 5.4, 5.9 and 5.14 that, using the constraint $\Delta(p)u(t+C+i) = 0$ for $i = 1, \ldots, P-C$, one can write $B_F U = B_{FT} U_T$, where $B_{FT} = B_F T$. [2] Therefore, the vector tracking error can be written as

$$E = A_P Y_P + B_{FT} U_T + B_P U_P - R. \tag{5.17}$$

[2] Note that postmultiplication by matrix T leaves the first C block columns of B_{FT} unchanged from B_F and the $(C+1)$-th block column of B_{FT} is a sum of the last $(P-C+1)$ block columns of B_F.

Next, using eqns. 5.12 and 5.13 one has $(I_{C+1} \otimes S(p))U_T = S_F U_T + S_P U_S$. Now the cost function 5.3 with the constraint on changes in future values of the control sequence, which is embedded in eqn. 5.17, can be written as

$$\begin{aligned} J_{P,C} &= E'(I_{P+1} \otimes \Upsilon)E + (S_F U_T + S_P U_S)'(I_{C+1} \otimes \Lambda)(S_F U_T + S_P U_S) \\ &= (A_P Y_P + B_{FT} U_T + B_P U_P - R)'\Upsilon_P(A_P Y_P + B_{FT} U_T + B_P U_P - R) + \\ &\quad +(S_F U_T + S_P U_S)'\Lambda_C(S_F U_T + S_P U_S). \end{aligned}$$

Differentiating the above cost with respect to U_T and setting the result to zero gives the control law 5.15.

Q.E.D.

It follows from Lemma 3.1 (p. 61) that the coefficient matrix $\beta_0^{(0)}$ is nonsingular. It is also assumed that the coefficient matrix S_0 of the filter $S(p)$ is nonsingular. Hence, matrices B_{FT} and S_F have full rank. Thus, matrix B_{rT} is invertible (Υ and Λ are positive definite and positive semidefinite matrices, respectively). Therefore, nonsingularity of the matrix B_{rT} is guaranteed through the use of the left interactor matrix in the multi-step-ahead predictor (see Lemma 3.1, p. 61) and in the choice of the performance criterion 5.3 (see approach (ii) on p. 60).

On the other hand, the original DMC strategy involves inversion of a matrix consisting of the coefficient matrices of the step-response model. This matrix is not, in general, guaranteed to be nonsingular. This is a consequence of deriving the multivariable DMC scheme as an extension to its version for scalar systems [30]. (For a scalar system, nonsingularity of the corresponding matrix can be guaranteed by the choice of controller parameters $P - C + 1 > k$, where k is system delay [149].)

We shall now comment on the computations involved in evaluation of the control signal from the control law 5.15. Firstly, the dimension of the matrix which is to be inverted in calculation of the control sequence, was reduced from $(P+1)m$ to $(C+1)m$ by putting $\Lambda_\Delta \to \infty$. For example, if $C = 0$ then the inverted matrix is of dimension $m \times m$. Secondly, evaluation of matrices B_F, B_P and A_P requires calculation of the coefficients of the multi-step-ahead predictors for $j = 0, \ldots, P$.

The algorithm given in Remark 3.1 (p. 65) can be used for this purpose for each j. However, if $P > 0$ (which is usually the case) then the recursive algorithm given in Remark 3.2 (p. 69) yields the matrices of coefficients of the j-step-ahead predictor in terms of matrices of coefficients of the $(j-1)$-step-ahead predictor. In other words, the j-th block row of matrices B_F, B_P and A_P is calculated recursively in terms of the $(j-1)$-th block row of the corresponding matrix (for $j \geq 1$).

The $C+1$ control m-vectors, U_T, are evaluated from the LRP control law 5.15. These controls are applied at sample instants $t, t+1, \ldots, t+P$. This implies that the feedback control law (i.e., the GLCL 2.35 (p. 33) representing the LRP controller) is time-varying. The LRP controllers introduced in [30,155,38,36,149] are other examples of the control laws which can be represented by the time-varying GLCL. However, it was suggested in [155,38,149,92] that for the long-range predictive controllers *receding horizon* strategy can be used.

For the receding horizon strategy, only the first m-vector $u(t)$ of the control sequence U_T is calculated and applied to the controlled system at every sample instant t. At the next sample instant $t+1$ the calculation of the first control m-vector of the new control sequence U_T is repeated.

The receding horizon strategy implies that the most recent input-output data is used in evaluation of control sequence U_T at every sample instant t; otherwise, the control sequence U_T, evaluated at sample instant t, is not affected by system outputs at time $t+1$, $t+2$, $\ldots$, $t+P$.

The receding horizon strategy reduces the computational burden, since only the first (block) row of matrix B_{rT}^{-1} is involved in calculation of $u(t)$ from eqn. 5.15. For the remainder of this chapter, the LRP control law 5.15 will be used in the receding horizon sense.

The LRP controller resulting from the minimization of cost function 5.3 has the following tuning knobs: prediction and control horizons, P and C respectively, weighting polynomial matrix $S(p)$ and weighting matrices Υ and Λ.

It was found in the simulation studies (some of which are described in subsection 5.1.2) that the prediction horizon P should be set to such a value that $(k+P)T_s$, where k is the degree of the interactor matrix and T_s is the sampling period, covers a significant part of the plant step response (e.g., plant rise-time) [30,152]. Then the choice of C comparable with P (but smaller) gives rise to a dead-beat type of response with quick settling of the controlled outputs. For small values of C, a more sluggish output response results (see examples in subsection 5.1.2). In many cases the choice of $C = 0$ leads to satisfactory closed-loop system performance. However, the simulation studies performed for scalar systems [150] indicate that the choice of the time-oriented parameters C and P may be more difficult for dynamically-complex plants (see discussion on p. 184).

If future values of the reference sequence are known (programmed control), the choice of P shapes the controller ability to react to the future change of the reference sequence in advance (see example 5.1.1, p. 163). The controller anticipatory action associated with programmed control can be helpful in improving the input-output performance for sluggish systems, including systems with large delay [94].

The weighting matrix Λ is a performance-oriented tuning knob which makes it possible to damp the movements of the actuator. This is the main purpose of its use, in contrast to controllers which minimize a single-stage cost function. In the latter case, the control weighting matrix Λ plays a crucial role in ensuring boundedness of the control signal for nonminimum phase plants (see Lemma 4.2, p. 122). For the LRP controller, however, boundedness of controls for nonminimum phase systems is achieved by setting $C < P$, for $\Lambda \geq 0$ (see examples in subsection 5.1.2). (Such a choice of C and P prevents the unbounded growth of the amplitude of the control signal by putting an infinite cost on changes in the control sequence after time $t+C$.) If $\Lambda \neq 0$, then the weighting polynomial matrix $S(p)$ can be chosen such that $S(1) = 0$ (e.g., $S(p) = I_m - pI_m$), so that there is zero steady-state error for a constant reference sequence.

Now two special cases of the LRP control law are discussed.

Weighted one-step-ahead controller. If $C = P = 0$ and $\Lambda > 0$, then the LRP control law reduces to the weighted one-step-ahead controller introduced for

$\overline{K}_L(q) = \xi_L(q)$ in [9, chapter 5]. It follows from eqns. 5.15, 3.7, 3.8 and 3.1 that the resulting GLCL, given by eqn. 2.35 (p. 33), involves

$$\begin{aligned} V(p) &= (\beta_0^{(0)})'\Upsilon\beta^{(0)}(p) + (S_0)'\Lambda S(p), && (5.18)\\ M(q) &= (\beta_0^{(0)})'\Upsilon\overline{K}_L(q), && (5.19)\\ K(p) &= (\beta_0^{(0)})'\Upsilon\alpha^{(0)}(p). && (5.20) \end{aligned}$$

The outputs of the closed-loop system satisfy

$$(\beta_0^{(0)})'\Upsilon\overline{K}_L(q)y(t) = (\beta_0^{(0)})'\Upsilon\overline{K}_L(q)r(t) - (S_0)'\Lambda S(p)u(t). \qquad (5.21)$$

One-step-ahead controller. If $C = P(\geq 0)$ and $\Lambda = 0$, then the LRP control law reduces to the one-step-ahead controller introduced for $\overline{K}_L(q) = \xi_L(q)$ in [9, Theorem 5.2.5, p.138]. Note that in this case $U_T = U$, $B_{FT} = B_F$, and $B_{rT} = B_F'\Upsilon_P B_F$. The LRP control law 5.15 is

$$U = (B_F)^{-1}\Upsilon_P\left\{R - A_P Y_P - B_P U_P\right\}.$$

Because of the lower triangular block Toeplitz structure of B_F, the control signal $u(t)$ is given by the GLCL defined by eqns. 5.18 to 5.20 for $\Lambda = 0$, i.e., the LRP control law reduces to the one-step-ahead controller.

Finally we shall comment on the significance of the properties of the left interactor matrix in the design of the LRP controller.

Comment 5.1 *The properties which characterize an interactor matrix (see Remark 2.1, p. 42) were embedded in the definition of the left nilpotent interactor (LNI) matrix (see Definition 2.3, p. 43).*

The first property, given by eqn. 2.42 (p. 43), ensures that matrix B_{rT} in the LRP control law eqn. 5.15 is nonsingular. Hence, the control signal $u(t)$ (as well as the control sequence U_T) can be determined uniquely as a function of past and present values of the controlled outputs and past controls.

The importance of the second property, given by eqn. 2.43 (p. 43), is revealed by eqn. 5.21: stability of $\overline{K}_L(z)$ is required to ensure asymptotic reference tracking.

The interactor matrix $\overline{K}_L(q)$ affects the closed-loop system output response (see eqn. 5.21 for the choice of the parameters $C = P = 0$). Nonuniqueness of the LNI matrix $K_L(q)$ introduces a degree of freedom in the design of the compensator (see example 5.1.1, figs. 5.1 and 5.2).

Aside 5.1 *It is the opinion of the author that the use of the LNI matrix in the design of (weighted) minimum variance controllers (MVC) for systems affected by stochastic disturbances will provide an additional degree of freedom in terms of influencing variances of the individual outputs of the closed-loop system.*

The problem of designing a MVC for a system described by ARMAX model 2.26 with the delay structure characterized by the unique interactor matrix $\xi_L(z)$ was considered in [36]. The resulting MVC does not minimize the individual output variances; the unconditional minimum variance control is achieved only for the first element of the output vector $y(t)$, whereas variances of the remaining outputs are minimized subject to constraints (this is due to the triangular structure of $\xi_L(z)$). The use of the LNI matrix in a similar design should lead to a MVC which minimizes variances of all outputs subject to constraints, i.e., the unconditional minimum variance control will not be achieved for any system output. Nonuniqueness of the LNI matrix would, however, introduce the possibility of achieving more uniform minimization of variances of all outputs.

5.1.2 Self-tuning long-range predictive control.

The long-range predictive control law 5.15 can be combined with a recursive parameter estimator to yield the indirect self-tuning controller involving estimation of the parameters of the system DARMA model. Such a self tuner will be referred to as the ST-LRP controller.

It is assumed that estimates of the parameters of the system DARMA model are evaluated at every sample instant t. Since the LRP control law is used in the receding horizon sense, the most recent estimates are used in evaluation of the control signal at every sample instant. However, in the self-tuning case, the cost function 5.3 is minimized under the simplifying assumption that process parameter estimates remain constant within the prediction horizon P.

The LRP control law 5.15 involves the interactor matrix $\overline{K}_L(z)$. Therefore,

Assumption 5.1 *The left interactor matrix $\overline{K}_L(z)$ is known.*

This assumption means that the approach 1 in the classification on page 38 is involved, i.e., "complete knowledge of the left interactor matrix is assumed". However, we do not restrict the characterization of the delay structure to the unique left interactor matrix $\xi_L(z)$ (approach 1 (c), p. 38); the LNI matrix $K_L(z)$, being a nonunique characterization of the delay, is considered as well (approach 1 (d), p. 38). On the other hand, Assumption 5.1 can be removed if an algorithm for the on-line calculation of the interactor matrix is employed in the self tuner, as was done in the ST-MPR-RNI strategy (see section 4.2). However, the algorithms for the calculation of the LNI matrix cannot be employed since they operate on the RDO representation (see fig. 2.8, p. 55), rather than the DARMA model which is estimated in the ST-LRP strategy. Alternatively, the algorithm for evaluation of $\xi_L(z)$ from the DARMA model via polynomial matrix division, proposed in [39], could be employed. This corresponds to the approach 5 (a) in the classification on page 39.

Furthermore, the required prior system knowledge involves an upper bound on the degree of the polynomial matrices of the DARMA model $\widehat{n}$ (see Assumption 4.2, p. 126).

The **ST-LRP controller** involves the following steps which are performed at every sample instant t:

step 1. calculate estimates $\widehat{A}_i$ and $\widehat{B}_i$ of the coefficient matrices A_i and B_i of the DARMA model using, say, the simplified multivariable RLS algorithm given by eqns. E.18 to E.22, Appendix E ($i = 1, \ldots, \widehat{n}$);

step 2. using estimates from step 1 in place of true DARMA model coefficient matrices, calculate matrices of coefficients $\widehat{F}^{(0)}$, $\widehat{\alpha}^{(0)}$, and $\widehat{\beta}^{(0)}$ [3] using the algorithm

[3] The symbol "$\widehat{\ }$" is used to differentiate results of calculations based on the estimates of parameters of the DARMA model from results which are based on true parameters.

given in Remark 3.1, p. 65;

step 3. calculate matrices of coefficients $\widehat{\alpha}^{(j)}$ and $\widehat{\beta}^{(j)}$ using the algorithm given in Remark 3.2, p. 68 ($j = 1, \ldots, P$);

step 4. form matrices $\widehat{B}_{FT}$, $\widehat{B}_P$, $\widehat{A}_P$, S_F and S_P, and calculate control signal $u(t)$ from eqn. 5.15 (i.e., the first m elements of the truncated control sequence U_T).

The macro XSLR, which is written in MATLAB commands and implements the ST-LRP strategy, is given in Appendix D. The performance of the ST-LRP controller is illustrated in the following examples.

Example 5.1.1.

In this example a nonminimum phase system is considered. It will be demonstrated in a simulation study that the choice $C < P$ ensures closed-loop system stability even if $\Lambda = 0$. On the other hand, the closed-loop system stability condition for such a system controlled by the MPE strategy requires a nonzero value of Λ, the choice of which is related to the (unknown) system parameters (see Lemma 4.2, p. 122). The influence of the values of C and P on the closed-loop system performance for both the programmed control and the fixed reference sequence model is shown as well. Furthermore, two different LNI matrices are considered for the design of the ST-LRP controller.

Consider a two-input, two-output ($m = 2$) system given by the following transfer matrix:

$$H_{y,u}(z) = \begin{bmatrix} \frac{1}{z-0.4} & \frac{z+0.2}{(z+0.6)(z-0.4)} \\ \frac{(z-0.4)}{(z+0.5)(z-0.7)} & \frac{1}{z+0.5} \end{bmatrix}.$$

The corresponding DARMA model 3.1 (p. 59) is given by

$$\begin{aligned} A(p) &= [I_2\ A_1\ A_2]\,(P_2 \otimes I_2) = \begin{bmatrix} 1 & 0 & 0.2 & 0 & -0.24 & 0 \\ 0 & 1 & 0 & -0.2 & 0 & -0.35 \end{bmatrix} (P_2 \otimes I_2), \\ B(p) &= p\,[B_1\ B_2]\,(P_1 \otimes I_2) = p \begin{bmatrix} 1 & 1 & 0.6 & 0.2 \\ 1 & 1 & -0.4 & -0.7 \end{bmatrix} (P_1 \otimes I_2), \end{aligned}$$

i.e., $n = 2$. Note that equation $\det B(z^{-1}) = 0$ has root at $z = 3.4$, i.e., this system is nonminimum phase.

The LNI matrix, calculated for this system using the algorithm developed in subsection 2.3.2 with QR matrix decomposition, and denoted LNI-QR, is given by

$$K_L(z) = \begin{bmatrix} -0.7071z & -0.7071z \\ -0.7071z^2 & 0.7071z^2 \end{bmatrix},$$

and $\det K_L(z) = -z^3$. The LNI matrix calculated for this system with LDU decomposition (Gaussian elimination), and denoted LNI-LDU, is the same as the unique left interactor matrix $\xi_L(z)$

$$K_L(z) = \xi_L(z) = \begin{bmatrix} z & 0 \\ -z^2 & z^2 \end{bmatrix},$$

and $\det K_L(z) = \det \xi_L(z) = z^3$. The degree of the LNI matrices is $k = 2$.

For the simulation studies the reference sequence was chosen as square wave with amplitude 1 and period of 50 sample intervals. The simplified multivariable RLS algorithm given in Appendix E was initialized with the covariance matrix $P(-1) = 10^4 I_8$ and the initial estimates of the coefficient matrices of the DARMA model $\widehat{A}_1(0) = \widehat{B}_1(0) = I_2$ and $\widehat{A}_2(0) = \widehat{B}_2(0) = 0$; forgetting factor $\alpha(t) = 0.95$. In all simulations described below the DARMA model parameter estimates converged to their true values after approximately 20 samples.

Programmed control.

At first it was assumed that the future $k + P$ values of the reference sequence are known at every sample instant t so that vector R can be evaluated according to eqn. 5.8 (programmed control).

For $C = P = 0$ the LRP controller reverts to the (weighted) one-step-ahead controller; it would be thus necessary to choose a nonzero Λ to ensure stability of the closed-loop system with nonminimum phase system [9, Theorem 5.2.5, p. 138]. Instead, $C < P$ and $\Lambda = 0$ is assumed: $C = 0$ and $P = 7, 5, 3, 1$ changing every 50 samples. Furthermore, $S(p) = I_2$ and $\Upsilon = I_2$. (The open-loop system rise-time, defined as time needed for system outputs to reach 90% of their final values after the step inputs, is 3 sample intervals.)

The input-output behaviour of the closed-loop system with the ST-LRP controller based on the LNI-QR and LNI-LDU matrices is shown in figs. 5.1 and 5.2, respectively. Note that the decrease in the value of P (i.e., decrease in the difference $P - C$) results in a more active control signal. This can be explained by the fact that the LRP controller approaches the one-step-ahead controller (since P becomes closer to C) which is unstable for a nonminimum phase plant. A bigger difference $P - C$ leads to a more sluggish input and slower output responses, which is useful during the learning phase of self tuner since the controller is more "cautious". For all values of P, however, the performance of the self-tuner was satisfactory.

It was assumed that at time t future values of the reference sequence up to sample instant $t + P + k$ are known to determine R (programmed control). Hence the LRP controller "can see" future change of the reference sequence and compensate for it in advance. For bigger values of P the compensation process starts earlier but the output response is slower. For smaller P, the compensation starts closer to the reference change, the control signal is more active and results in increased output overshoot. The overshoot decays slower for smaller P. Thus the overall time required for outputs to reach new steady state is largely independent of the value of P.

Note that the structure of the LNI matrix influences the closed-loop system performance. It can be observed that the overshoots in output responses are smaller and die away faster for the LNI-LDU matrix than for the LNI-QR matrix. This illustrates a degree of freedom in the design of the control system performance introduced by nonuniqueness of the interactor matrix.

The next simulation study with programmed control illustrates the influence of the value of control horizon C on the performance of the closed-loop system. The ST-LRP controller based on the LNI-LDU matrix was simulated with $C = 1$, $P = 7, 5, 3$ (changing every 50 samples) and $\Lambda = 0$. The input-output behaviour of the closed-loop system is shown in fig. 5.3. The increased value of C permits more variations in the control sequence since C can be interpreted as the control signal settling time

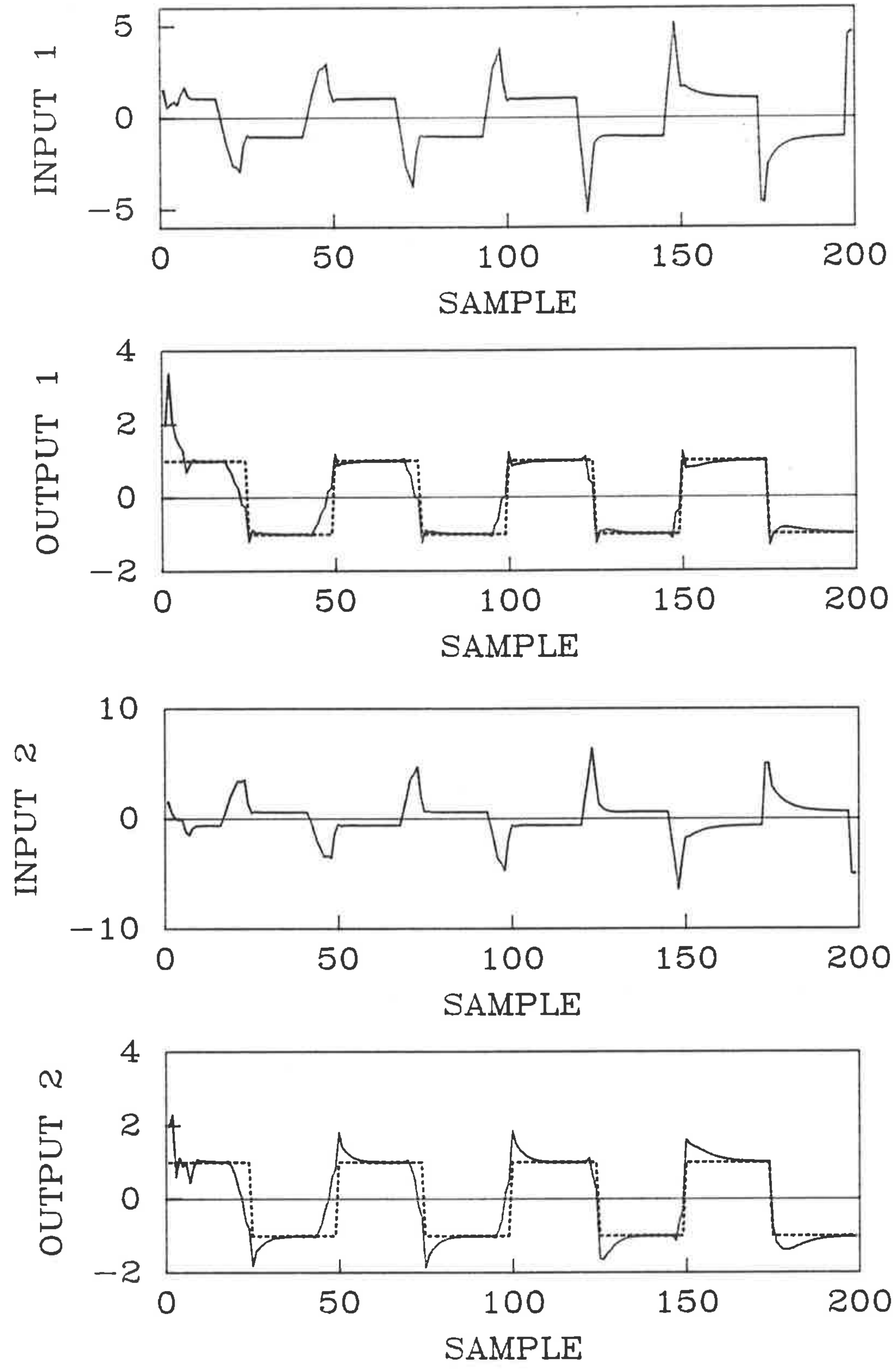

Figure 5.1: ST-LRP controller based on the LNI-QR matrix with $C = 0$, $P = 7, 5, 3, 1$ (changing every 50 samples), $\Lambda = 0$: input, output and reference sequences (programmed control).

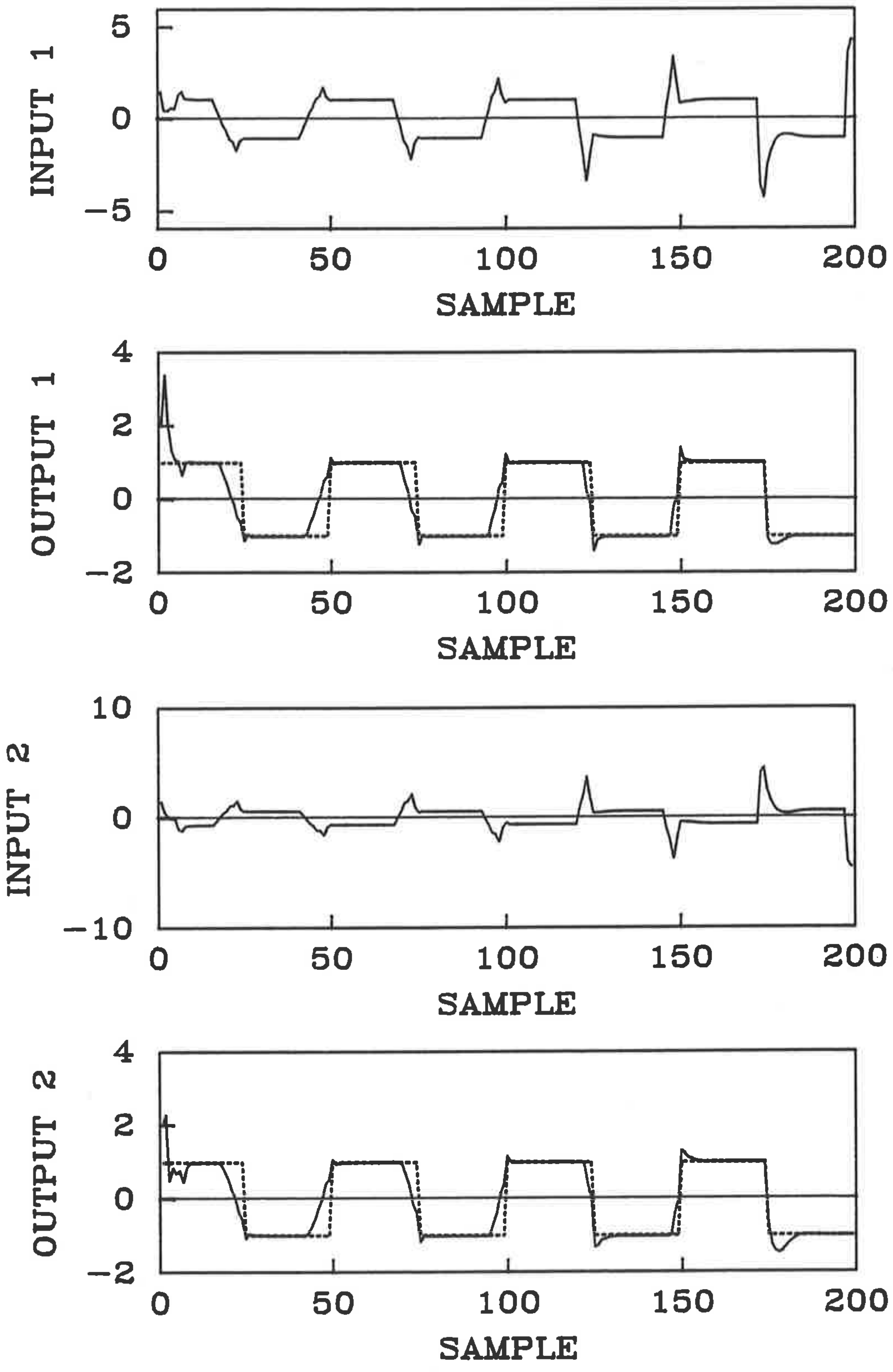

Figure 5.2: ST-LRP controller based on the LNI-LDU matrix with $C = 0$, $P = 7, 5, 3, 1$ (changing every 50 samples), $\Lambda = 0$: input, output and reference sequences (programmed control).

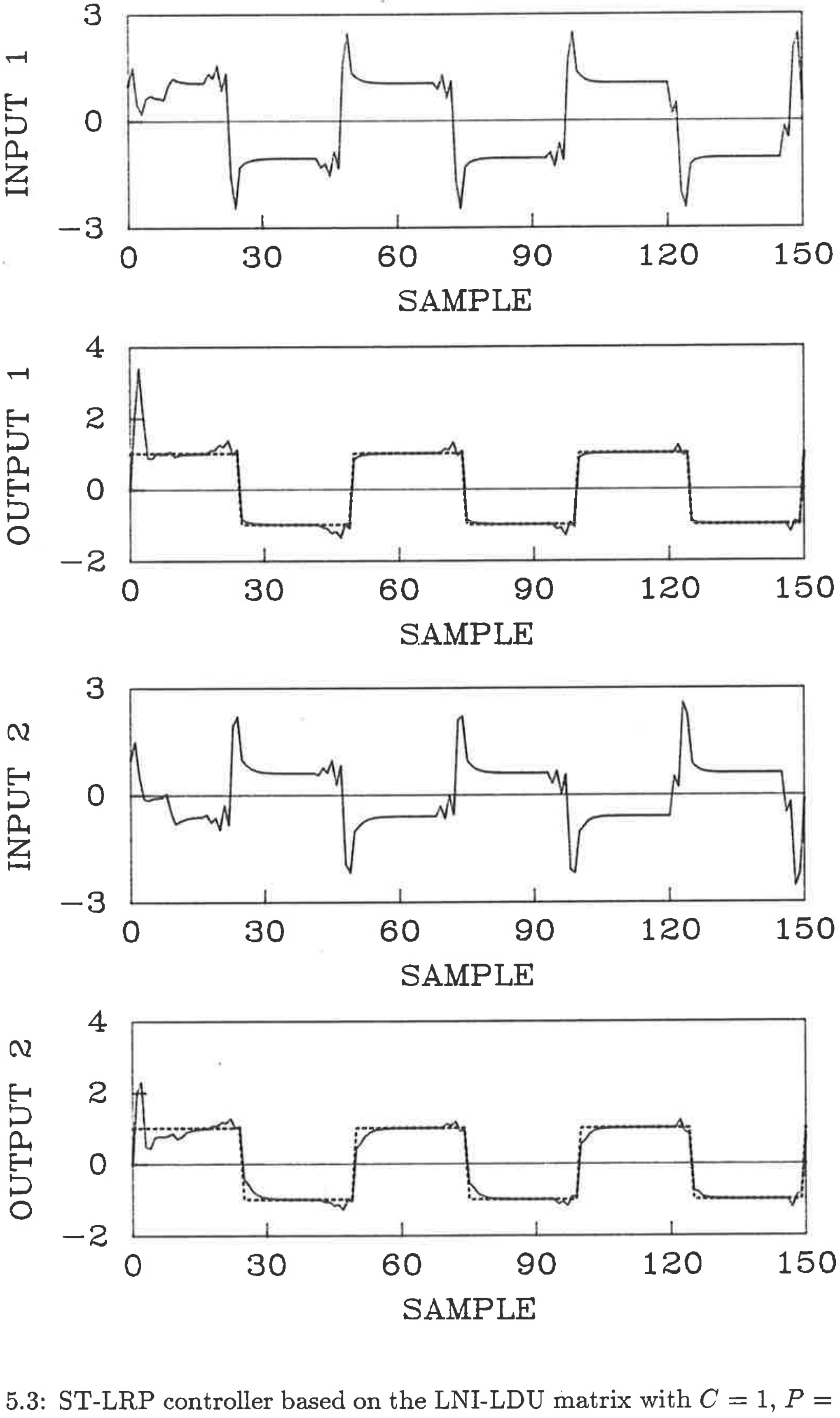

Figure 5.3: ST-LRP controller based on the LNI-LDU matrix with $C = 1$, $P = 7, 5, 3$ (changing every 50 samples), $\Lambda = 0$: input, output and reference sequences (programmed control).

(the control horizon C permits $C+1$ changes in the control sequence U). This can be observed before and after the reference step, where two major changes in the control signal occur. The influence of P on the amplitude of control signal after the reference change is negligible in comparison with the corresponding influence for $C = 0$. The variations in control sequence, preceding the change in the reference sequence, result in "ringing" of the outputs. This undesirable effect becomes less significant for smaller P. The output responses are faster in comparison with those associated with $C = 0$.

Fixed reference sequence model control.

The next simulation study illustrates the effect of the fixed reference sequence model on the closed-loop system performance [4] (see p. 153). For a nondiagonal interactor matrix, the use of the fixed reference sequence model leads not only to the delay in tracking but also to a degradation of the input-output transient performance. These effects are seen by comparison of figs. 5.4 and 5.2 for the ST-LRP controller based on the fixed reference sequence model and programmed control, respectively. The undesirable transient performance, due to coupling introduced by the off-diagonal elements in the interactor matrix, can be influenced by the choice of the controller parameters and the choice of the interactor matrix. It can be seen in fig. 5.4 that, the greater the difference $P - C$, the smoother are outputs. In fact, outputs reach new steady-state values faster for $P = 7$ than for $P = 1$.

It should be pointed out that a similar deterioration in output transient response after a reference change can be observed for the MPE controllers based on minimization of a single-stage cost function, if the reference sequence value k samples ahead is unknown [5].

Let us now summarize the results of this example. For values of $C < P$ with

[4] This part of the example is referred to on p. 130 as an illustration of the effect of unknown future values of $\{r(t)\}$ on the closed-loop system performance with a controller based on the (nondiagonal) left interactor matrix.

[5] See discussion on p. 130.

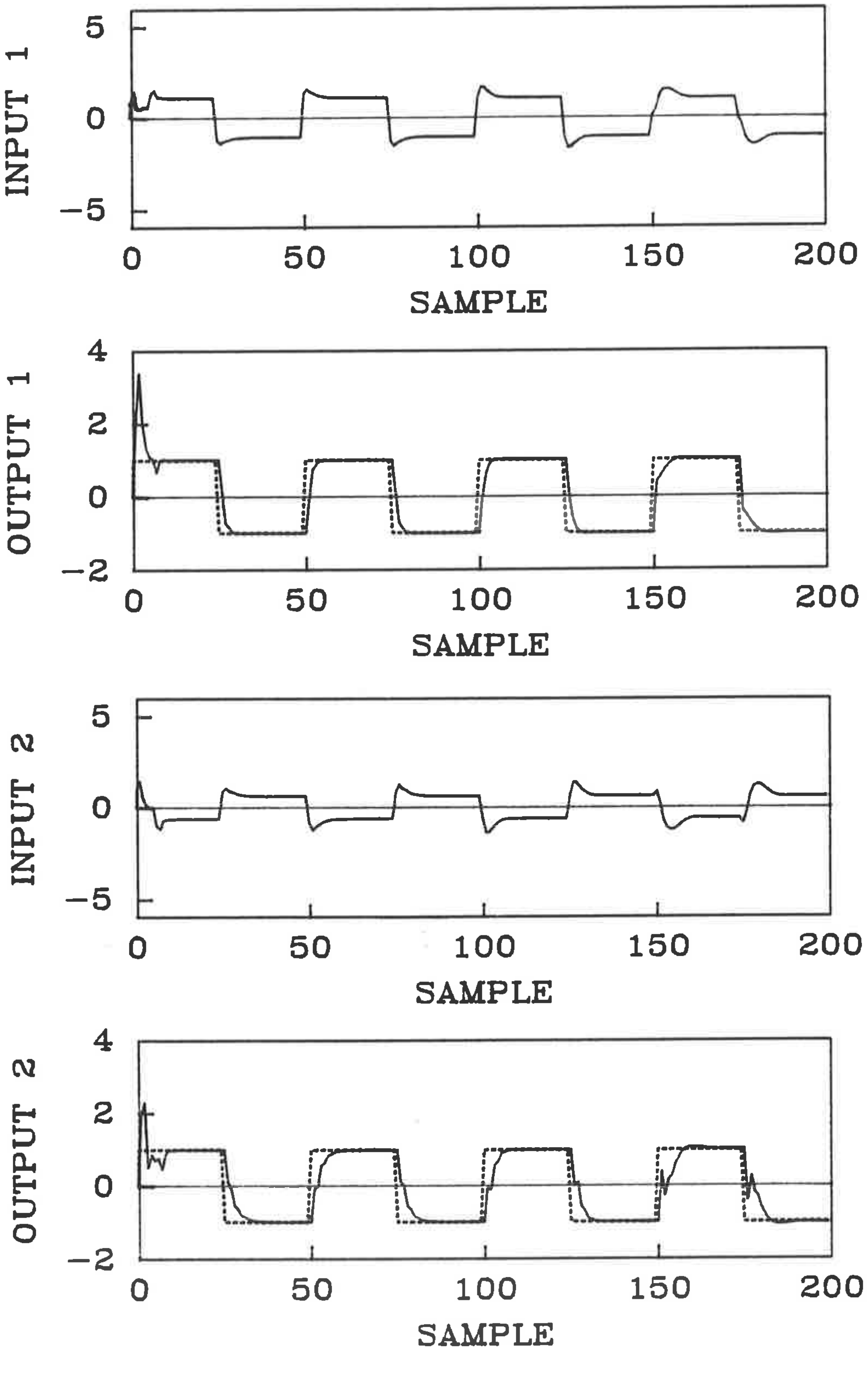

Figure 5.4: ST-LRP controller based on the LNI-LDU matrix with $C = 0$, $P = 7, 5, 3, 1$ (changing every 50 samples), $\Lambda = 0$: input, output and reference sequences (fixed reference sequence model).

$\Lambda = 0$, the ST-LRP controller applied to a nonminimum phase plant led to a stable closed-loop system. However, the choice $C = 0$ resulted in smoother output responses and involved a smaller computational burden (inversion of matrix B_{rT}). Larger values of P yielded slower responses, but with lower control effort. The time-oriented tuning knobs P and C offer greater flexibility in the design of the closed-loop system performance than the tuning knobs of the MPE controller. An appropriate choice of C and P might improve the closed-loop system performance in comparison with the MPE control for the case of unknown future reference sequence values.

Example 5.1.2. Application of the ST-LRP controller to robotics.

The adaptive control schemes based on the long-range predictive approach have been evaluated in applications involving slow sampling, varying from a few seconds to hours. Examples of applications to control chemical processes (catalytic cracking [30], PVC plant (synthesis of vinyl chloride) and distillation column [156,157]), blast-furnace process [118], speed control of a sinter strand [127], paper machine [158], and other industrial processes [159] were reported. Examples of applications involving fast sampling are rare [130]. It is demonstrated below that the ST-LRP strategy is capable of controlling systems requiring high sampling rates.

In section 4.3 the problem of designing a self-tuning controller for a robot manipulator was considered. The ST-MPE-RNI controller was successfully applied to a simulated two-link planar manipulator representing links 2 and 3 of the PUMA 560 robot. In the following example the same problem is addressed using the ST-LRP controller.

As in section 4.3, it is assumed that the robot manipulator can be adequately modeled by the DARMA model which approximates the nonlinear model of the manipulator. The sampling period is now chosen to be $T_s = 25\times10^{-3}$ seconds, i.e., the sampling rate is ten times higher than for the ST-MPE-RNI strategy and corresponds to typical values assumed in digital control of robots [141]. The linearized, discrete-time, state-space models and DARMA models corresponding to the operating points Q_1 and Q_2 (see eqns. 4.13 and 4.14, p. 139) were found as described in section 4.3

and are given in table 5.1.

The reference signal $r(t) = [r_1(t)\ r_2(t)]'$ (in radians) was defined as follows

$$r_1(t) = \begin{cases} 0.2\sin(\pi t) & \text{for } 0 \leq t < 2, \text{ and } 4 \leq t \leq 6 \text{ s} \\ 0.4\sin(\pi t) & \text{for } 2 \leq t < 4 \text{ s} \end{cases}$$

$$r_2(t) = \begin{cases} 0.5t & \text{for } 0 \leq t < 1, \text{ and } 4 \leq t \leq 6 \text{ s} \\ 0.5 & \text{for } 1 \leq t < 2 \text{ s} \\ -t & \text{for } 2 \leq t < 3 \text{ s} \\ -0.5 & \text{for } 3 \leq t < 4 \text{ s.} \end{cases}$$

The two-link manipulator was simulated using the DARMA model defined in table 5.1, at the operating point Q_1 for the period $0 \leq t < 1.5$ s, and at Q_2 for $1.5 \leq t \leq 6$ s. Assuming perfect tracking of the above reference sequence, the robot is at the operating point Q_1 at time $t = 0$ s, and at Q_2 at time $t = 1.5$ s. The rate of change of the above reference sequence is 6.25 times faster than that one specified for the ST-MPE-RNI scheme (see eqns. 4.16 and 4.17, p. 141).

Firstly, let us comment on the possibility of employing the ST-MPE-RNI strategy. Note that the roots of the equation $\det B(z^{-1}) = 0$ (denoted by z_{zi} in table 5.1) lie on the unit circle. Hence, the ST-MPE-RNI controller requires a nonzero weighting matrix Λ to satisfy closed-loop system stability conditions (see Assumption 4.1, p. 126). However, the choice of Λ is not straight-forward and, in fact, it was difficult to find appropriate Λ for fast sampling, such as $T_s = 25 \times 10^{-3}$ s. Furthermore, elimination of the λ-offset by the filter $S(p)$ introduces additional dynamics into the closed-loop system.

Let us now consider the ST-LRP strategy. It is demonstrated below that the ST-LRP controller yields satisfactory performance of the closed-loop system for a high sampling rate and with $\Lambda = 0$ ($S(p) = I_2$, $\Upsilon = I_2$).

The interactor matrix $\overline{K}_L(z) = zI_2$ is assumed to be known (see Assumption 5.1, p. 162). The following parameters of the ST-LRP controller were chosen: $C = 0$, $P = 3$ and $\Lambda = 0$. (Step responses of the nonlinear robot model are presented in [144] and show sustained oscillations.)

Model	at the operating point Q_1	at the operating point Q_2
A_s	$\begin{matrix} .9906 & .0074 & .0249 & .0001 \\ .0128 & -.9710 & .0001 & .0248 \\ -.7474 & .5861 & .9906 & .0074 \\ 1.0163 & -2.3103 & .0128 & -.9710 \end{matrix}$	$\begin{matrix} .9917 & .0050 & .0249 & .0000 \\ .0090 & -.9785 & .0001 & .0248 \\ -.6589 & .3941 & .9917 & .0050 \\ .7125 & -1.7153 & .0090 & -.9785 \end{matrix}$
B_s	$\begin{matrix} .0002 & -.0005 \\ -.0005 & .0017 \\ .0163 & -.0405 \\ -.0405 & .1367 \end{matrix}$	$\begin{matrix} .0002 & -.0004 \\ -.0004 & .0013 \\ .0131 & -.0303 \\ -.0303 & .1049 \end{matrix}$
C	$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$	$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$
$[A_1\, A_2]$	$\begin{matrix} -1.9813 & -.0147 & 1 & 0 \\ -0.0256 & -1.9419 & 0 & 1 \end{matrix}$	$\begin{matrix} -1.9835 & -.0099 & 1 & 0 \\ -0.0179 & -1.9569 & 0 & 1 \end{matrix}$
$[B_1\, B_2]$ $\times 10^{-2}$	$\begin{matrix} .0205 & -.0510 & .0205 & -.0510 \\ -.0510 & .1719 & -.0510 & .1719 \end{matrix}$	$\begin{matrix} .0165 & -.0380 & .0165 & -.0380 \\ -.0380 & .1317 & -.0380 & .1317 \end{matrix}$
z_{pi}	$0.9620 \pm 0.2549i$, $0.9946 \pm 0.1036i$	$0.9757 \pm 0.2190i$, $0.9945 \pm 0.1046i$
z_{zi}	$-1, -1$	$-1, -1$

Table 5.1: Matrices of the state-space and DARMA models of the robot manipulator linearized at the operating points Q_1 and Q_2 ($T_s = 25 \times 10^{-3}$ s).

The simplified RLS algorithm was initialized as in example 5.1.1. The limit on the control signal amplitude was assumed to be 75 Nm.

Fixed reference sequence model control.

Firstly, the fixed reference sequence model was employed. The input-output behaviour of the closed-loop system is shown in fig. 5.5. (The joint angle $\theta_1(t)$ is plotted around constant level of $-\frac{\pi}{2}$ rad taken as a zero level in fig. 5.5.) Both outputs track the reference sequence closely after the initial estimation phase of 50 samples. Note that there is a zero steady state error due to the choice of $\Lambda = 0$. The input-output performance was not affected by the transition form Q_1 to Q_2 at sample instant 60.

The trajectories of the estimates $\widehat{A}_i$ and $\widehat{B}_i$ $(i = 1, 2)$ are shown in figs. 5.6 and 5.7, respectively. The estimates of A_i converged closely to their true values after about 50 iterations, and those of B_i after about 30 samples. There were difficulties in following the step change of B_i parameters after transition form Q_1 to Q_2 at sample 60 (note that the elements of B_i are much smaller than those of A_i due to fast sampling [160, p. 57]).

The input-output performance was unaffected by the drift of estimates of B_i after sample 220. Such drifts of estimates can be often observed if a constant forgetting factor less then one is used (in this simulation $\alpha(t) = 0.95$). Sometimes the "burst of estimates", which is associated with exponential data weighting techniques, occurs [6]. If the major excitation of the controlled system is due to the reference sequence, there might be periods of poor excitation which often lead to the numerical instability of the estimator and possibly instability of the closed-loop system. There are many methods to avoid the "burst of estimates" [6]. Variable forgetting factors are commonly employed as a facility for preventing burst without loosing the ability to track variations in system parameters [146,11,147,148,13]. It is also important to enhance the robustness of the estimator against the "burst" by improving the numerical stability of the implemented algorithms. The square-root and U-D factorization techniques are commonly used for this purpose [161,162,163,164,165]. In this example, however, the RLS algorithm was implemented in the potentially ill-conditioned form

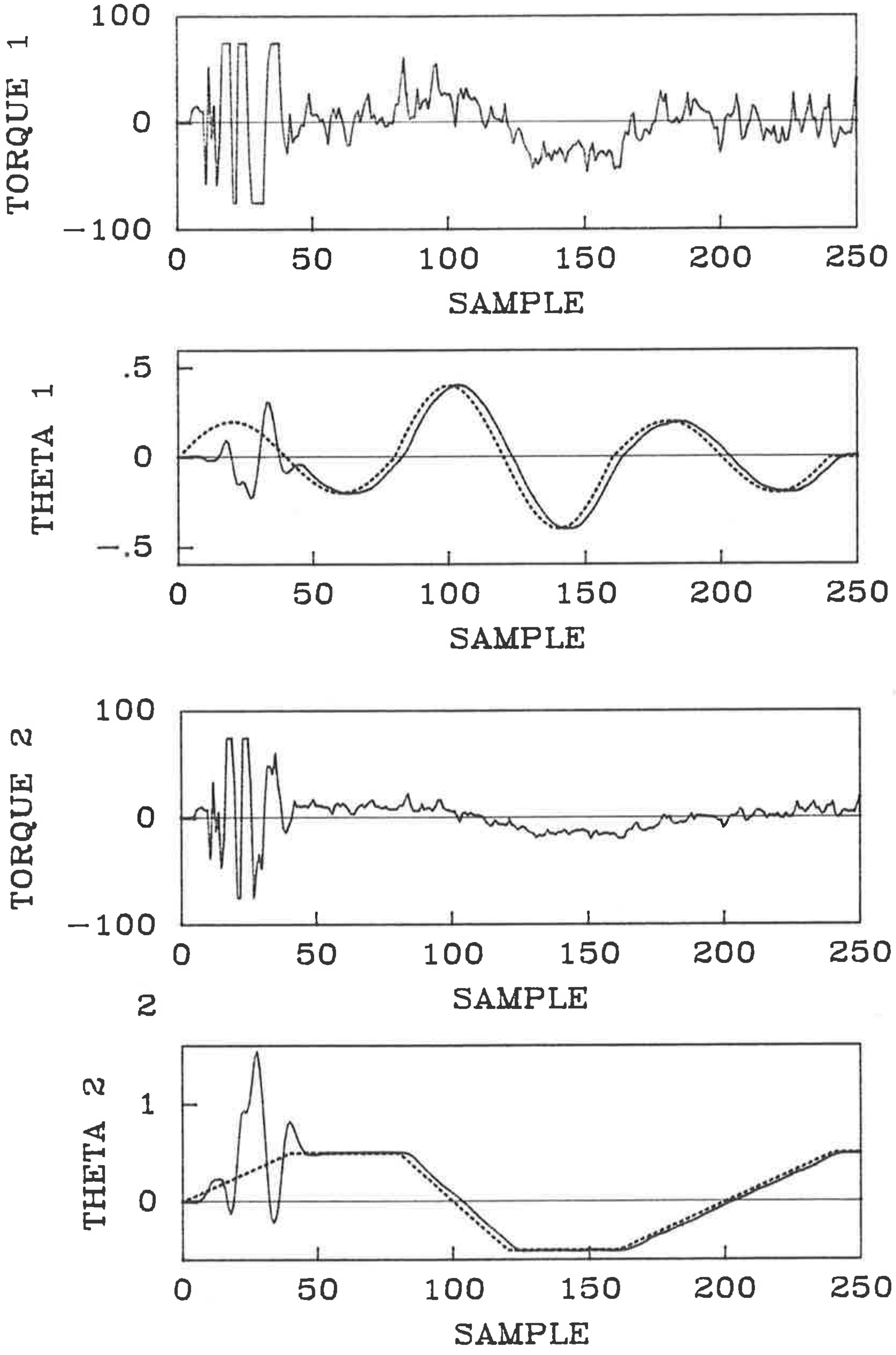

Figure 5.5: ST-LRP controller applied to the two-link robot manipulator $C = 0$, $P = 3$, $\Lambda = 0$: torques, angular joint positions $\theta_1(t)$, $\theta_2(t)$, and reference sequences (fixed reference sequence model).

given by eqns. E.21 and E.22, Appendix E).

The simulation was repeated for $P = 4$ ($C = 0$, $\Lambda = 0$) leading to satisfactory performance as well. For $P = 1$ and 2 the control effort was excessive, although it could be decreased by appropriate choice of the weighting matrix $\Lambda \neq 0$.

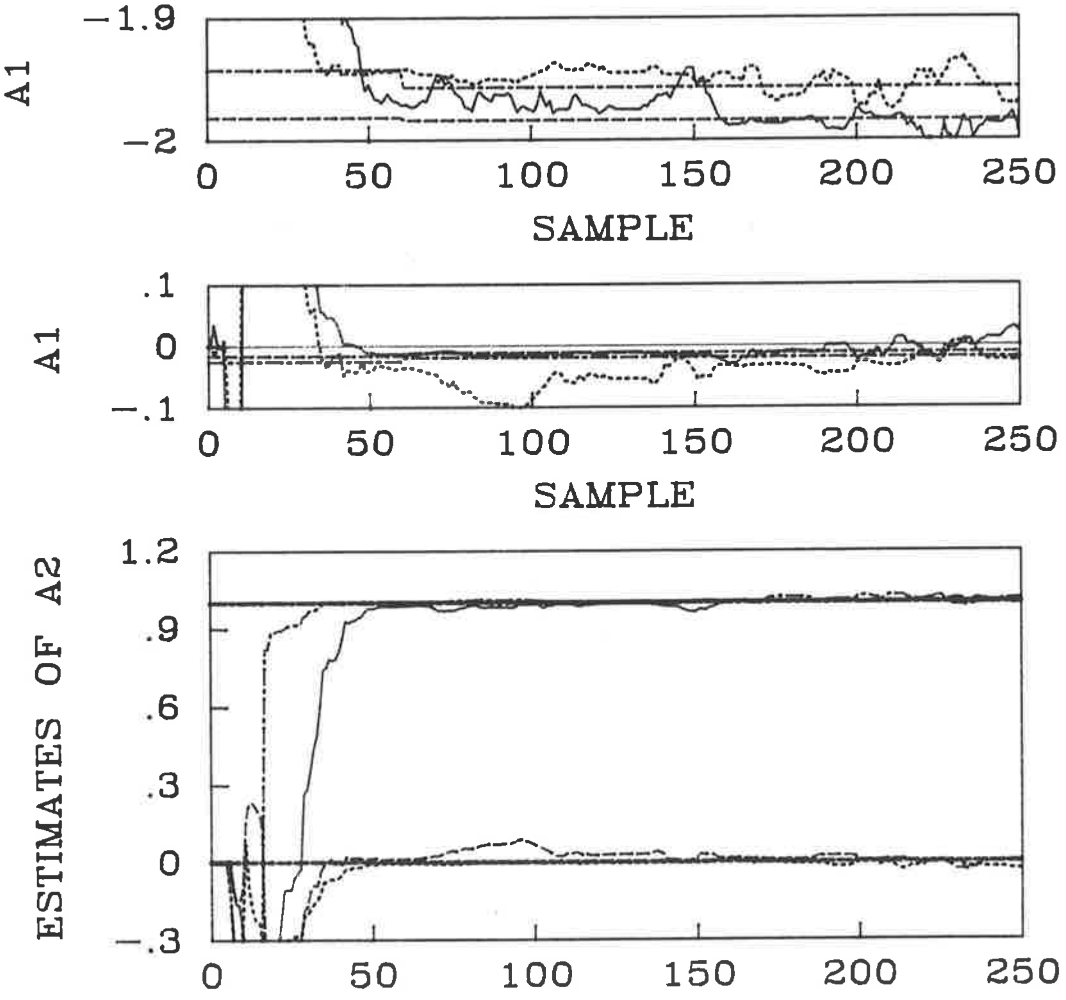

Figure 5.6: ST-LRP controller applied to the two-link robot manipulator $C = 0$, $P = 3$, $\Lambda = 0$: estimates and true values of the elements a_{11}, a_{22} (top), and a_{12}, a_{21} (middle) of the coefficient matrix A_1, and coefficient matrix A_2 (bottom) - fixed reference sequence model.

Programmed control.

In robot control, future values of the reference sequence are often known [141] and can be used in calculation of vector R according to eqn. 5.8 (programmed control).

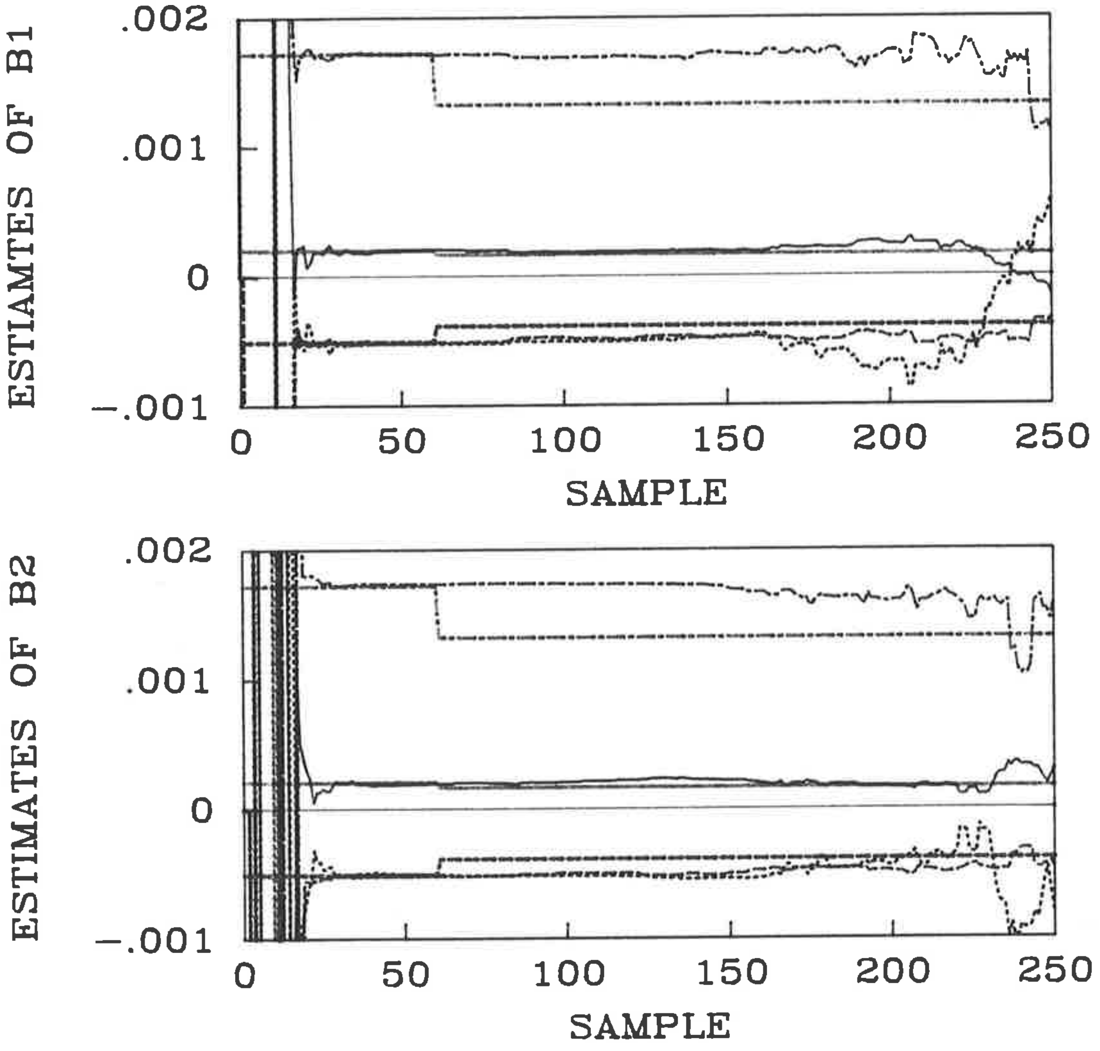

Figure 5.7: ST-LRP controller applied to the two-link robot manipulator $C = 0$, $P = 3$, $\Lambda = 0$: estimates and true values of the coefficient matrices B_1 and B_2 (fixed reference sequence model).

In order to illustrate the effect of programmed control on the system performance, the simulation was repeated for the same parameters of the ST-LRP scheme ($C = 0$, $P = 3$, $\Lambda = 0$) assuming knowledge of the $k + P$ future values of $\{r(t)\}$ at every sample instant t. The input-output behaviour is shown in fig. 5.8. To facilitate the comparison with the fixed reference sequence model, the same scale was chosen in fig. 5.8 as in 5.5 (the maximum values of joint angles were $\theta_1 = 1.15$ and $\theta_2 = 4.22$ rad, the minimum values were $\theta_1 = -1.25$ and $\theta_2 = -2.36$ rad). Note that the control sequences (torques) are significantly smoother than those associated with the fixed reference sequence model (see fig. 5.5). At the same time there is no degradation in

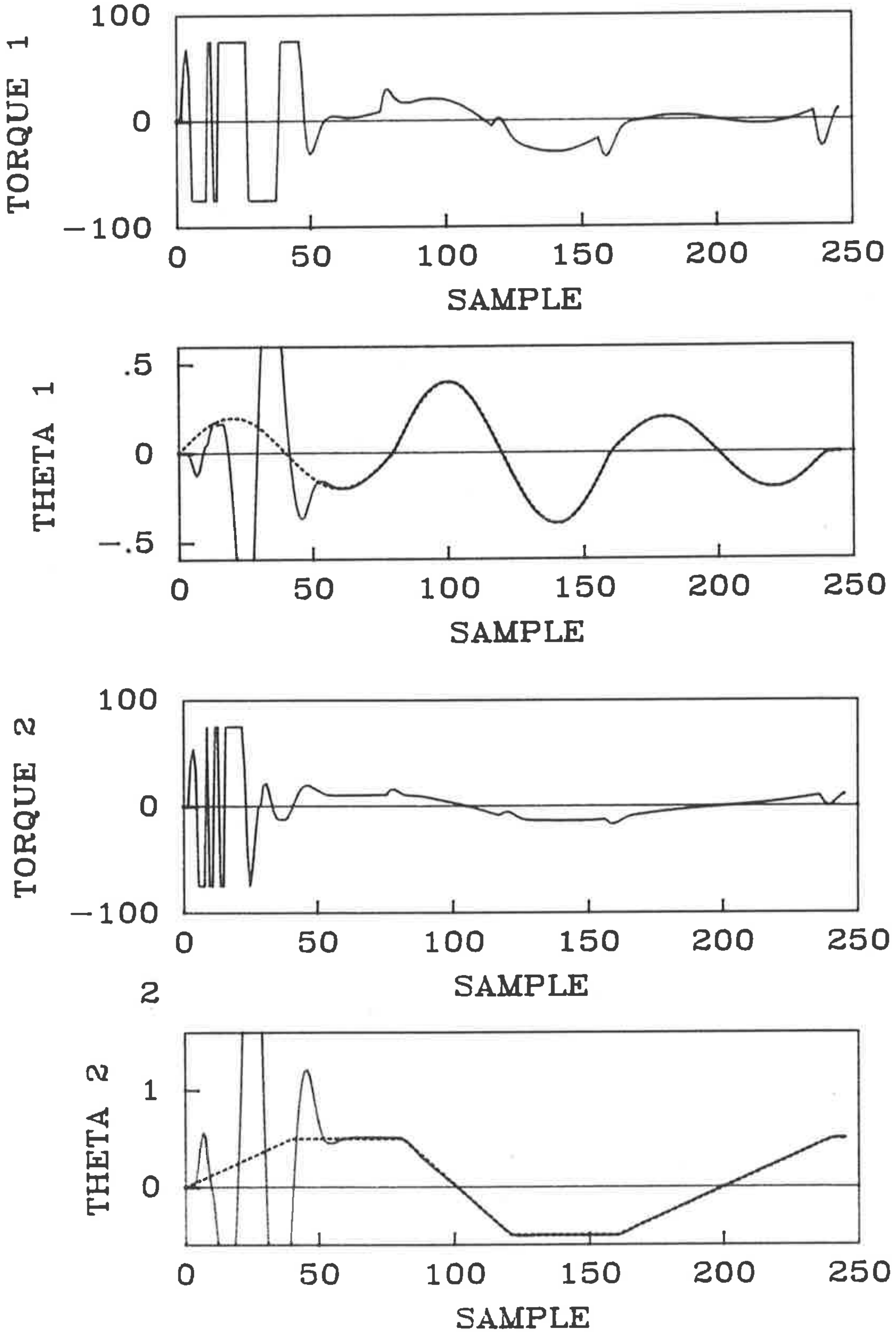

Figure 5.8: ST-LRP controller applied to the two-link robot manipulator $C = 0$, $P = 3$, $\Lambda = 0$: torques, $\theta_1(t)$, $\theta_2(t)$, and reference sequences (programmed control).

output responses. In fact, the outputs track the reference sequence closer than those for the fixed reference sequence model. The effect of programmed control on the control sequence is very desirable since it reduces component wear without affecting the output responses.

The effect of improving the control signal by programmed control seems to be quite general for systems with diagonal interactor matrices [6]. However, for systems with nondiagonal interactor matrices, programmed control leads to improved output responses but not necessarily to reduced control effort in comparison with the fixed reference sequence model (see example 5.1.1).

Summarizing, simulation studies presented in this example show that the time-oriented tuning knobs of the LRP controller facilitate control of dynamically-complex systems, such as robot manipulators. Furthermore, selection of values of the tuning knobs, which lead to satisfactory closed-loop system performance, is easier for the LRP controller than for the MPE strategy.

5.2 Self-tuning long-range predictive control of a two-input, single-output stochastic system.

In this section:

- the long-range predictive (LRP) control law is presented which minimizes multi-stage cost function for a stochastic scalar system;

- the stability analysis of the closed-loop system with the LRP controller is presented.

As already mentioned in the introduction to this chapter (see p. 149), the GPC strategy was introduced for the CARIMA model in [29,149] as an extension of the DMC strategy developed for the step-response plant model [30].

[6] A similar effect of programmed control was recently reported for scalar systems [94].

The GPC strategy has a number of properties which are desirable in self-tuning control [29]. However, the author found in the simulation studies that the performance of the closed-loop system under the GPC self-tuning controller deteriorates if the system satisfies the ARMAX model, rather than the CARIMA model [7] (see discussion on p. 32 and example 5.2.1, p. 185). Therefore, the corresponding LRP controller was developed for the ARMAX model following the approach used to derive the GPC [29] (see Rogoziński and Gibbard [150]). The summary of the results is presented in subsection 5.2.1. New results involve analysis of the stability and behaviour of the closed-loop system resulting from the LRP control law and are presented in subsection 5.2.2.

5.2.1 Long-range predictive control of a two-input, single-output stochastic system.

Let us consider a two-input ($l = \dim u(t) = 1$, $s = \dim \omega(t) = 1$), single-output stochastic system having the feedback configuration FI ($y_F(t) \equiv y(t)$, $m = f = \dim y(t) = 1$). The system is given by the ARMAX model 2.26 (p. 29) in which the delay $k \geq 1$ is factored from the polynomial $B(p)$ leading to the following form of the model

$$\begin{aligned} A(p)y(t) &= p^k\overline{B}(p)u(t) + C(p)\omega(t), \qquad (5.22)\\ A(p) &= [1 \;\; a_1 \;\; \ldots \;\; a_{n_A}]\, P_{n_A}, \\ \overline{B}(p) &= [b_0 \;\; b_1 \ldots \;\; b_{n_B}]\, P_{n_B}, \quad b_0 \neq 0 \\ C(p) &= [1 \;\; c_1 \;\; \ldots \;\; c_{n_C}]\, P_{n_C}, \end{aligned}$$

with initial conditions on $\{y(t)\}$. It is assumed that roots of $C(z^{-1})$ lie strictly inside the unit circle. The stochastic disturbance $\omega(t)$ is a zero-mean stationary white gaussian sequence (see eqns. 2.4 and 2.5, p. 21).

The LRP control law is developed with the aim to minimize the following cost

[7] This is confirmed by an article [94] published during the writing of this thesis.

function

$$J_{P,C} = E\left\{\sum_{j=0}^{P}[y(t+k+j)-r(t+k+j)]^2+\sum_{j=0}^{C}\lambda u(t+j)^2\right\}, \tag{5.23}$$

given the constraint on the variations of the control signal after time $t+C$, i.e.,

$$(1-p)u(t+C+i)=0 \quad \text{for} \quad i=1,\dots,P-C,$$

where the expectation operator $E\{.\}$ is conditioned on data available up to time t. Note that cost function 5.23 is a stochastic counterpart of the performance criterion 5.3 (p. 153) for a scalar system [8]. Therefore, the LRP control law, resulting from minimization of the cost 5.23, has the same parameters P, C, and control weighting coefficient λ, as its counterpart derived for a MIMOS deterministic system in section 5.1.

The derivation of the LRP control law presented below for the ARMAX model is based on that of the GPC for the CARIMA model in [29].

For the purpose of derivation of the LRP controller the optimal multi-step-ahead prediction of the output $y(t+k_p)$ of the system 5.22, introduced in [127], is expressed in the following form ($k_p \geq k$)

$$y^o(t+k_p|t) = \alpha^{(k_p)}(p)y_c(t)+\beta_F^{(k_p)}(p)u(t+k_p-k)+\beta_G^{(k_p)}(p)u_c(t-1), \tag{5.24}$$

where the filtered system output $y_c(t)$ and input $u_c(t)$ are defined by

$$C(p)y_c(t)=y(t), \quad C(p)u_c(t)=u(t). \tag{5.25}$$

The polynomial $\alpha^{(k_p)}(p) = \left[\alpha_0^{(k_p)} \ \dots \ \alpha_{n_A-1}^{(k_p)}\right] P_{n_A-1}$, satisfies the following Diophantine equation

$$C(p) = F^{(k_p)}(p)A(p)+p^{k_p}\alpha^{(k_p)}(p), \tag{5.26}$$

where $F^{(k_p)}(p)$ is a polynomial of degree k_p-1. The polynomials $\beta_F^{(k_p)}(p)$ and $\beta_G^{(k_p)}(p)$ are defined as follows

$$\beta_F^{(k_p)}(p) \quad = \quad \beta^{(k_p)}(p)f^{(k_p-k+1)}(p)$$

[8] The cost function weighting increments in the control signal, rather than its magnitude, was considered as well for the development of the LRP controller in Rogoziński and Gibbard [150].

$$
\begin{aligned}
&= \left[\beta^{F(k_p)}_{-k_p+k} \ \dots \ \beta^{F(k_p)}_{-1} \ \beta^{F(k_p)}_{0} \ \dots \ \beta^{F(k_p)}_{n_B+k_p-1}\right] P_{n_B+2k_p-k-1}, \\
\beta^{(k_p)}_G(p) &= \beta^{(k_p)}(p) g^{(k_p-k+1)}(p) = \left[\beta^{G(k_p)}_{0} \ \dots \ \beta^{G(k_p)}_{n_B+n_C+k_p-2}\right] P_{n_B+n_C+k_p-2},
\end{aligned}
$$

where $\beta^{(k_p)}(p) = F^{(k_p)}(p)\overline{B}(p)$, and the polynomials $f^{(k_p-k+1)}(p)$ (of degree $k_p - k$) and $g^{(k_p-k+1)}(p)$ (of degree $n_C - 1$) satisfy the following Diophantine equation

$$1 = f^{(k_p-k+1)}(p)C(p) + p^{k_p-k+1} g^{(k_p-k+1)}(p). \tag{5.27}$$

If the initial conditions on $\{y(t)\}$ are known and used in calculation of $y^o(t+k_p|t)$, then the prediction 5.24 is optimal for all sample instants t. If the initial conditions are unknown, then the prediction 5.24 is asymptotically optimal since the effect of incorrect initial conditions diminishes exponentially. (This is because $C(z^{-1})$ has all roots strictly inside the unit circle [9, Remark 7.4.3, p. 269].)

The prediction 5.24 with $k_p = k + j$ is substituted with the corresponding prediction error $\epsilon(t + k_p) = y(t + k_p) - y^o(t + k_p|t) = F^{(k_p)}(p)\omega(t + k_p)$ for the output $y(t+k+j)$ in the cost function 5.23. It is assumed that $E\{u(t+i)\epsilon(t+k+j)\} = 0$ for $j = 0, \dots, P$ and $i \leq j$. The cost function is minimized with respect to the present and future control signals U_t, defined as

$$U_t = [u(t) \ \dots \ u(t+C)]', \tag{5.28}$$

leading to the following LRP control law

$$U_t = B_{rt} \left[\begin{array}{cccc} I_{P+1} & -A_p & -B_p & -B_g \end{array} \right] \left[\begin{array}{c} R_s \\ Y_c \\ U_p \\ U_c \end{array} \right], \tag{5.29}$$

where

$$
\begin{aligned}
R_s &= [r(t+k) \ \dots \ r(t+k+P)]', \qquad (5.30) \\
Y_c &= [y_c(t) \ \dots \ y_c(t-n_A+1)]', \\
U_p &= [u(t-1) \ \dots \ u(t-n_B-k-P+1)]', \\
U_c &= [u_c(t-1) \ \dots \ u_c(t-n_B-n_C-k-P+1)]'.
\end{aligned}
$$

The real matrices involved in the LRP control law 5.29 consist of coefficients of the polynomials of the multi-step-ahead predictor 5.24 for $k_p = k, \ldots, k+P$, and are defined below. The $(C+1) \times (C+1)$ matrix B_{rt} is given by

$$B_{rt} = \left(B'_{fT} B_{fT} + \lambda I_{C+1}\right)^{-1} B'_{fT},$$

where the $(P+1) \times (C+1)$ matrix $B_{fT} = B_f T$ (T is given by eqn. 5.14 (p. 155) with $m = 1$), and the $(P+1) \times (P+1)$ matrix B_f is defined as

$$B_f = \begin{bmatrix} \beta_0^{F(k)} & 0 & 0 & \cdots & 0 \\ \beta_0^{F(k+1)} & \beta_0^{F(k)} & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ \beta_0^{F(k+P-1)} & \beta_0^{F(k+P-2)} & \cdots & \beta_0^{F(k)} & 0 \\ \beta_0^{F(k+P)} & \beta_0^{F(k+P-1)} & \cdots & \beta_0^{F(k+1)} & \beta_0^{F(k)} \end{bmatrix}.$$

Furthermore, the $(P+1) \times (n_B + k + P - 1)$ matrix B_p is defined as

$$B_p = \begin{bmatrix} \beta_1^{F(k)} & \cdots & \beta_{n_B+k-1}^{F(k)} & 0 & \cdots & 0 \\ \vdots & & & \ddots & \ddots & \vdots \\ \beta_1^{F(k+P-1)} & & \cdots & & \beta_{n_B+k+P-2}^{F(k+P-1)} & 0 \\ \beta_1^{F(k+P)} & & \cdots & & & \beta_{n_B+k+P-1}^{F(k+P)} \end{bmatrix},$$

and the $(P+1) \times (n_B + n_C + k + P - 1)$ matrix B_g is given by

$$B_g = \begin{bmatrix} \beta_0^{G(k)} & \cdots & \beta_{n_B+n_C+k-2}^{G(k)} & 0 & \cdots & 0 \\ \vdots & & & \ddots & \ddots & \vdots \\ \beta_0^{G(k+P-1)} & & \cdots & & \beta_{n_B+n_C+k+P-3}^{G(k+P-1)} & 0 \\ \beta_0^{G(k+P)} & & \cdots & & & \beta_{n_B+n_C+k+P-2}^{G(k+P)} \end{bmatrix},$$

and the $(P+1) \times n_A$ matrix A_p is

$$A_p = \begin{bmatrix} \alpha_0^{(k)} & \cdots & \alpha_{n_A-1}^{(k)} \\ \vdots & & \vdots \\ \alpha_0^{(k+P)} & \cdots & \alpha_{n_A-1}^{(k+P)} \end{bmatrix}.$$

The coefficients of polynomials $\alpha^{(k_p)}(p)$, $\beta^{(k_p)}(p)$, $f^{(k_p-k+1)}(p)$, and $g^{(k_p-k+1)}(p)$, which are required to determine the above matrices, can be calculated recursively using an algorithm similar to that given in Remark 3.2 (p. 69). This reduces the computational burden required for solution of the two Diophantine equations 5.26 and 5.27 for each prediction horizon $k_p = k, \ldots, k+P$.

The guidance based on the simulation studies for the choice of the LRP controller parameters C, P, and λ, was discussed in [150] and is presented below [9]. (Corresponds results were obtained for the DMC [30,151,152] and for the GPC [29] strategies.) Both C and P can be chosen on the basis of the step response of the process to be controlled. It was found that for well-damped plants the choice $C = 0$ was satisfactory. It may be necessary to set $C > 0$ to obtain the desired performance for more dynamically-complex plants, e.g., for a nonminimum phase plant having an oscillating unstable mode. The second tuning knob P facilities the adjustment of the speed of the response. The larger the value of P the slower is the response and the more "cautious" is the control. For well-damped systems the choice of P is not crucial and can usually be such that $(k+P)T_s$ is approximately equal to the open-loop system rise-time (T_s is the sampling interval). For nonminimum phase plants $(k+P)T_s$ should cover at least the rise-time of the open-loop system. For the control of an oscillatory unstable system, $(k+P)T_s$ should not be less than the period of the first oscillation. The choice of λ is not crucial but provides a means for containing excessive control effort.

The *self-tuning LRP controller* results by combining a recursive parameter estimation algorithm with the LRP control law 5.29. The recursive extended least squares (RELS) algorithm [9, p. 319] can be employed to estimate coefficients of the polynomials $A(p)$, $\overline{B}(p)$, and $C(p)$ of the ARMAX model 5.22.

The simulation studies described in [150] demonstrate the superior robustness of the self-tuning LRP controller to underparameterization in comparison with the pole placement techniques of [44], and to variable system delay k in comparison with the generalized minimum variance strategy of [24]. Similar results were originally reported for the GPC in [29].

[9] The analytical results are presented in subsection 5.2.2.

It was assumed that roots of $C(z^{-1})$ lie strictly inside the unit circle. This is required in view of eqn. 5.25, but excludes the possibility of describing a system with a deterministic disturbance, the generator model of which is incorporated in the ARMAX model (see discussion on p. 31) [10]. However, the ARMAX model augmented with the offset term (see eqn. 2.32, p. 31) can be considered for the development of the LRP control law. A number of methods of offset elimination suitable for self-tuning control was proposed [10,84,85] and two of them (classified in [85] as offset estimation and incremental estimation) have been incorporated in the self-tuning LRP controller under discussion (Rogoziński and Gibbard [150]).

Example 5.2.1.

Let us consider a stable, nonminimum phase system given by the ARMAX model 5.22 with $a_1 = -0.95$, $b_0 = 1$, $b_1 = 2$, $c_1 = -0.7$, and $k = 2$. This system model was used in [23, example 3] to illustrate the performance of the generalized minimum variance (GMV) self-tuning regulator based on minimization of a single-stage cost function. In [23], the stability of the GMV regulator was guaranteed by selecting a value of the control weighting coefficient λ greater than a critical value; however the ARMAX model coefficients must be known for the calculation of the latter value. In the following simulation it is shown that the LRP self-tuning regulator leads to a similar closed-loop system performance as that of the GMV regulator. Furthermore, this is achieved by adjustment of two time-oriented controller parameters C and P. It is also shown that the regulation performance resulting from the GPC technique is unsatisfactory.

The same simulation conditions as in [23] were chosen: the RELS estimator [9, p. 319] was initialized with the covariance matrix $P(-1) = 100I$, and initial parameter estimates were set to zero except for $\widehat{b}_0(0) = 1$. Constant forgetting factor was used $\alpha = 0.995$. The control signal magnitude was restricted to avoid excessive

[10] Alternatively the (suboptimal) restricted complexity multi-step-ahead predictor could be considered for derivation of the LRP control law (the k-step-ahead predictor of this type was introduced in [9, p. 272]).

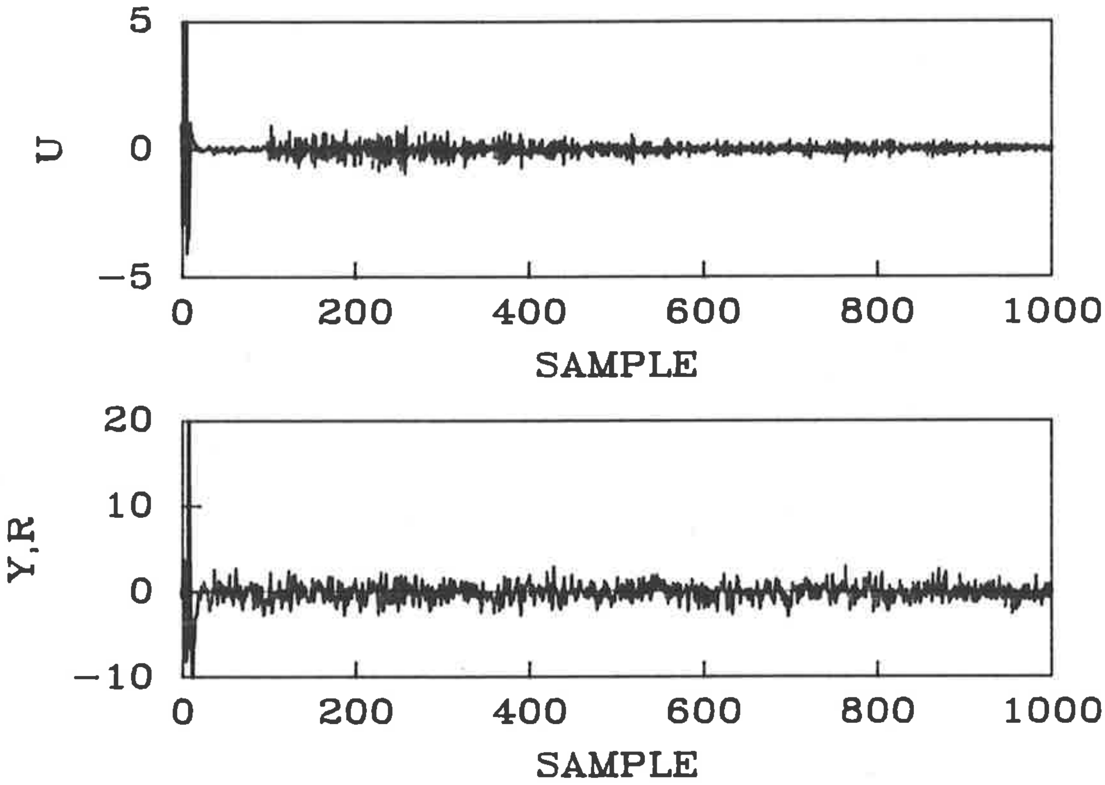

Figure 5.9: The sequences of the control signal $u(t)$ and regulated output $y(t)$ resulting from the LRP self-tuning regulator: $C = 0, 1, 2, 3$ (changing every 100 samples and constant thereafter), $P = 4$, $\lambda = 0$.

control action $|u(t)| \leq 10$. The input noise variance was set to $\sigma^2 = 1$.

For a prediction horizon $P = 4$, the control horizon was changed every 100 samples in the sequence $C = 0, 1, 2, 3$, and remained constant thereafter. The control weighting coefficient λ was set to zero.

The input-output behaviour of the closed-loop system is shown in fig. 5.9. It can be observed that all values of $C < P$ lead to stable control, and a significant difference in the control signal amplitude exists only between two cases $C = 0$ and $C > 0$. In fact it was found that the choice of C had a negligible effect on the output variance provided $C < P$ was selected (for $C = P$ and $\lambda = 0$, the LRP control law becomes the MV control law leading to instability for a nonminimum phase system).

In order to assess the convergence of the self tuner, the difference cumulative loss

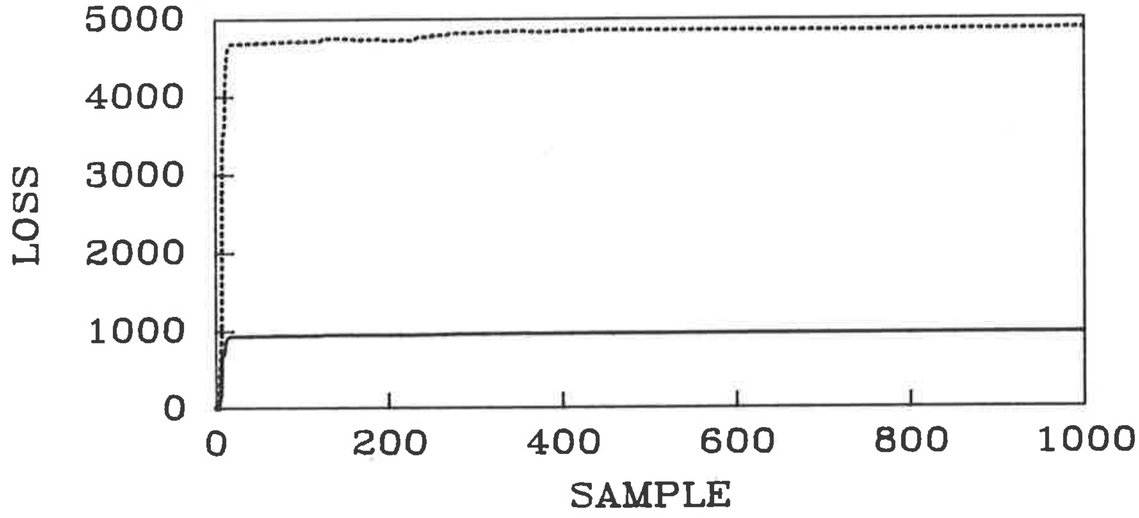

Figure 5.10: The difference cumulative loss (solid line) and vector loss (broken line) functions for the LRP self-tuning regulator: $C = 0, 1, 2, 3$ (changing every 100 samples and constant thereafter), $P = 4$, $\lambda = 0$.

function was calculated as

$$CL(t) = \sum_{\tau=1}^{t} \left[y(\tau)^2 - y_o(\tau)^2 \right] ,$$

where $y_o(t)$ is the system output resulting from the non-adaptive LRP controller designed for the true system parameters. This function is shown in fig. 5.10 together with the difference cumulative vector loss function defined as

$$CVL(t) = \sum_{\tau=1}^{t} \left[\sum_{i=0}^{P} \left[y(\tau+i)^2 - y_o(\tau+i)^2 \right] \right] .$$

In addition the output autocorrelation function was evaluated over the last 501 samples as

$$\frac{1}{501} \sum_{500}^{1000} y(t)y(t+s), \quad s = 0, \ldots, 10.$$

In fig. 5.11 the output autocorrelation functions are displayed (1) for the open-loop system (symbol x), and for the closed-loop system (2) with a non-adaptive regulator designed for the true open-loop system parameters (symbol o), and (3) with the LRP self tuner (symbol +). Both difference loss functions and the output autocorrelation

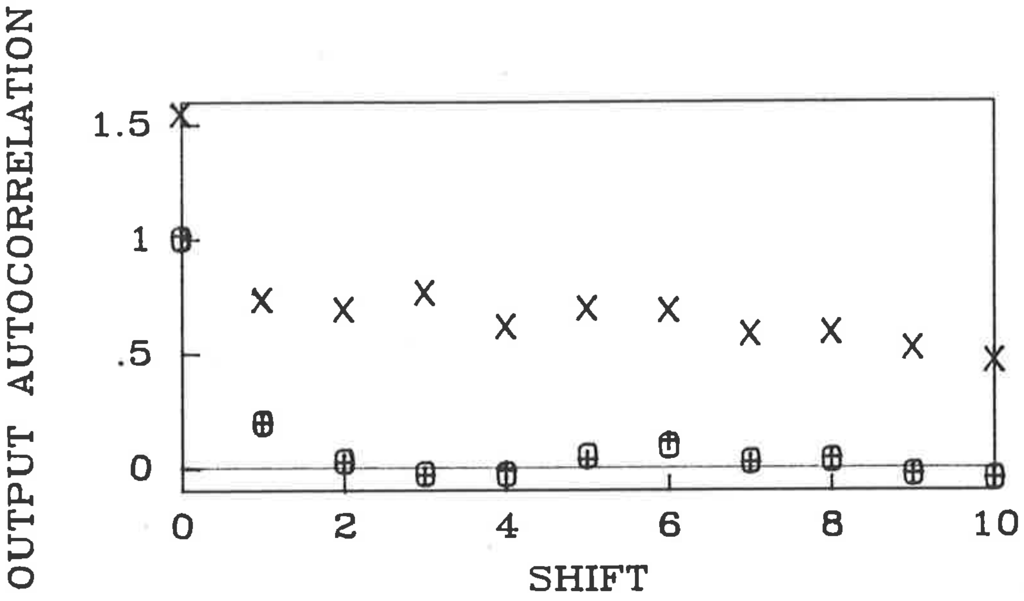

Figure 5.11: The output autocorrelation functions for the open-loop system (symbol x), and for the closed-loop system with the LRP non-adaptive regulator (symbol o) and self-tuning regulator (symbol +), with $C = 3$, $P = 4$, $\lambda = 0$.

function demonstrate satisfactory convergence of the self tuner. Good regulation by the LRP self tuner is revealed by the output variance (calculated for the last 501 samples) being close to unity.

For the purpose of comparison, the GPC non-adaptive regulator developed for the CARIMA model [29] was simulated for the same system. The difference in the noise characteristics (recall that the CARIMA model assumes that the noise filter has a pole on the unit circle) led to a significant deterioration of the closed-loop system performance. For $C = 0$, $P = 4$ and $\lambda = 0$, the output variance was $\sigma^2 = 1.75$; the choice of $C = 3$ increased the variance to 2.77.

5.2.2 Long-range predictive control – closed-loop system analysis.

It should be noted that although similar LRP controllers (DMC and GPC) were developed and proven useful in industrial applications, the closed-loop system analysis and stability conditions were not presented. With the application of the LRP controllers to nonminimum phase and/or unstable open-loop systems a means for establishing stability is needed [11].

The ARMAX model 5.22 was assumed for the derivation of the self-tuning LRP controller in subsection 5.2.1. However, for the purpose of the closed-loop system analysis it is convenient to derive an equivalent LRP control law from the corresponding state-space model given by eqn. 2.7 (p. 22). Let us consider the observer form of the state-space model having *implicit delay* [74]

$$\begin{bmatrix} qx(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A_s & K & B_s \\ C & 1 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \omega(t) \\ u(t) \end{bmatrix}, \tag{5.31}$$

where

$$A_s = \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & & \vdots \\ -a_{n-1} & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ -a_n & 0 & \dots & 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & 1 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad B_s = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b_0 \\ \vdots \\ b_n \end{bmatrix}, \quad K = \begin{bmatrix} k_1 \\ \vdots \\ k_n \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$
$$C = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}. \tag{5.32}$$

The dimension of the state space is $n+k$, where $n = \max(n_A, n_B, n_C)$, and $k_i = c_i - a_i$ for $i = 1, \dots, \max(n_A, n_C)$ ($a_i = 0$, $b_j = 0$, $k_l = 0$ for $i > n_A$, $j > n_B$, $l > \max(n_A, n_C)$).

Let us derive the LRP control law for the system given by the state-space model 5.31. For this purpose we need the optimal multi-step-ahead output prediction expressed in terms of the system state-space model as follows [9, p. 262]

$$E\{y(t+k_p)|y(0)\dots y(t)\} =$$

[11] Difficulties encountered in attempts to stability analysis of the GPC strategy are pointed out in [49].

$$= \quad CA_s^{k_p}\hat{x}(t|t-1) + \sum_{i=0}^{k_p-1} CA_s^{k_p-1-i}B_s u(t+i) + CA_s^{k_p-1}K\hat{\omega}(t), \qquad (5.33)$$

where $\hat{x}(t|t-1)$ is the estimate of state $x(t)$ given data up to time $t-1$, resulting from the Kalman filter for the system 5.31 and $\hat{\omega}(t) = y(t) - C\hat{x}(t|t-1)$ is the innovations sequence. The Kalman filter for the system innovations model 5.31 is given by [46]

$$\hat{x}(t+1|t) = A_s\hat{x}(t|t-1) + B_s u(t) + K\hat{\omega}(t).$$

Furthermore, let us assume that the initial conditions on $\{y(t)\}$ in model 5.22 are known so that the multi-step-ahead prediction 5.24 is optimal for all sample instants t. The corresponding initial condition in the state-space model 5.31 is $x(0)$. The above Kalman filter is initialized with $\hat{x}(0|-1) = x(0)$; then $\hat{x}(t|t-1) = x(t)$ and $\hat{\omega}(t) = \omega(t)$ for all t [46]. Furthermore, considering the property of Markov parameters for a system with delay of k sample intervals one has $CA_s^{j-1}B_s = 0$ for $j = 1, \ldots, k-1$ [80]. Therefore, the optimal multi-step-ahead output prediction, expressed in terms of the state-space model 5.31 for $k_p \geq k$, follows from eqn. 5.33

$$y^o(t+k_p|t) = CA_s^{k_p}x(t) + \sum_{i=0}^{k_p-k} CA_s^{k_p-1-i}B_s u(t+i) + CA_s^{k_p-1}K\omega(t). \qquad (5.34)$$

The prediction 5.34 is used in the derivation of the LRP control law instead of the prediction 5.24. This leads to the following expression for the control sequence U_t (eqn. 5.28) minimizing the cost function 5.23

$$U_t = B_{st}\left(R - B_X x(t)\right), \qquad (5.35)$$

where the vector of future reference sequence values R_s is defined by eqn. 5.30, and the $(C+1) \times (P+1)$ matrix B_{st} is given by

$$B_{st} = \left(B'_{ST}B_{ST} + \lambda I_{C+1}\right)^{-1} B'_{ST},$$

where the $(P+1) \times (C+1)$ matrix $B_{ST} = B_S T$ (T is given by eqn. 5.14 (p. 155) with $m = 1$), with the $(P+1) \times (P+1)$ matrix B_S and the $(P+1) \times (n+k-1)$

matrix B_X, defined as follows

$$B_S = \begin{bmatrix} CA_s^{k-1}B_s & 0 & \dots & 0 \\ CA_s^kB_s & CA_s^{k-1}B_s & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ CA_s^{k+P-1}B_s & CA_s^{k+P-2}B_s & \dots & CA_s^{k-1}B_s \end{bmatrix}, \quad B_X = \begin{bmatrix} CA_s^k \\ CA_s^{k+1} \\ \vdots \\ CA_s^{k+P} \end{bmatrix}.$$

We assume that the LRP control law is used in the receding horizon sense. Hence $u(t) = \mu U_t$, where the $1 \times (C+1)$ vector μ is defined as $\mu = [1 \ \ 0 \ \ \dots \ \ 0]$. Substituting from eqn. 5.35 for the control signal in eqn. 5.31, one has the following state-space model of the closed-loop system resulting from the LRP control law 5.35

$$\begin{bmatrix} qx(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} \tilde{A} & K & \tilde{B} \\ C & 1 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \omega(t) \\ u_R(t) \end{bmatrix}, \tag{5.36}$$

where the $(P+1) \times 1$ input vector $u_R(t) = R_s = [r(t+k) \ \ \dots \ \ r(t+k+P)]'$, and

$$\begin{aligned} \tilde{A} &= A_s - B_s \mu B_{st} B_X, \\ \tilde{B} &= B_s \mu B_{st}. \end{aligned} \tag{5.37}$$

The transfer function relating output $y(t)$ to the noise $\omega(t)$ can be found from the above state-space model. Then the variance of the output regulated with the LRP controller can be calculated using the algorithm introduced in [83, section 5.2]. The influence of the parameters of LRP control law P, C, and λ, on the location of the closed-loop system poles can be examined by calculation of eigenvalues of the matrix $\tilde{A}$ (see example 5.2.2). Furthermore, we can conclude that

Lemma 5.2 *The closed-loop system resulting from the LRP control law 5.29 (or equivalently from 5.35) used in the receding horizon sense, is asymptotically stable provided the $(n+k) \times (n+k)$ state transition matrix $\tilde{A}$ (eqn. 5.37) has all its eigenvalues inside the unit circle.*

The above lemma establishes stability condition which can be easily verified for a system with known parameters because of the direct relationship between the polynomials of the ARMAX model 5.22 and matrices 5.32 of the state-space model.

Example 5.2.2.

For the system considered in example 5.2.1, the poles of the closed-loop system resulting from the LRP control law will be calculated from eqn. 5.37. For various values of C, the closed-loop pole lies at the following locations: $C = 0$, $z = 0.6621$; $C = 1$, $z = -0.1925$; $C = 2$, $z = -0.4433$; $C = 3$, $z = -0.4914$. For $C = P = 4$, the LRP controller reduces to the minimum variance controller, which assigns the closed-loop poles to open-loop zeros [83], i.e., $z = -2$, leading to unstable closed-loop system.

Let us now illustrate the relationship between the LRP and the generalized minimum variance (GMV) [23] controllers. For this purpose the values of the control weighting coefficient λ required by the GMV controller to assign the closed-loop system poles to the positions resulting from the choice of $P = 4$, $\lambda = 0$, and $C = 0, 1, 2, 3$ for the LRP controller, were found to be $\lambda = 9.25$; 1.5821; 1.117; 1.0465. The major difference between the first and subsequent values of λ associated with the GMV controller corresponds to the cases $C = 0$ and $C > 0$ for the LRP controller. This explains why the significant difference in the amplitude of the control signal produced by the LRP controller was observed only between two cases $C = 0$ and $C > 0$ (see fig. 5.9).

5.3 Concluding remarks.

In this chapter, the long-range predictive (LRP) control based on minimization of the *multi-stage* performance criteria is considered as an alternative to the minimum prediction error (MPE) control based on minimization of the *single-stage* cost functions, in application to self-tuning control.

The new LRP control laws have been developed for *multivariable deterministic* systems and *scalar stochastic* systems as extensions to the DMC and GPC approaches. The LRP controllers facilitate control of a wide *variety of systems* (in particular, nonminimum phase and/or unstable open-loop systems).

The LRP controllers, derived for the DARMA and ARMAX system models, possess features desirable in self-tuning control which are *not* available for the DMC strategy based on the step-response plant representation. These features involve: (i) the LRP controller can be applied to open-loop *unstable* systems, (ii) design of the LRP controller is based on the characterization of a multivariable system delay structure by the left interactor matrix (this guarantees that the control law is well-defined without *ad hoc* solutions to ensure invertibility of B_{rT}), (iii) loss of accuracy resulting from using the truncated step-response model is *avoided* by using the DARMA or ARMAX models (iv) the DARMA and ARMAX models involve, in general, *fewer* parameters for adequate system representation than the step-response model (this improves convergence rate of the LRP self tuners), (v) elimination of the effect of *deterministic disturbances*, (vi) treatment of systems subject to deterministic (offset) and *stochastic disturbances*, (vii) rigorous *stability analysis* (presented for scalar stochastic systems).

The LRP strategies are characterized by two *time-oriented* parameters C and P, and the control weighting parameter Λ. The choice of $C = 0$ was found in the simulation studies to result in satisfactory closed-loop system performance for many systems. Furthermore, $\Lambda = 0$, and $C < P$, with P such that $(k + P)T_s$ covers a significant part of the open-loop system step response, lead to stable closed-loop system for nonminimum phase plants. On the other hand, the MPE strategies rely on the choice of the control weighting parameter $\Lambda \neq 0$ for such plants. The choice of Λ is not straight-forward in self-tuning MPE control since it is related to the plant (unknown) parameters.

The LRP control law facilitates design of the input-output performance by the choice of its time-oriented parameters, a feature not available for the MPE controllers based on the single-stage cost function.

The computational burden of the LRP controller is greater than that of the MPE controllers (based on the assumption of prior knowledge of the interactor matrix). However, if $C = 0$ then the dimension of the matrix which is to be inverted at each sample instant is the same for both the LRP and MPE controllers.

If future values of the reference sequence are known, then the LRP controller facilitates *programmed control*, a feature desirable in robot control. For systems with diagonal interactor matrices the programmed control *reduces* control effort without degradation of output responses. If future values of the reference sequence are unknown, then the input-output performance deteriorates for systems with nondiagonal interactor matrices.

The main *disadvantage* of the LRP controller for multivariable systems is the requirement of prior knowledge of the interactor matrix. This drawback has been overcome by another long-range predictive control strategy proposed in [38]. For the latter strategy, only an upper bound on the degree of the interactor matrix must be known *a priori*. However, the cost function considered in [38] penalizes only one future value of the output (similarly to the single-stage cost functions). As a result, the strategy may fail to stabilize open-loop *unstable* systems, as indicated in [29] for scalar systems.

The LRP control strategy for scalar stochastic systems modeled by the ARMAX model was introduced following the approach proposed for systems modeled by the CARIMA model (the GPC strategy). For systems which are adequately represented by the ARMAX model, the LRP control *outperforms* the GPC strategy in elimination of the effect of stochastic disturbance.

The developments not available for the GPC strategy involve description of the closed-loop system involving the LRP stochastic controller. In particular, the influence of controller parameters, P, C, and λ, on the location of the closed-loop system poles can be easily investigated. Furthermore, the variance of the output regulated by the LRP controller can be determined. The closed-loop system *stability criterion* is derived.

The LRP self tuner for scalar systems was found in simulation studies [150] to possess superior robustness properties against model underparameterization in comparison with self-tuning pole-placement controllers, and against variable system delay in comparison with the self-tuning GMV strategy.

Chapter 6

The utilization of additional outputs for feedback in self-tuning minimum prediction error control.

In this chapter the self-tuning minimum prediction error control of SIMO *deterministic* (section 6.1) and TIMO *stochastic* (section 6.2) systems having the feedback configuration FD (see page 18) is considered. Recall that the feedback configuration FD implies that there are some additional system outputs, apart of the controlled outputs, which can be utilized for feedback.

The utilization of additional system outputs for feedback have been recognised in many practical applications of *nonadaptive control.* For example, the use of supplementary signals, such as output power, machine speed and angle, in the excitation control of large a.c. turbogenerators has been found to improve markedly the damping of machine oscillations [166].

Self-tuning control was considered for, and found attractive in, application to power system stabilization during the past decade (see IEEE Transactions on Power Systems). However, self-tuning power system stabilisers, developed for a single-machine power system (which can be embedded in a multi-machine power system), utilize only one output variable for feedback (e.g., the rotor speed [167]). Recently, another approach was proposed as a compromise between the design of a self tuner which involves one feedback output and benefits resulting from utilization of additional system variables for feedback. Namely, the (single) feedback variable is formed by a weighted sum of the rotor speed and the terminal voltage [168]; however, the choice of the weighting coefficient in the sum is somewhat arbitrary and its influence on the system performance has not been investigated.

The above discussion reveals a need for the development of self-tuning controllers which utilize explicitly additional system outputs for feedback. The purpose of this chapter is to develop such self-tuning control strategies and to assess possible benefits resulting from this approach. The predictors introduced in sections 3.2 and 3.3 are employed for the development of self-tuning controllers. To the author's knowledge the approach to self-tuning control proposed in this chapter is original. The properties of the new strategies developed in this chapter are compared with the properties of conventional self tuners for systems having the feedback configuration FI.

6.1 Self-tuning minimum prediction error control of a single-input, multi-output deterministic system having the feedback configuration FD.

In this section:

- the MPE controller is developed for a deterministic SIMO system having the feedback configuration FD;
- the self-tuning MPE controller is introduced for a deterministic SIMO system having the feedback configuration FD;
- simulation studies demonstrate improvements in the performance of the self-tuning MPE controller within the feedback configuration FD in comparison with the corresponding self tuner within the feedback configuration FI.

6.1.1 A minimum prediction error controller in the feedback configuration FD for a system with known parameters.

Let us consider a SIMO system satisfying Assumptions 3.4 and 3.5 (p. 78). Consider the following quadratic single-stage cost function [9, chapter 5]

$$J(t+k) = \frac{1}{2}[y(t+k) - r(t+k)]^2 + \frac{\lambda}{2}\tilde{u}(t)^2, \tag{6.1}$$

where the control weighting coefficient $\lambda \geq 0$, k is system delay from the control input $u(t)$ to the controlled output $y(t)$, and the auxiliary control signal $\tilde{u}(t)$ is related to the control signal $u(t)$ via the following model

$$t(p)\tilde{u}(t) = s(p)u(t), \tag{6.2}$$

where

$$\begin{aligned} t(p) &= 1 + t_1 p + \cdots + t_{n_t} p^{n_t}, \quad n_t \geq 0 && (6.3) \\ s(p) &= 1 + s_1 p + \cdots + s_{n_s} p^{n_s}, \quad n_s \geq 0. && (6.4) \end{aligned}$$

The cost functions of the type of $J(t+k)$ have been commonly used for the development of self-tuning controllers for SISO systems [169,22,23,25,24,9], i.e., in such cases the feedback configuration FI has been assumed. A controller minimizing criterion 6.1 for a SISO system will be referred to as the MPE-SISO controller; such a controller is presented, for example, in [9, Theorem 5.2.3, p. 124]. In this section we shall develop the MPE controller minimizing performance criterion 6.1 for SIMO systems having the feedback configuration FD. The resulting controller will be referred to as the MPE-SIMO controller.

The k-step-ahead SIMO-type predictor introduced in subsection 3.2.1 (see Lemma 3.3, p. 79) is employed in the development of the MPE-SIMO control law. We have the following result.

Lemma 6.1 *For a SIMO system satisfying Assumptions 3.4 and 3.5 (p. 78) and for the bounded reference sequence* $\{r(t)\}$

(i) the control law minimizing cost function 6.1 is given by

$$u(t) = \begin{bmatrix} \rho & \rho\alpha(p) & \rho\overline{\beta}(p) + \lambda\overline{s}(p) & \lambda\overline{t}(p) \end{bmatrix} \begin{bmatrix} r(t+k) \\ -y_F(t) \\ -u(t-1) \\ \tilde{u}(t-1) \end{bmatrix}, \tag{6.5}$$

where the auxiliary control signal $\tilde{u}(t)$ is calculated from eqn. 6.2, and polynomials $\alpha(p)$ and $\beta(p)$, defined by eqns. 3.65 and 3.66 (p. 79), satisfy the Diophantine equation 3.67

$$p^{-k}N(p) = \begin{bmatrix} \alpha(p) & \beta(p) \end{bmatrix} \begin{bmatrix} N_F(p) \\ d(p) \end{bmatrix},$$

and

$$\begin{aligned} \rho &= \frac{\beta_0}{\beta_0^2 + \lambda}, & (6.6) \\ \overline{\beta}(p) &= p^{-1}\left[\beta(p) - \beta_0\right], & (6.7) \\ \overline{s}(p) &= p^{-1}\left[s(p) - 1\right], & (6.8) \\ \overline{t}(p) &= p^{-1}\left[t(p) - 1\right]; & (6.9) \end{aligned}$$

(ii) the closed-loop system resulting from the control law 6.5 is described by

$$w(p) \begin{bmatrix} u(t) \\ y(t+k) \\ y_F(t+k) \end{bmatrix} = t(p) \begin{bmatrix} d(p) \\ p^{-k}N(p) \\ p^{-k}N_F(p) \end{bmatrix} r(t+k), \tag{6.10}$$

where

$$w(p) = p^{-k}t(p)N(p) + \overline{\lambda}s(p)d(p) \tag{6.11}$$

and

$$\overline{\lambda} = \frac{\lambda}{\beta_0}; \tag{6.12}$$

(iii) the closed-loop system has bounded sequences of the control signal $\{u(t)\}$ and outputs $\{y(t)\}$ and $\{y_F(t)\}$ provided

$$w(z^{-1}) = z^k t(z^{-1})N(z^{-1}) + \overline{\lambda}s(z^{-1})d(z^{-1}) \neq 0 \quad \text{for} \quad |z| \geq 1. \tag{6.13}$$

Proof.

The proof of this lemma corresponds to the proof of [9, Theorem 5.2.3, p. 124]. The difference lies in the use of the SIMO-type predictor given by eqn. 3.68 (p. 80) instead of the predictor for a SISO system, and the system RDO representation 3.61 (p. 79) instead of the system DARMA model.

Q.E.D.

The control law 6.5 requires $\beta_0 \neq 0$; this is guaranteed in view of Comment 3.1 (p. 81).

The MPE-SISO control law for the corresponding system having the feedback configuration FI is given by eqn. 6.5 with $y(t)$ in place of $y_F(t)$ and with the polynomials $\alpha_s(p)$ and $\beta_s(p)$ of the SISO-type predictor 3.71 (p. 81) instead of $\alpha(p)$ and $\beta(p)$.

The stability condition 6.13 is the same for both the MPE-SISO and MPE-SIMO controllers. Furthermore, there always exists a choice of λ, $t(p)$ and $s(p)$ such that the closed-loop system is stable; the guidance for this choice is given in [23,25,24] [9, Remark 5.2.1, p. 125]. Moreover, the choice of $s(p)$ such that $s(1) = 0$ ensures zero steady-state error for the controlled output and constant reference sequence.

Comment 6.1 *In view of Comment 3.2 (p. 82), the degree of the polynomials $\alpha(p)$ and $\beta(p)$ of the MPE-SIMO controller may be smaller than that of the polynomials $\alpha_s(p)$ and $\beta_s(p)$ of the MPE-SISO controller. This shortens the controller memory and might lead to the reduction in the number of controller coefficients. Boths features of the MPE-SIMO control law become especially important in its application to self-tuning control. In such an application, faster convergence rate of parameter estimates can be expected for case FD than for FI due to the reduction in the number of estimated parameters (see subsection 6.1.3). Furthermore, more recent data is involved in the estimator due to a shorter memory of the controller for case FD. This is especially desirable in self-tuning control of time-varying systems.*

6.1.2 A self-tuning minimum prediction error controller in the feedback configuration FD.

In this subsection the direct self-tuning MPE strategies for a SIMO deterministic system having the feedback configuration FD are introduced. These strategies are adopted from the self-tuning MPE controllers developed for SISO systems having the feedback configuration FI in [9, chapter 6] (ST-MPE-SISO controllers).

Let us consider a system satisfying Assumptions 3.4 and 3.5 (p. 78). The MPE control law for a system with known parameters is given in Lemma 6.1. In order to introduce a self-tuning version of the control law 6.5 we make the following assumption.

Assumption 6.1 *(i) For the purpose of parameter estimation, the present value $y(t)$ of the controlled output is known;*

(ii) the time delay k from the control input $u(t)$ to the controlled output $y(t)$ is known;

(iii) an upper bound on the observability index ν_F is known;

(iv) $w(z^{-1}) \neq 0 \quad for \quad |z| \geq 1$;

(v) the sign of β_0 is known, $\operatorname{sign}\beta_0$.

Assumption 6.1 (i) is required for the calculation of equation error E.1 (Appendix E) of the parameter estimator employed in the self-tuning controller. Assumption 6.1 (ii) guarantees (in view of Comment 3.1, p. 81) that $\beta_0 \neq 0$; part (iii) is required to ensure the existence of polynomials $\alpha(p)$ and $\beta(p)$ (see Lemma 3.3, p. 79); part (iv) is required to ensure closed-loop system stability (see Lemma 6.1, part (iii)). Assumption 6.1 (v) is required to implement the constrained parameter estimation algorithm [9, chapter 3].

Self-tuning MPE controller (ST-MPE-SIMO strategy).

Recall that the SIMO-type k-step-ahead predictor can be written as (see eqns. 3.81 to 3.83, p. 86)

$$\begin{aligned} y(t) &= \Theta'\phi(t-k), && (6.14)\\ \Theta &= [\alpha\ \beta]', && (6.15)\\ \phi(t) &= \begin{bmatrix} y_F(t)' & \ldots & y_F(t-n_p)' & u(t) & \ldots & u(t-n_p) \end{bmatrix}'. && (6.16) \end{aligned}$$

Eqn. 6.14 defines a system model the parameters of which are estimated in the ST-MPE-SIMO strategy. Hence, denote the vector of parameter estimates of Θ resulting from a recursive parameter estimation algorithm by

$$\widehat{\Theta}(t) = \left[\widehat{\alpha}(t)\ \ \widehat{\beta}(t)\right]'. \tag{6.17}$$

It is also assumed that

Assumption 6.2 *A lower bound on the magnitude of β_0 is known, $|\beta_0|_{\min}$.*

The **ST-MPE-SIMO controller** applied to a SIMO system satisfying Assumptions 3.4 and 3.5 (p. 78), under conditions of Assumptions 6.1 and 6.2, consists of the following steps which are performed at every sample instant t:

step 1. calculate vector of estimates $\widehat{\Theta}(t)$ using, say, the RLS algorithm (see eqns. E.1 to E.4, Appendix E), with the equation error defined as

$$e(t) = y(t) - \tilde{\Theta}(t-1)'\phi(t-k); \tag{6.18}$$

step 2. set

$$\tilde{\Theta}(t) = \left[\tilde{\alpha}(t)\ \ \tilde{\beta}(t)\right]' = I_{(n_p+1)f,n_p}\widehat{\Theta}(t) + i_{(n_p+1)f+1}\tilde{\beta}_0(t), \tag{6.19}$$

where the $(n_p+1)(f+1) \times (n_p+1)(f+1)$ matrix $I_{(n_p+1)f,n_p}$ is defined as

$$I_{(n_p+1)f,n_p} = \begin{bmatrix} I_{(n_p+1)f} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I_{n_p} \end{bmatrix},$$

and the $(n_p+1)(f+1) \times 1$ vector $i_{(n_p+1)f+1}$ is the $((n_p+1)f+1)$-th column of the identity matrix of order $(n_p+1)(f+1)$, and

$$\tilde{\beta}_0(t) = \begin{cases} i'_{(n_p+1)f+1}\widehat{\Theta}(t) & \text{if } \; i'_{(n_p+1)f+1}\widehat{\Theta}(t)\,(\operatorname{sign}\beta_0) \geq |\beta_0|_{\min} \\ |\beta_0|_{\min}\,(\operatorname{sign}\beta_0) & \text{if } \; i'_{(n_p+1)f+1}\widehat{\Theta}(t)\,(\operatorname{sign}\beta_0) < |\beta_0|_{\min} \end{cases} \tag{6.20}$$

step 3. calculate the control signal

$$u(t) = \left[\; \tilde{\rho}(t) \quad \tilde{\rho}(t)\tilde{\alpha}(t,p) \quad \tilde{\rho}(t)\overline{\tilde{\beta}}(t,p) + \lambda\overline{s}(p) \quad \lambda\overline{t}(p) \;\right] \begin{bmatrix} r(t+k) \\ -y_F(t) \\ -u(t-1) \\ \tilde{u}(t-1) \end{bmatrix}, \tag{6.21}$$

where

$$\begin{aligned} \tilde{\rho}(t) &= \frac{\tilde{\beta}_0(t)}{\tilde{\beta}_0(t)^2 + \lambda}, \\ \overline{\tilde{\beta}}(t,p) &= p^{-1}\left[\tilde{\beta}(t,p) - \tilde{\beta}_0(t)\right], \end{aligned}$$

and $\tilde{\alpha}(t,p) = \tilde{\alpha}(t)P_{n_p}$, and $\tilde{\beta}(t,p) = \tilde{\beta}(t)P_{n_p}$, where $P_{n_p} = [1 \; p \; \ldots \; p^{n_p}]'$.

Step 2 of the ST-MPE-SIMO strategy eliminates the possibility that the *estimate* of the coefficient β_0 will become zero; for this purpose Assumption 6.2 is needed.

The above ST-MPE-SIMO strategy is a version for SIMO systems of the self-tuning MPE controller developed for SISO systems in [9, Remark 6.3.4, p. 189]. The self-tuning controller for the system having the feedback configuration FI (ST-MPE-SISO strategy) results from the ST-MPE-SIMO scheme by replacing $y_F(t)$ by $y(t)$, and α and β by α_s and β_s of the SISO-type predictor.

Note that the ST-MPE-SIMO strategy estimates parameters of the model given by eqn. 6.14 and not the parameters of the MPE-SIMO control law 6.5. Consequently, the implementation of the ST-MPE-SIMO algorithm requires further calculations on the parameter estimates (step 3), and the control law 6.21 is nonlinear in parameter estimates $\tilde{\Theta}(t)$. It is possible, however, to rearrange the self-tuning algorithm implementing the MPE-SIMO strategy so that parameters of the control law are estimated directly. For this purpose we employ the approach proposed in [9, pp. 192-194] for

MPE self tuners for SISO systems. Since the resulting control law is *linear* in estimated parameters, this self-tuning scheme will be referred to as the ST-MPE-L-SIMO strategy.

Self-tuning MPE controller (ST-MPE-L-SIMO strategy).

Let us reparameterize the SIMO-type k-step-ahead predictor given in Lemma 3.3 (p. 79) as follows. Multiplying both sides of eqn. 3.68 by ρ and then adding $\rho\overline{\lambda}u(t)$ to both sides leads to the model

$$u(t-k) = \Theta'\phi(t-k), \tag{6.22}$$

where ρ and $\overline{\lambda}$ are defined by eqns. 6.6 and 6.12, and

$$\begin{aligned}
\Theta &= \left[\rho \;\; \alpha_\rho \;\; \overline{\beta}_\rho\right]', && (6.23)\\
\alpha_\rho &= \left[\rho\alpha_0 \;\; \ldots \;\; \rho\alpha_{n_p}\right], && (6.24)\\
\overline{\beta}_\rho &= \left[\rho\beta_1 \;\; \ldots \;\; \rho\beta_{n_p}\right], && (6.25)\\
\phi(t) &= \left[y(t+k) + \overline{\lambda}u(t);\right. \\
&\quad -y_F(t)' \;\; \ldots \; - y_F(t-n_p)'; \;\; -u(t-1) \;\; \ldots \; - u(t-n_p)\,]'. && (6.26)
\end{aligned}$$

Eqn. 6.22 describes a system model the parameters of which are estimated in the ST-MPE-L-SIMO strategy. Hence, denote the $(n_p+1)(f+1)\times 1$ vector of parameter estimates of Θ resulting from a recursive parameter estimation algorithm by

$$\widehat{\Theta}(t) = \left[\widehat{\rho}(t) \;\; \widehat{\alpha}_\rho(t) \;\; \widehat{\overline{\beta}}_\rho(t)\right]'. \tag{6.27}$$

The control law 6.5 can be written as

$$u^o(t) = \Theta'\overline{\phi}(t), \tag{6.28}$$

where the $(n_p+1)(f+1)\times 1$ regression vector $\overline{\phi}(t)$ is

$$\begin{aligned}
\overline{\phi}(t) &= \left[r(t+k) + \overline{\lambda}\left(\overline{t}(p)\tilde{u}(t-1) - \overline{s}(p)u(t-1)\right);\right. \\
&\quad -y_F(t)' \;\; \ldots \; - y_F(t-n_p)'; \;\; -u(t-1) \;\; \ldots \; - u(t-n_p)\,]'. && (6.29)
\end{aligned}$$

It is also assumed that

Assumption 6.3 *A lower bound on the magnitude of ρ is known, $|\rho|_{\min}$, and the scalar $\overline{\lambda}$ is given, such that* $\operatorname{sign}\overline{\lambda} = \operatorname{sign}\beta_0$.

The above assumption implies that $\overline{\lambda}$ is specified for the ST-MPE-L-SIMO strategy, which means that the cost function 6.1 is minimized with $\lambda = \overline{\lambda}\beta_0$ (see eqn. 6.12). It is required that $\operatorname{sign}\overline{\lambda} = \operatorname{sign}\beta_0$, because $\lambda \geq 0$ [9, Remark 6.3.7, p. 199].

The **ST-MPE-L-SIMO controller** applied to a SIMO system satisfying Assumptions 3.4 and 3.5 (p. 78), under conditions of Assumptions 6.1 and 6.3, consists of the following steps which are performed at every sample instant t:

step 1. calculate vector of estimates $\widehat{\Theta}(t)$ using, say, the RLS algorithm (see eqns. E.1 to E.4, Appendix E), with the equation error defined as

$$e(t) = u(t-k) - \tilde{\Theta}(t-1)'\phi(t-k); \tag{6.30}$$

step 2. set

$$\tilde{\Theta}(t) = I_{0,(f+1)n_p+f}\widehat{\Theta}(t) + i_1\tilde{\rho}(t), \tag{6.31}$$

where the $(n_p+1)(f+1) \times (n_p+1)(f+1)$ matrix $I_{0,(f+1)n_p+f}$ is defined as

$$I_{0,(f+1)n_p+f} = \begin{bmatrix} 0 & 0 \\ 0 & I_{(n_p+1)f+f} \end{bmatrix},$$

and the $(n_p+1)(f+1) \times 1$ vector i_1 is the first column of the identity matrix of order $(n_p+1)(f+1)$, and

$$\tilde{\rho}(t) = \begin{cases} i_1'\widehat{\Theta}(t) & \text{if } i_1'\widehat{\Theta}(t)\,(\operatorname{sign}\beta_0) \geq |\rho|_{\min} \\ |\rho|_{\min}\,(\operatorname{sign}\beta_0) & \text{if } i_1'\widehat{\Theta}(t)\,(\operatorname{sign}\beta_0) < |\rho|_{\min} \end{cases} \tag{6.32}$$

step 3. calculate the control signal

$$u(t) = \tilde{\Theta}(t)'\overline{\phi}(t). \tag{6.33}$$

The above ST-MV-L-SIMO strategy is a version for SIMO systems of the self-tuning MPE controller developed for SISO systems in [9, pp. 192-196].

Step 2 of the ST-MPE-L-SIMO strategy eliminates the possibility that the estimate of the parameter ρ will become zero; for this purpose Assumption 6.3 is needed.

Note that the ST-MPE-L-SIMO controller is a *direct* strategy, i.e., parameters of the control law are estimated. Furthermore, the control law 6.33 is linear in $\tilde{\Theta}(t)$.

The *global convergence* of the ST-MPE-L-SIMO strategy is established below using the technique introduced for the self-tuning MPE controllers for SISO systems in [9, chapter 6].

Lemma 6.2 *Assume that future value of the bounded reference sequence $r(t+k)$ is known (or is computable) at time t. For a system satisfying Assumptions 3.4 and 3.5 (p. 78), and subject to Assumptions 6.1 and 6.3, the ST-MPE-L-SIMO strategy (implemented with the RLS algorithm with no forgetting) yields*

(a) bounded sequences $\{u(t)\}$, $\{y(t)\}$, and $\{y_F(t)\}$;

(b)

$$\lim_{t\to\infty} [u(t) - u^o(t)] = 0, \tag{6.34}$$

where $u^o(t)$ is the control signal minimizing cost function $J(t+k)$ given by eqn. 6.1 with $\lambda = \overline{\lambda}\beta_0$.

Proof: see Appendix F.

The above lemma shows that the control signal generated by the ST-MPE-L-SIMO strategy converges to the control signal which would be produced by the control law 6.5 for a system with known parameters and with $\lambda = \overline{\lambda}\beta_0$. No persistent excitation condition is required for the reference sequence $\{r(t)\}$. However, nothing is said about convergence of estimates of parameters of the controller to their true values.

6.1.3 Performance improvements in self-tuning control of deterministic systems by utilization of additional outputs for feedback: simulation studies of minimum prediction error control of a robot arm.

In this subsection, simulation studies demonstrate properties of the ST-MPE-SIMO strategy applied to a hydraulic industrial manipulator considered in examples 3.2.1 (p. 83) and 3.2.2 (p. 88).

In the simulation studies described in this subsection the ST-MPE-SIMO strategy was implemented *without* the test on the lower bound of the absolute value of estimate $\widehat{\beta}_0(t)$, i.e., step 2 (see eqns. 6.19 and 6.20) is omitted [1] and thus $\tilde{\Theta}(t) = \widehat{\Theta}(t)$ (see eqns. 6.18 and 6.21). A number of simulation studies suggests that the situation in which $\widehat{\beta}_0(t) = 0$ is unlikely to occur in practice.

The RLS estimator employed in both the ST-MPE-SIMO and ST-MPE-SISO strategies was implemented with the U-D factorization method [164] (see also discussion on page 174).

Example 6.1.1.

The purpose of this example is to demonstrate the improvement in the performance of the self-tuning MPE controller resulting from utilization of additional system outputs for feedback.

Let us consider the model of a single joint of the manipulator POLAR 6000 described in examples 3.2.1 and 3.2.2 (see eqn. 3.72, p. 83). The task is to control the manipulator joint $y(t) = \theta(t) = y_1(t) = Cx(t)$ along the X-axis [130].

For the purpose of implementation of the ST-MPE-SIMO strategy it is assumed that there are three outputs available for feedback, i.e., $y_F(t) = [y_1(t)\ y_2(t)\ y_3(t)]' =$

[1] Recall that step 2 was introduced in the ST-MPE-SIMO strategy to prevent the estimate $\widehat{\beta}_0(t)$ from becoming zero.

$C_F x(t)$ and

$$C_F = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix},$$

where $c_{31} = c_{13}$, $c_{32} = c_{23}$ and $c_{33} = 0.0051275$. The observability index is $\nu_F = 1$ and $n_p = \nu_F - 1 = 0$ is chosen. The coefficients of the polynomials $\alpha(p)$ and $\beta(p)$, which are estimated in the ST-MPE-SIMO strategy, are found from eqn. 3.69 (cf. example 3.2.1) as

$$\begin{bmatrix} \alpha_{10} & \alpha_{20} & \alpha_{30} & \beta_0 \end{bmatrix} = \begin{bmatrix} 2.7124 & 1 & 0 & 0.0008128 \end{bmatrix}.$$

(Since $\alpha_{30} = 0$ the MPE-SIMO scheme can be implemented without feedback from the third output (cf. examples 3.2.1 and 3.2.2) but the more general case is considered for the self-tuning control.) Note that one of the roots of the polynomial $N(z^{-1})$ is $z = -3.6324$, therefore $\lambda \neq 0$ is required to satisfy the stability condition 6.13. The following set of parameters of the MPE-SIMO control law was chosen: $t(p) = 1 - 1.6p + 0.7p^2$, $s(p) = 1 + 0.2p$, and $\lambda = 2 \times 10^{-6}$. The problem of λ-offset is eliminated due to the process integrating properties ($d(1) = 0$). The control signal magnitude limit was set $|u(t)| \leq 30$ [mA] according to the specification of the maximum excitation current in the servovalve of the hydraulic motor [130].

Alternatively, one can assume that the only output which is available for feedback is the output which is to be controlled, i.e., the system has the feedback configuration FI. The ST-MPE-SISO controller, minimizing cost function 6.1 for a system having the feedback configuration FI, was simulated with the same parameters of the control law $t(p)$, $s(p)$, λ, and the limit $|u(t)|$. The coefficients of the polynomials $\alpha_s(p)$ and $\beta_s(p)$ which are estimated by this strategy are given by eqns. 3.74 and 3.75 (p. 84), i.e., 6 parameters are estimated in comparison with 4 parameters for the ST-MPE-SIMO scheme.

For both ST-MPE-SIMO and ST-MPE-SISO strategies, the RLS algorithm was initialized with $P(-1) = 10I$, and all parameter estimates were zero except for $\widehat{\beta}_0(0) = \widehat{\beta}_{s0}(0) = 1$; a constant forgetting factor $\alpha(t) = 0.97$ was used.

The following cumulative loss (CL) function

$$\mathrm{CL}(t) = \sum_{\tau=1}^{t} [y(\tau) - r(\tau)]^2 ,$$

was calculated for $t = 1, \ldots, 120$ to compare the performance of the closed-loop systems resulting from both strategies.

The input-output behaviour of the closed-loop system and the cumulative loss function for both strategies are depicted in fig. 6.1.

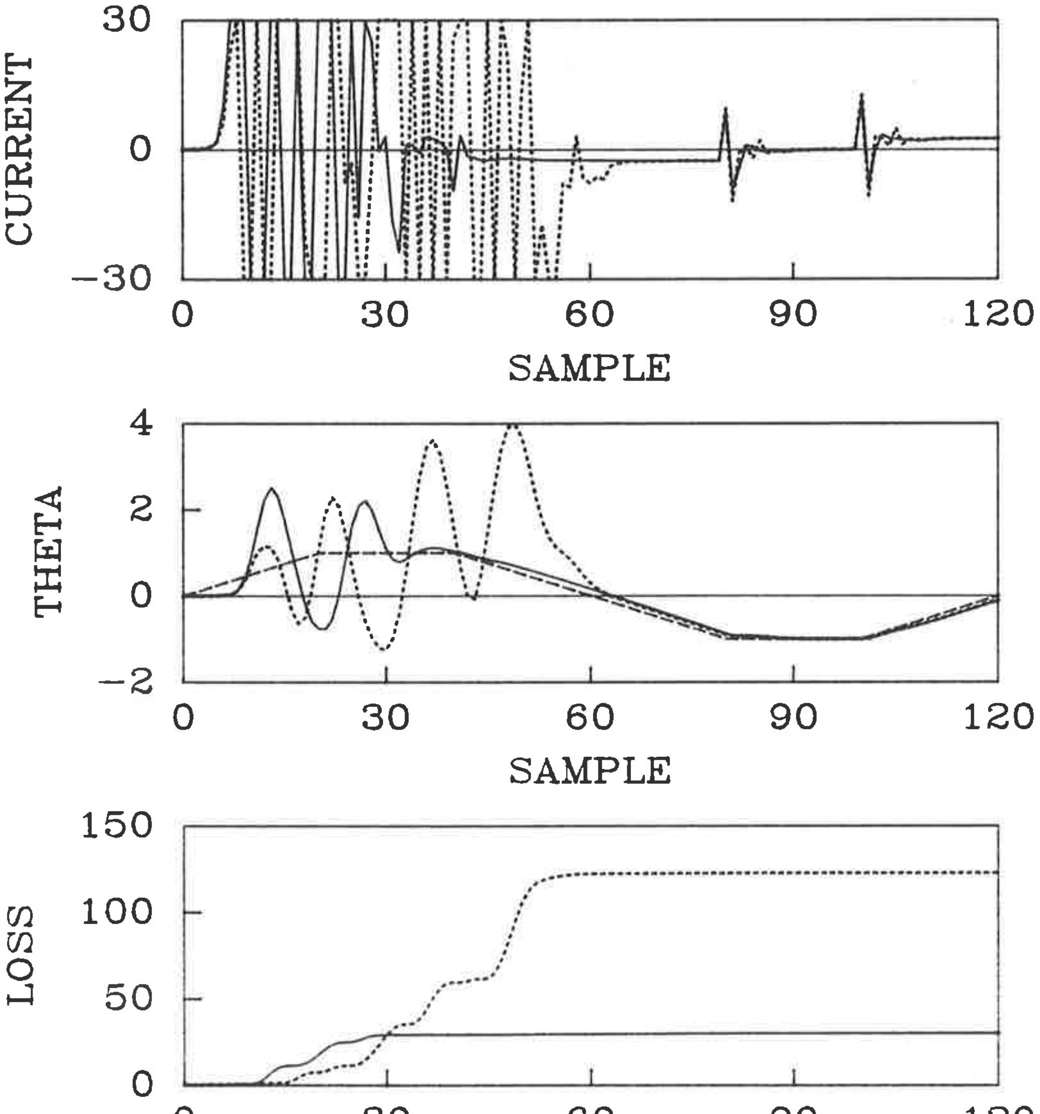

Figure 6.1: The input (current [mA]), output (θ [rad]) with the reference sequence (broken line) and the cumulative loss for the ST-MPE-SIMO controller (solid lines) and ST-MPE-SISO controller (dotted lines) applied to the robot arm.

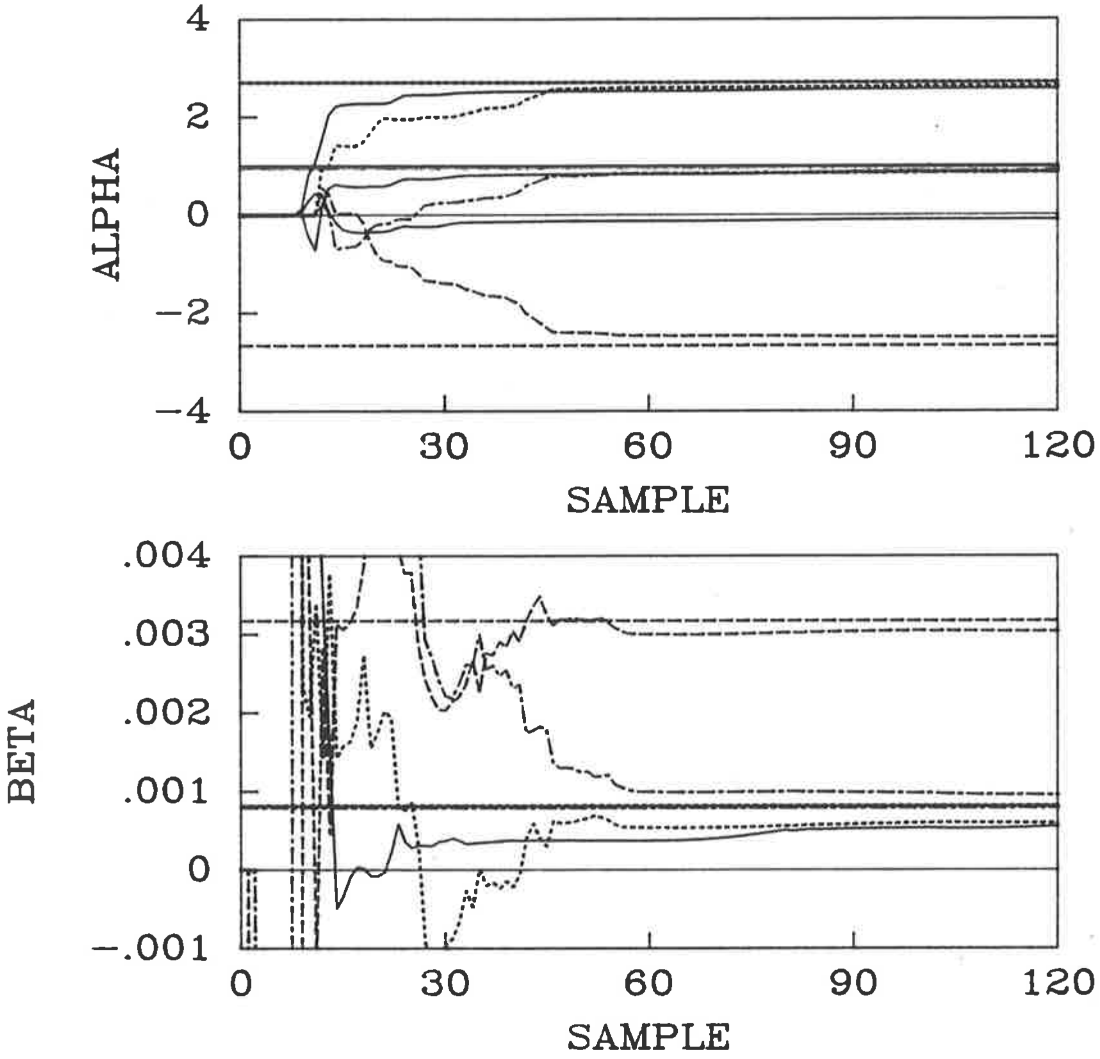

Figure 6.2: The estimates of the coefficients of the polynomials $\alpha(p)$ and $\beta(p)$ of the ST-MPE-SIMO controller (solid lines) and the polynomials $\alpha_s(p)$ and $\beta_s(p)$ of the ST-MPE-SISO controller (broken lines) applied to the robot arm.

It can be seen that the ST-MPE-SISO strategy produced a very oscillatory control sequence, which resulted in poor output performance and increase of the loss function.

The differences in the performance of both self-tuning strategies (which aim at the same objective) are due to the differences in the convergence rate of the parameter estimates. The benefit of estimating fewer parameters is clearly visible in this example. The convergence rate of the estimates of the ST-MPE-SIMO strategy was faster than that of the ST-MPE-SISO scheme (see fig. 6.2). Note that faster convergence rate is observed for the estimates of the ST-MPE-SIMO strategy despite the fact that the undesirable input-output behaviour associated with the ST-MPE-SISO controller provided "rich" data for the estimator in comparison with relatively less exciting input produced by the SIMO-type self tuner. The estimates of $\alpha(p)$ and $\beta(p)$ converged close to their true values after approximately 25 samples; for $\dot{\alpha}_s(p)$ and $\beta_s(p)$, 45 iterations were required.

It is interesting to note that the SIMO approach led to a compromise between a desirable input-output system performance (i.e., the achievement of the *control objective*) and the requirement of sufficient excitation for the purpose of *estimation*. The two goals are conflicting and form a dilemma in self-tuning control which is sometimes solved by some *ad hoc* procedures (e.g., supervision of adaptation phase and parameter estimation, input probing in open loop [170,10], etc.) or analytically, leading to dual adaptive control (see chapter 1). The self-tuning strategy for a SIMO system was capable of ensuring fast convergence due to the reduction in the number of estimated parameters while providing good input-output performance.

A number of simulation studies confirmed the superiority of the SIMO based self tuners over the usual SISO-type design in terms of the performance of the self-tuning control system.

Example 6.1.2.

In example 6.1.1 the improvement in performance of the self-tuning controller applied to the robot manipulator was due to the reduction in the number of estimated parameters for the feedback configuration FD in comparison with FI. In the following example the self-tuning controllers for both feedback configurations FD and FI involve the same number of estimated parameters. However, the benefits resulting from the separation of the feedback and controlled outputs for a system having the feedback configuration FD are illustrated in the application of self-tuning control to a robot manipulator.

The feedback configuration FD allows us to select system outputs for feedback. In the following example the controlled output $y(t)$ is not selected as a feedback variable. The measurement of $y(t)$ is required, however, for the purpose of estimation (see Assumption 6.1 (i), p. 200). On the other hand, if the system has the feedback configuration FI then the controlled output is the feedback variable.

Let us consider the model of the single joint of the POLAR 6000 robot given by eqn. 3.72, p. 83. The controlled output is $y(t) = y_1(t) = Cx(t)$, where $C = [c_{11}\ c_{12}\ c_{13}]$. It is assumed that the measurement of the controlled output $y(t)$ is contaminated with a zero mean white noise $\omega(t)$, so that the measured signal $y_m(t)$ (observation of $y(t)$) is given by

$$y_m(t) = y(t) + \omega(t). \tag{6.35}$$

Furthermore, assume that the feedback output is now given by $y_F(t) = y_2(t) = C_F x(t)$, where $C_F = [c_{21}\ \ c_{22}\ \ c_{23}]$; then the observability index $\nu_F = 3$. Hence, a complete separation of the output which is to be controlled and the feedback output is assumed. The corresponding output measurement configuration for the implementation of the ST-MPE-SIMO strategy is depicted in fig. 6.3.

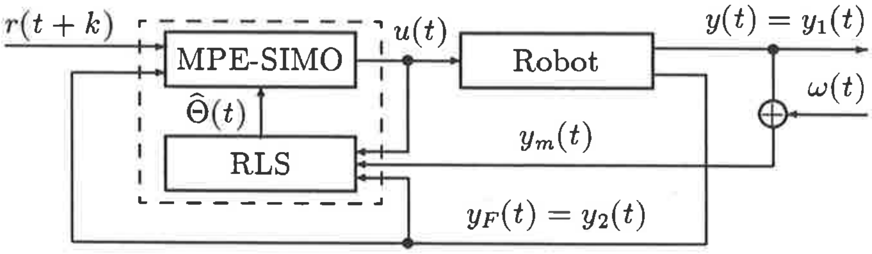

Figure 6.3: The output measurement configuration for the ST-MPE-SIMO strategy.

For the purpose of comparison let us now assume that the robot manipulator is in the feedback configuration FI, i.e., the controlled output is the only output available for feedback. In this case $C_F = C = [c_{11}\ \ c_{12}\ \ c_{13}]$. The corresponding output measurement configuration for the implementation of the ST-MPE-SISO strategy is depicted in fig. 6.4.

Let us now compare models the parameters of which are estimated by the RLS method in the implementation of the ST-MPE-SIMO and ST-MPE-SISO schemes.

The ST-MPE-SIMO strategy involves estimation of parameters of the model given by eqns. 6.14 to 6.16 with the equation error defined by eqn. 6.18 (p. 201). It follows

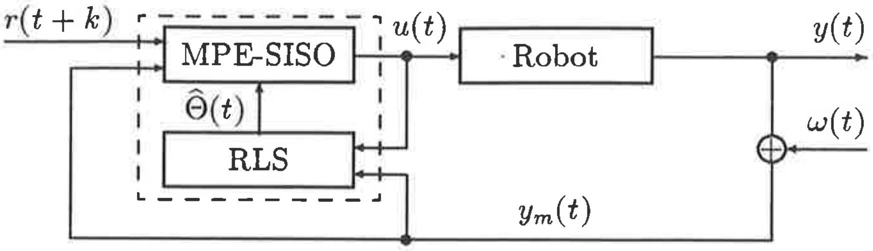

Figure 6.4: The output measurement configuration for the ST-MPE-SISO strategy.

form eqn. 6.35 that the model becomes now

$$y_m(t) = \Theta'\phi(t-k) + \omega(t). \tag{6.36}$$

The corresponding equation error is defined as

$$e(t) = y_m(t) - \widehat{\Theta}(t-1)'\phi(t-k),$$

where the parameter estimate vector $\widehat{\Theta}(t)$ is defined by eqn. 6.17. Since the noise term $\omega(t)$ is uncorrelated with data of the regression vector $\phi(t)$ given by eqn. 6.16 in the model 6.36, the RLS method yields unbiased parameter estimates [160, p. 370].

On the other hand, it follows from eqns. 3.71 (p. 81) and 6.35 that the model involved in the ST-MPE-SISO strategy is given by

$$y_m(t) = \Theta_s'\phi_s(t-k) + n(t), \tag{6.37}$$

where the noise term is $n(t) = \left[1 - p^k\alpha_s(p)\right]\omega(t)$, and the vector of coefficients of polynomials of the SISO-type predictor 3.71 (p. 81) is $\Theta_s = [\alpha_s \;\; \beta_s]'$, and the regression vector is

$$\phi_s(t) = \left[\begin{array}{cccccc} y_m(t) & \ldots & y_m(t-n+1) & u(t) & \ldots & u(t-n+1) \end{array}\right]'.$$

Note that for the SISO case, the noise term $n(t)$ is no longer white noise and the

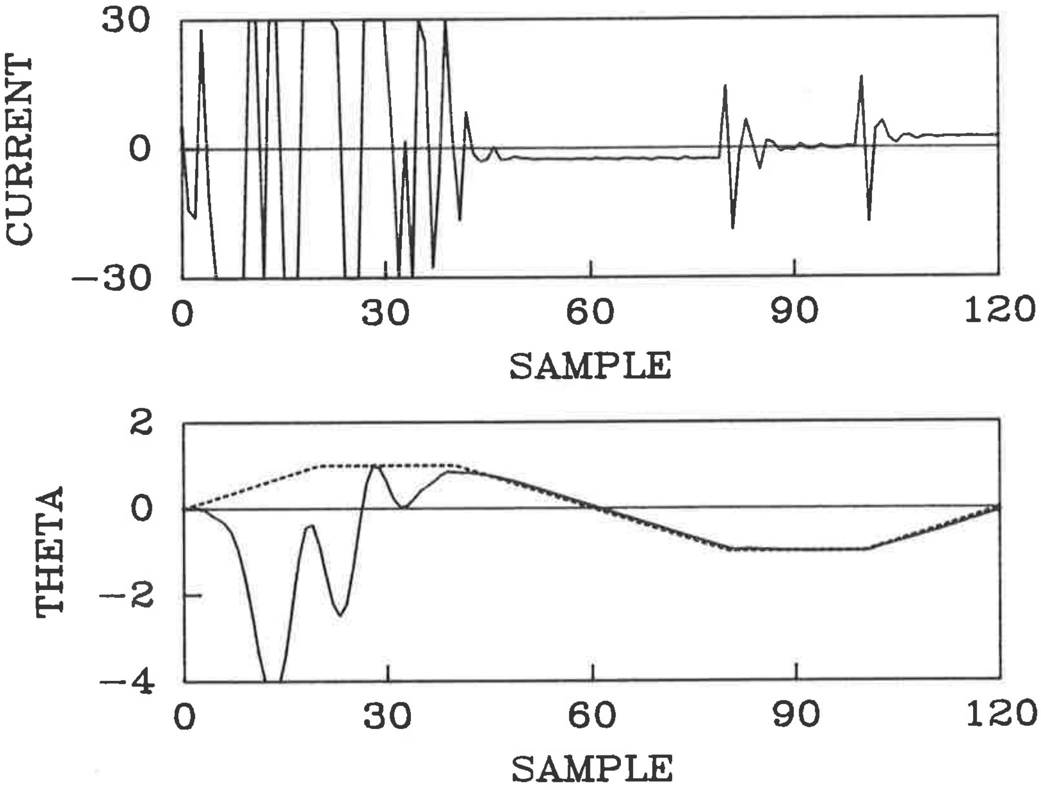

Figure 6.5: The input (current [mA]) and output (θ [rad]) resulting from the ST-MPE-SIMO controller applied to the robot arm with the controlled output measurement noise.

RLS algorithm yields biased estimates [2] [160]. The corresponding equation error is

$$e(t) = y_m(t) - \widehat{\Theta}_s(t-1)'\phi_s(t-k).$$

True value of Θ_s estimated by the ST-MPE-SISO strategy is given by eqns. 3.74 and 3.75 (p. 84). For the ST-MPE-SIMO controller $n_p = \nu_F - 1 = 2$ and the solution to eqn. 3.69 yields

$$\begin{aligned} \alpha &= [\alpha_{20}\ \alpha_{21}\ \alpha_{22}] = [-2.1809\ \ 2.7124\ \ -1.1155]\,, \\ \beta &= [\beta_0\ \beta_1\ \beta_2] = [0.0008128\ \ 0.0122906\ \ 0.0016147]\,, \end{aligned}$$

[2] It should be noted that some other recursive parameter estimation methods (e.g., RELS and RML, see Appendix E) can be applied to yield unbiased estimates of the model 6.37. It is known, however, that the convergence rate of parameter estimates resulting form such estimators is slow in comparison with that of the RLS method (when unbiased). Furthermore, some other conditions are required to ensure convergence (we shall return to the corresponding problems in subsection 6.2.2).

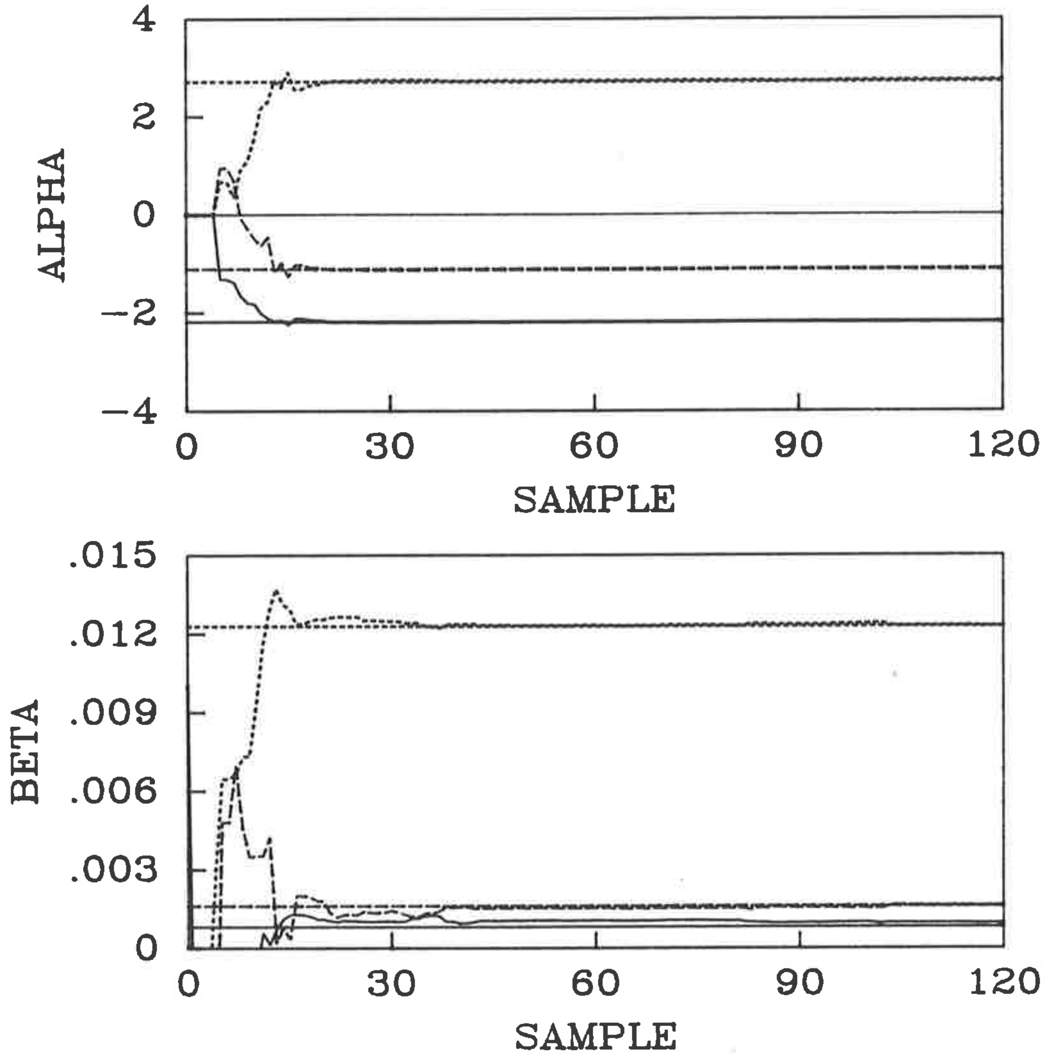

Figure 6.6: The estimates of the coefficients of the polynomials $\alpha(p)$ and $\beta(p)$ of the ST-MPE-SIMO controller applied to the robot arm with the controlled output measurement noise.

i.e., both strategies involve estimation of the same number of parameters.

For both ST-MPE-SIMO and ST-MPE-SISO strategies, the parameters of the control law (i.e., $t(p)$, $s(p)$, λ, $|u(t)|$) were chosen as in example 6.1.1. The RLS algorithm was initialized with $P(-1) = 10^3 I$, and all parameter estimates were zero except for $\widehat{\beta}_0(0) = \widehat{\beta}_{s0}(0) = 0.01$; no forgetting was used. The dispersion of the (gaussian) white noise $\omega(t)$ in eqn. 6.35 was chosen $\sigma = 0.05$.

The input-output behaviour of the closed-loop system resulting from the use of the ST-MPE-SIMO strategy is depicted in fig. 6.5 (p. 213). It may be observed that after the initial phase of tuning the control signal achieved its desirable form, and

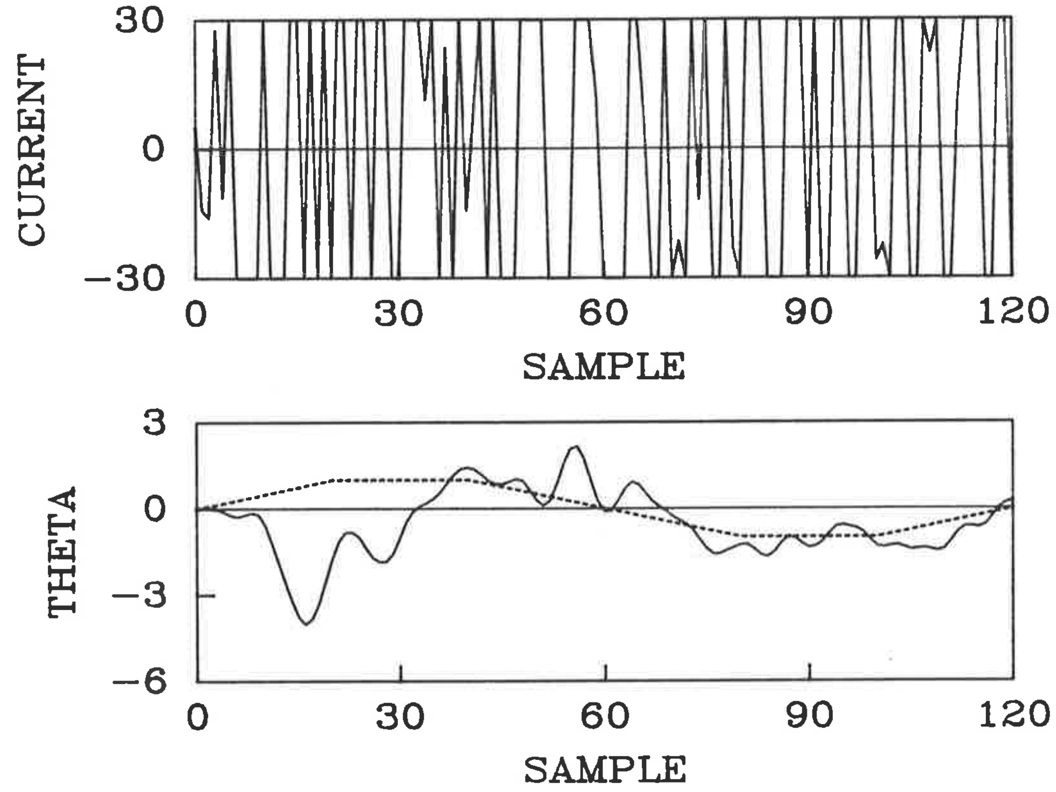

Figure 6.7: The input (current [mA]) and output (θ [rad]) resulting from the ST-MPE-SISO controller applied to the robot arm with the controlled output measurement noise.

the controlled output followed the reference sequence closely after a few oscillations. The estimates of the coefficients of the polynomials $\alpha(p)$ and $\beta(p)$ converged close to their true values after about 15 iterations (see fig. 6.6, p. 214).

Next the ST-MPE-SISO strategy was simulated. The input-output behaviour of the closed-loop system is shown in fig. 6.7. The control signal did not converge to its desirable form and, in fact, was saved only by the limit on its magnitude. The reason is that the estimator yields biased parameter estimates due to the measurement noise (see fig. 6.8, p. 216).

This example demonstrated possible benefits resulting from the separation of the controlled output from the feedback outputs for the system having the feedback configuration FD and in the presence of the measurement noise on the controlled output.

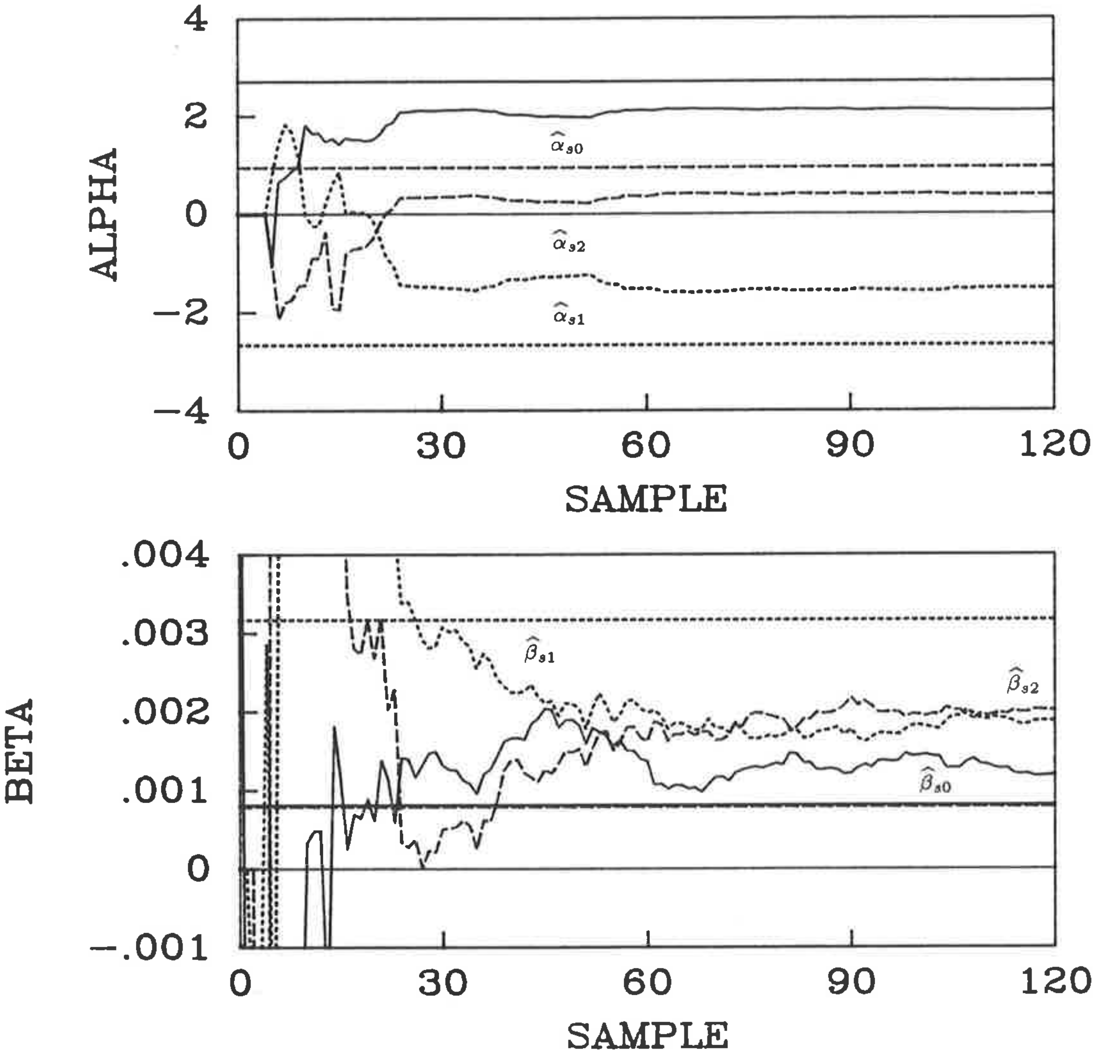

Figure 6.8: The estimates of the coefficients of the polynomials $\alpha_s(p)$ and $\beta_s(p)$ of the ST-MPE-SISO controller applied to the robot arm with the controlled output measurement noise.

6.2 Self-tuning minimum prediction error control of a two-input, multi-output stochastic system having the feedback configuration FD.

In this section:

- the weighted minimum variance (WMV) controller is derived for a stochastic TIMO system having the feedback configuration FD;
- the self-tuning minimum variance (MV) controller is introduced for a stochastic TIMO system having the feedback configuration FD (ST-MV-TIMO strategy);
- global convergence of the ST-MV-TIMO strategy is established;
- properties of the ST-MV-TIMO strategy are discussed;
- a survey of stochastic self-tuning controllers for systems having the feedback configuration FI is presented with respect to the methods of convergence analysis and required system assumptions (details are presented in Appendix G);
- simulation studies demonstrate improvements in the performance of the self-tuning MV controller within the feedback configuration FD in comparison with the corresponding self tuner within the feedback configuration FI.

6.2.1 A weighted minimum variance controller in the feedback configuration FD for a system with known parameters.

Let us consider a TIMO system satisfying Assumptions 3.7 to 3.10 (p. 91). The control law developed in this subsection minimizes the mean-square error between the controlled output $y(t+k)$ and its desired value $r(t+k)$ with possible trade-off between the minimized error and required control effort.

Consider the following quadratic single-stage cost function [83, p. 173] [160, p. 249] [9, p. 411] [59, p. 285]

$$J(t+k) = E\left\{\frac{1}{2}[y(t+k) - r(t+k)]^2 + \frac{\lambda}{2}\tilde{u}(t)^2|\mathcal{F}_t\right\}, \tag{6.38}$$

where the control weighting coefficient $\lambda \geq 0$, k is system delay from the control input $u(t)$ to the controlled output $y(t)$, and the auxiliary control signal $\tilde{u}(t)$ is defined by eqns. 6.2 to 6.4 (p. 197).

The cost functions of the type of $J(t+k)$ have been commonly used to develop self-tuning stochastic controllers for TISO systems [169,22,23,25,24,9,59], i.e., in such cases the feedback configuration FI has been assumed. A controller minimizing criterion 6.38 for a TISO system will be referred to as the WMV-TISO controller; such a controller is developed for instance in [9, Theorem 10.3.4, p. 418] [59, Theorem 12.4, p. 301]. In this section, we shall consider the WMV controller minimizing performance criterion 6.38 for TIMO systems having the feedback configuration FD. The resulting controller will be referred to as the WMV-TIMO controller.

The development of the WMV-TIMO controller is based on the optimal k-step-ahead TIMO-type predictor introduced in subsection 3.3.1 (see Lemma 3.5 on page 94 for $k_p = k$). We have the following result.

Theorem 6.1 *For a TIMO system satisfying Assumptions 3.7 to 3.10 (p. 91), 3.11 (p. 113), and for the bounded reference sequence $\{r(t)\}$*

(i) the control law minimizing cost function 6.38 is given by

$$u(t) = \begin{bmatrix} \rho & \rho\alpha_\omega(p) & \rho\overline{\beta}_\omega(p) + \lambda\overline{s}(p) & \lambda\overline{t}(p) \end{bmatrix} \begin{bmatrix} r(t+k) \\ -y_F(t) \\ -u(t-1) \\ \tilde{u}(t-1) \end{bmatrix}, \qquad (6.39)$$

where the auxiliary control signal $\tilde{u}(t)$ is calculated from eqn. 6.2; polynomials $\overline{s}(p)$ and $\overline{t}(p)$ are defined by eqns. 6.8 and 6.9; polynomials $\alpha_\omega(p)$ and $\beta_\omega(p)$, which are defined by eqns. 3.97 and 3.98 (p. 94), satisfy the polynomial equation 3.99 for $k_p = k$

$$p^{-k}\begin{bmatrix} N(p) & c_\omega(p) \end{bmatrix} = \begin{bmatrix} \alpha_\omega(p) & \beta_\omega(p) & F(p) \end{bmatrix} \begin{bmatrix} N_F(p) & C_\omega(p) \\ d(p) & 0 \\ 0 & p^{-k}d(p) \end{bmatrix}, \qquad (6.40)$$

and

$$\rho = \frac{\beta_0}{\beta_0^2 + \lambda}, \qquad (6.41)$$

$$\overline{\beta}_\omega(p) = p^{-1}\left[\beta_\omega(p) - \beta_0\right]; \qquad (6.42)$$

(ii) the resulting closed-loop system is described by

$$w(p)\begin{bmatrix} u(t) \\ y(t+k) \\ d(p)y_F(t+k) \end{bmatrix}$$
$$= \begin{bmatrix} t(p)d(p) & -p^k t(p)\alpha_\omega(p)C_\omega(p) \\ p^{-k}t(p)N(p) & p^{-k}t(p)N(p)F(p)+\overline{\lambda}s(p)c_\omega(p) \\ p^{-k}t(p)d(p)N_F(p) & w(p)C_\omega(p)-t(p)N_F(p)\alpha_\omega(p)C_\omega(p) \end{bmatrix}\begin{bmatrix} r(t+k) \\ \omega(t+k) \end{bmatrix} \tag{6.43}$$

where

$$w(p) = p^{-k}t(p)N(p) + \overline{\lambda}s(p)d(p), \tag{6.44}$$

and $\overline{\lambda} = \frac{\lambda}{\beta_0}$*;*

(iii) the closed-loop system sequences $\{u(t)\}$ *and* $\{y(t)\}$ *are sample mean-square bounded almost surely if*

$$z^k t(z^{-1})N(z^{-1}) + \overline{\lambda}s(z^{-1})d(z^{-1}) \neq 0 \quad \text{for} \quad |z| \geq 1; \tag{6.45}$$

furthermore, the closed-loop system sequence $\{y_F(t)\}$ *is sample mean-square bounded almost surely if condition 6.45 is satisfied and*

$$d(z^{-1}) \neq 0 \quad \text{for} \quad |z| \geq 1. \tag{6.46}$$

Proof.

Part (i). The optimality of the control law 6.39 can be shown following the derivation of the WMV controller for TISO systems (see e.g., [9, chapter 10]) using the k-step-ahead TIMO-type predictor 3.101 (p. 95).

Part (ii).

The control law 6.39 can be written using eqns. 3.101, 3.102 and 6.2 as

$$y(t+k) + \overline{\lambda}\tilde{u}(t) = r(t+k) + F(p)\omega(t+k). \tag{6.47}$$

Multiplying both sides of eqn. 6.47 by $t(p)d(p)$ and using eqn. 3.89 one has

$$\begin{aligned} & p^{-k}t(p)N(p)d(p)x_R(t) + p^{-k}t(p)c_\omega(p)d(p)x_{R\omega}(t) + \overline{\lambda}d(p)t(p)\tilde{u}(t) = \\ = \quad & t(p)d(p)\left[r(t+k) + F(p)\omega(t+k)\right]. \end{aligned}$$

Using eqns. 3.89, 6.2 and 6.44

$$w(p)u(t) = t(p)\left[d(p)r(t+k) + (p^{-k}F(p)d(p) - p^{-k}c_\omega(p))\omega(t)\right],$$

and using eqn. 6.40 one has

$$w(p)u(t) = t(p)d(p)r(t+k) - t(p)\alpha_\omega(p)C_\omega(p)\omega(t). \tag{6.48}$$

Multiplying eqn. 6.47 by $p^{-k}t(p)N(p)$ and using eqns. 6.2 and 3.89 one has

$$p^{-k}t(p)N(p)y(t+k) + \overline{\lambda}p^{-k}s(p)d(p)N(p)x_R(t) = p^{-k}t(p)N(p)\left[r(t+k) + F(p)\omega(t+k)\right].$$

Using eqns. 3.89 and 6.44

$$\begin{aligned} w(p)y(t+k) &= p^{-k}t(p)N(p)r(t+k) + \\ &+ p^{-k}t(p)N(p)F(p)\omega(t+k) + \overline{\lambda}p^{-k}s(p)c_\omega(p)d(p)x_{R\omega}(t), \end{aligned}$$

and from eqn. 3.89

$$w(p)y(t+k) = p^{-k}t(p)N(p)r(t+k) + \left[p^{-k}t(p)N(p)F(p) + \overline{\lambda}s(p)c_\omega(p)\right]\omega(t+k). \tag{6.49}$$

Multiplying eqn. 6.47 by $p^{-k}t(p)N_F(p)$ and using eqns. 3.89 and 6.2 one has

$$\begin{aligned} & p^{-k}t(p)N_F(p)\left[p^{-k}N(p)x_R(t) + p^{-k}c_\omega(p)x_{R\omega}(t)\right] + \overline{\lambda}p^{-k}s(p)d(p)N_F(p)x_R(t) \\ = \quad & p^{-k}t(p)N_F(p)\left[r(t+k) + F(p)\omega(t+k)\right]. \end{aligned}$$

Using eqn. 3.89

$$\begin{aligned} & p^{-k}t(p)N(p)\left[y_F(t+k) - p^{-k}C_\omega(p)x_{R\omega}(t)\right] \\ & + \overline{\lambda}s(p)d(p)\left[y_F(t+k) - p^{-k}C_\omega(p)x_{R\omega}(t)\right] + p^{-k}t(p)N_F(p)c_\omega(p)x_{R\omega}(t+k) \\ & = p^{-k}t(p)N_F(p)\left[r(t+k) + F(p)\omega(t+k)\right], \end{aligned}$$

and from eqns. 3.89 and 6.44

$$\begin{aligned} w(p)y_F(t+k) &= p^{-k}t(p)N_F(p)r(t+k) + \\ &+ \left[p^{-k}t(p)N_F(p)F(p) + \overline{\lambda}s(p)C_\omega(p)\right]\omega(t+k) \\ &+p^{-k}t(p)\left[N(p)C_\omega(p) - N_F(p)c_\omega(p)\right]x_{R\omega}(t+k). \end{aligned}$$

Multiplying the above equation by $d(p)$, and using eqns. 3.89 and 6.44 one has

$$\begin{aligned} d(p)w(p)y_F(t+k) &= p^{-k}d(p)t(p)N_F(p)r(t+k) + [w(p)C_\omega(p)+ \\ &+p^{-k}t(p)N_F(p)(F(p)d(p) - c_\omega(p))\Big]\,\omega(t+k), \end{aligned}$$

and from eqn. 6.40

$$\begin{aligned} w(p)d(p)y_F(t+k) &= p^{-k}t(p)d(p)N_F(p)r(t+k) + \\ &+ [w(p)C_\omega(p) - t(p)N_F(p)\alpha_\omega(p)C_\omega(p)]\,\omega(t+k). \quad (6.50) \end{aligned}$$

Finally, eqn. 6.43 follows from eqns. 6.48, 6.49 and 6.50.

Part (iii).

The description of the closed-loop system, given by eqn. 6.43, relates "outputs" $u(t)$, $y(t)$ and $y_F(t)$ to "inputs" $r(t)$ and $\omega(t)$ via the LDO models. The LDO representation is equivalent to an observable state-space model [9, p. 27]. Furthermore, the reference sequence $\{r(t)\}$ is bounded and the noise sequence $\{\omega(t)\}$ is sample mean-square bounded almost surely (see Assumption 3.11, p. 113). Hence, part (iii) follows from [9, Lemma B.3.3, p. 486] subject to conditions 6.45 and 6.46.

Q.E.D.

The control law 6.39 requires $\beta_0 \neq 0$; this is guaranteed in view of Comment 3.4 (p. 99).

It is particularly interesting to compare the WMV-TIMO control law with its TISO counterpart, i.e., the WMV controller for a system having the feedback configuration FI. The RDO model of the corresponding TISO system is given by eqn. 3.89

(p. 92) with $y_F(t) = y(t)$. It can be shown using the optimal TISO-type predictor 3.137 (p. 108) and following the derivation of the WMV controller for a TISO system given by the ARMAX model [83, Theorem 4.1, p. 175] [9, Theorem 10.3.4, p. 418] [59, Theorem 12.4, p. 301], that the WMV-TISO control law is given by

$$u(t) = \left[\begin{array}{ccccc} \rho & \rho\alpha_{sw}(p) & \rho p^{-1}(\beta_{sw}(p) - \beta_0) + \lambda\bar{s}(p) & \lambda\bar{t}(p) & c_\omega(p) - 1 \end{array} \right] \left[\begin{array}{r} r(t+k) \\ -y(t) \\ -u(t-1) \\ \tilde{u}(t-1) \\ y^o(t+k|t) \end{array} \right], \tag{6.51}$$

where parameter ρ is defined by eqn. 6.41 and polynomials $\alpha_{sw}(p)$ and $\beta_{sw}(p)$ result from eqns. 3.138 and 3.139 (p. 108).

The optimality of the WMV-TISO and WMV-TIMO controllers.

The WMV-TISO controller is based on the optimal k-step-ahead TISO-type predictor given by eqn. 3.137 (p. 108). The variances of the prediction errors for both TISO- and TIMO-type predictors are identical (see Comment 3.6, p. 108). Hence, it is not surprising that

Comment 6.2 *Both the WMV-TISO and WMV-TIMO strategies minimize variance of the tracking error identically.*

This can be seen by comparing the relations between $y(t)$ and $r(t)$, and between $y(t)$ and $\omega(t)$ for the closed-loop system resulting from the WMV-TIMO control law (see eqn. 6.43) and resulting from the WMV-TISO control law (see [9, Theorem 10.3.4, p. 418] for the corresponding result).

Comparison of assumptions concerning the noise polynomial $c_\omega(z^{-1})$ required for the WMV-TISO and WMV-TIMO controllers.

The noise polynomial $c_\omega(z^{-1})$ is a factor of the characteristic polynomial of the closed-loop system involving the WMV-TISO controller [83, Remark 4, p. 176] [59, p. 294].

Therefore, it is required that $c_\omega(z^{-1})$ has its roots *strictly inside the unit circle*, i.e.,

$$c_\omega(z^{-1}) \neq 0 \quad \text{for} \quad |z| \geq 1. \tag{6.52}$$

For TISO systems with $c_\omega(z^{-1})$ having roots on the unit circle, a *suboptimal* control law can be derived using the (time-invariant) restricted complexity predictor [9, Theorem 10.3.3, p.416] (see also example 6.2.3, p. 255).

On the other hand, the noise polynomial $c_\omega(z^{-1})$ is not a factor of the characteristic polynomial of the closed-loop system involving the WMV-TIMO control law (see eqns. 6.45 and 6.46). Therefore, the WMV-TIMO controller permits roots of $c_\omega(z^{-1})$ *on the unit circle* (cf. Comment 3.7, p. 110).

The effect of arbitrary initial conditions on the performance of the closed-loop system resulting from the WMV-TIMO and WMV-TISO controllers.

The design of the WMV controllers is based on the optimal predictors. To ensure optimality of the prediction for all sample instants it is necessary to consider appropriate initial conditions for the predictor (see p. 110). Similarly, to ensure optimality of the WMV-TIMO control law for all sample instants, the initial conditions must be considered. However, if arbitrary initial conditions are chosen, then it follows from eqns. 6.43, 6.45, and 6.46 that the input and output sequences $\{u(t)\}$, $\{y(t)\}$, $\{y_F(t)\}$, for the closed-loop system converge asymptotically to the input and output sequences that would have resulted from the use of the control law 6.39 with correct initial conditions. Hence, the WMV-TIMO strategy yields optimal performance asymptotically for arbitrary initial conditions. Note that this property holds *regardless* of the location of the roots of the noise polynomial $c_\omega(z^{-1})$.

On the other hand, the WMV-TISO control law yields optimal performance asymptotically for arbitrary initial conditions if $c_\omega(z^{-1})$ has all roots *strictly inside the unit circle* (cf. [9, Theorem 10.3.2, p. 414]). In contrast to the WMV-TIMO control law, the rate of decay of the effect of incorrect initial conditions depends on the location of the roots of $c_\omega(z^{-1})$ for the WMV-TISO strategy. If $c_\omega(z^{-1})$ has roots on the unit circle, then the WMV-TISO controller based on the restricted complexity predictor yields *suboptimal* performance asymptotically for arbitrary initial conditions [9, Theorem 10.3.3, p. 416].

Similarity between the MPE controllers for stochastic and deterministic multi-output systems.

Comment 6.3 *Note that the WMV-TISO control law 6.51 involves the optimal prediction $y^o(t+k_p|t)$ of the controlled output filtered by the polynomial $c_\omega(p)-1$. On the other hand, the WMV-TIMO control law 6.39 does* not *involve predictions. In fact, the same sequences are involved in the WMV-TIMO control law for* stochastic *systems and the MPE-SIMO control law 6.5 for* deterministic *systems. This similarity between controllers for stochastic and deterministic* multi-output *systems is especially important in self-tuning control, since a number of properties of self tuners for deterministic systems extends to the stochastic case (see subsection 6.2.2).*

Output measurement configuration.

The feedback configuration FD permits implementation of the WMV-TIMO controller without feedback from the controlled output $y(t)$ (see example 6.2.1). Therefore the controlled output need not to be measured. However, the measurement of more than one output ($f > 1$) is required (see Assumption 3.7, p. 91).

Example 6.2.1.

In this example the flexibility of the *output measurement* configuration offered by the feedback configuration FD is illustrated. For this purpose we shall consider the MV control law as a special case of the WMV controller. In particular, it is shown that the MV control can be achieved *without* feedback from (i.e., without measurement of) the controlled output.

Let us consider a three output, third order system ($n = 3$) defined by the state-space model 3.134 or by the RDO representation 3.135 (p. 106). The MV regulator is designed to minimize variance of the output $y(t) = y_1(t)$, ($m = 1$). The two remaining outputs are used for feedback, i.e., $y_F(t) = [y_2(t)\ y_3(t)]'$, $(f = 2)$. The block diagram of the closed-loop system is depicted in fig. 6.9.

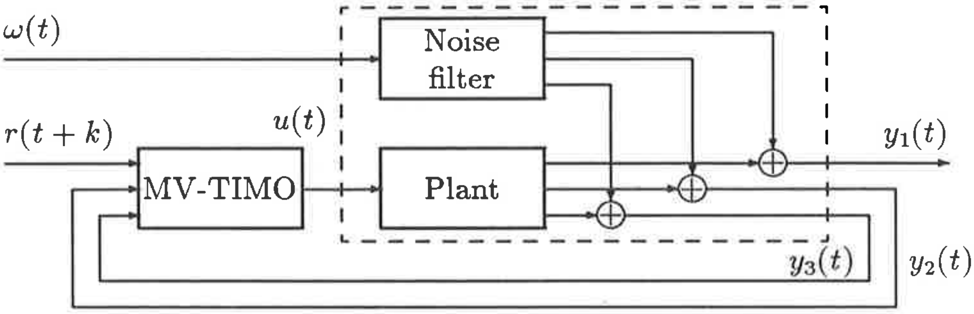

Figure 6.9: The block diagram of the closed-loop system.

The MV-TIMO control law is given by eqn. 6.39 with $\lambda = 0$. The coefficients of the polynomials $\alpha_\omega(p)$ and $\beta_\omega(p)$ (of the $(k = 1)$-step-ahead predictor) required by the MV control law 6.39 were calculated from eqn. 3.113 (p. 99) using the RDO model (cf. example 3.3.1, p. 105) and are given by

$$\begin{aligned}\alpha_\omega &= [\alpha_{20}\ \alpha_{30}\ \alpha_{21}\ \alpha_{31}\ \alpha_{22}\ \alpha_{32}] = \\ &= [3.3928\ \ -3.0228\ \ -3.9392\ \ 3.3119\ \ 1.1879\ \ -0.0831]\,, \\ \beta_\omega &= [\beta_0\ \beta_1\ \beta_2] = [1\ \ -0.3964\ \ 1.6032]\,.\end{aligned}$$

The variance of the noise $\omega(t)$ was set to $\sigma^2 = 1$. In fig. 6.10 the output $y(t) = y_1(t)$ is shown for the open-loop and closed-loop system configuration. The control signal $u(t)$ generated by the MV-TIMO regulator is shown in fig. 6.10 as well.

For the purpose of quantitative comparison of the effect of the regulation, the variances of all outputs were calculated from the simulation data for the open-loop and closed-loop system configurations. Number of samples was N=200 and N=1000 (cf. eqn. 6.67, p. 240). Furthermore, the theoretical open-loop system variances were evaluated using the method presented in [83, Theorem 2.3, p. 121]. These results are presented in table 6.1. Note that the variance of the controlled output was reduced close to the theoretically minimal value of 1. Furthermore, the MV-TIMO regulation of the output $y(t) = y_1(t)$ had a desirable effect on the feedback outputs $y_F(t)$ as

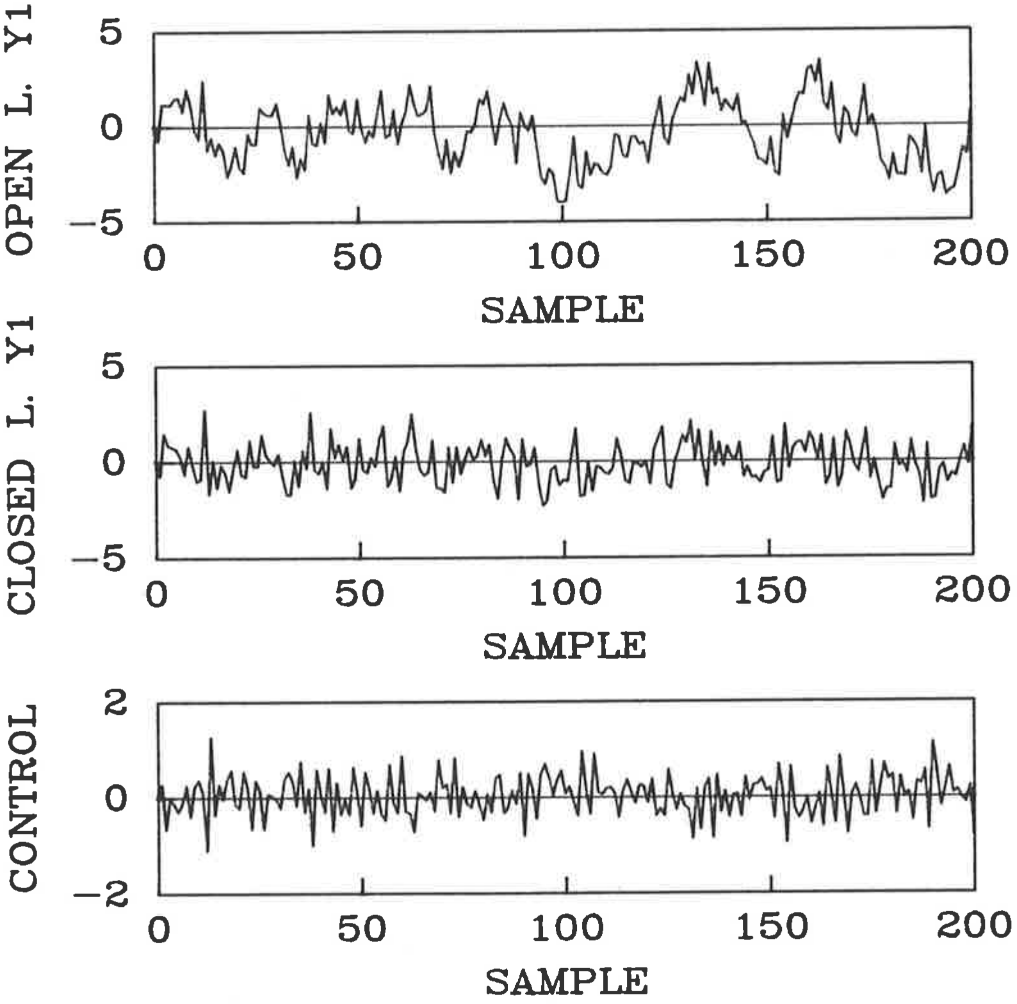

Figure 6.10: The sequences of the open-loop output $y_1(t)$, and the closed-loop system controlled output $y(t) = y_1(t)$, and control input $u(t)$ generated by the MV-TIMO controller without feedback from the controlled output.

Output variance					
	open-loop			closed-loop	
N	theoretical	200	1000	200	1000
$\sigma^2_{y_1}$	2.05	2.87	2.15	1.13	1.02
$\sigma^2_{y_2}$	2.16	2.95	2.25	1.15	1.06
$\sigma^2_{y_3}$	1.32	1.55	1.35	1.19	1.12

Table 6.1: The output variances derived from theoretical and from simulation results for the open-loop system and closed-loop system using MV-TIMO controller implemented without feedback from the controlled output.

well, i.e., the variances of $y_2(t)$ and $y_3(t)$ were reduced in comparison with the open-loop system. The open-loop and closed-loop sequences of the feedback outputs are depicted in fig. 6.11.

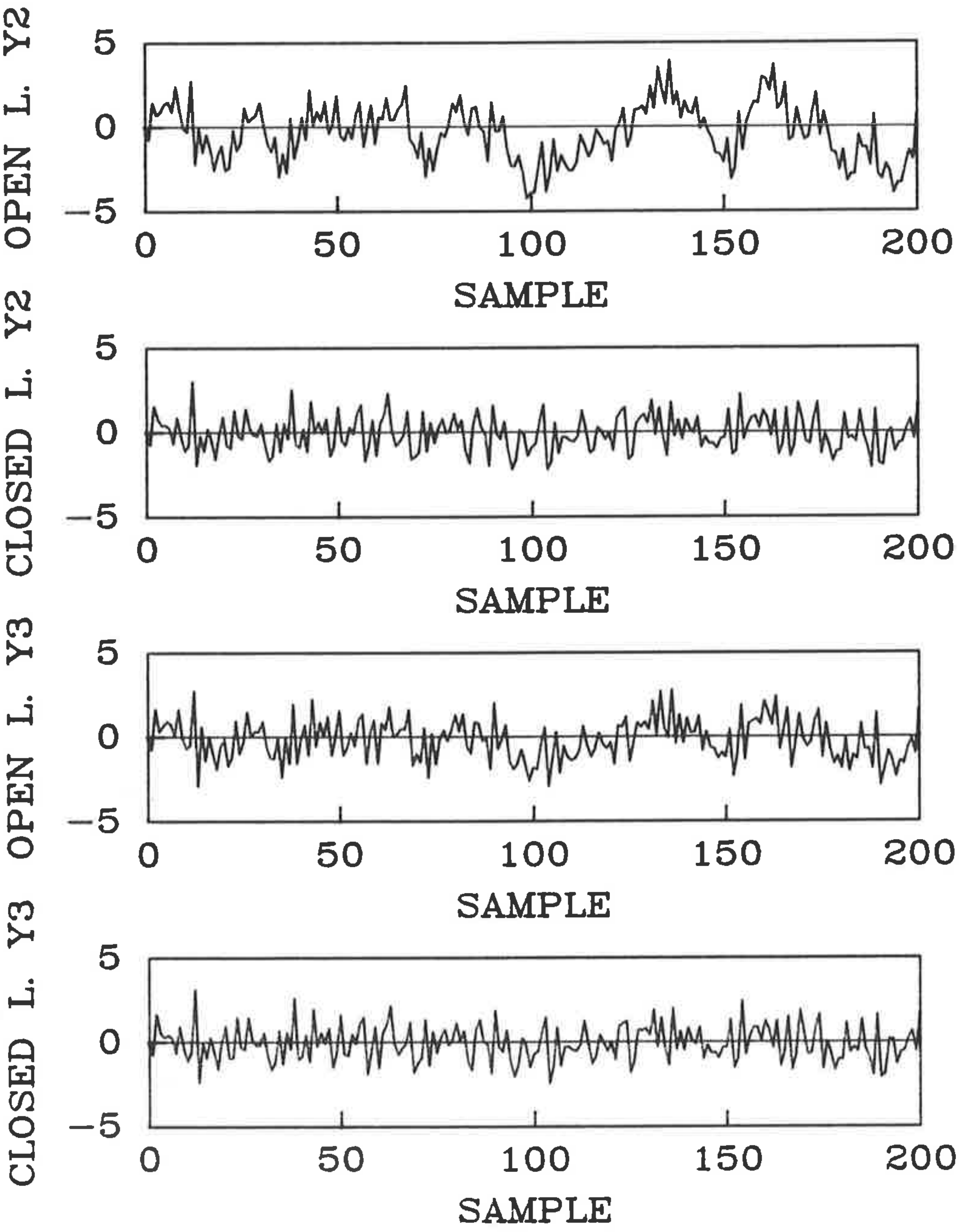

Figure 6.11: The sequences of the feedback outputs $y_2(t)$ and $y_3(t)$ for the open-loop and closed-loop system configurations (the MV-TIMO controller without feedback from the controlled output).

6.2.2 A self-tuning minimum variance controller in the feedback configuration FD.

In this subsection the self-tuning minimum variance strategy for a TIMO system having the feedback configuration FD is introduced. The strategy illustrates possible benefits resulting from utilization of additional outputs for feedback in self-tuning control of stochastic systems. The MV control law is considered here, but the *weighted* MV self-tuning controller can be developed and analyzed in a similar way.

Self-tuning MV controller (ST-MV-TIMO strategy).

Let us consider a system satisfying Assumptions 3.7 to 3.10 (p. 91) and 3.11 (p. 113). The WMV control law for a system with known parameters is given in Theorem 6.1. In order to introduce a self-tuning version of the MV control law ($\lambda = 0$ in eqn. 6.39) for a system with unknown parameters, we make the following assumption.

Assumption 6.4 *(i) For the purpose of parameter estimation, the present value $y(t)$ of the controlled output is known;*

(ii) the time delay k from the control input $u(t)$ to the controlled output $y(t)$ is known;

(iii) $z^k N(z^{-1}) \neq 0$ and $d(z^{-1}) \neq 0$ for $|z| \geq 1$.

Assumption 6.4 (i) is required for the calculation of the equation error E.1 (Appendix E) of the parameter estimator employed in the self tuner. Assumption 6.4 (ii) guarantees (in view of Comment 3.4, p. 99) that $\beta_0 \neq 0$; part (iii) is required to ensure closed-loop system stability under the MV control (see Theorem 6.1, part (iii)).

The control objective of the ST-MV-TIMO strategy is to minimize the (conditional) mean-square tracking error with probability one by applying a sample mean-square bounded control sequence $\{u(t)\}$. The feedback outputs sequence $\{y_F(t)\}$ will be shown to be sample mean-square bounded with probability one.

In order to derive the ST-MV-TIMO strategy, recall that the TIMO-type k-step-ahead predictor can be written as (see eqns. 3.140 to 3.142 (p. 112) and note that $k_i = k - k_p = 0$)

$$\begin{aligned} y^o(t|t-k) &= \Theta'\phi(t-k), && (6.53) \\ \Theta &= [\alpha_\omega \ \beta_\omega]', && (6.54) \\ \phi(t) &= \begin{bmatrix} y_F(t)' & \dots & y_F(t-n_p)' & u(t) & \dots & u(t-n_p) \end{bmatrix}'. && (6.55) \end{aligned}$$

Eqn. 6.53 defines system model the parameters of which are estimated in the ST-MV-TIMO strategy. Hence, denote the vector of parameter estimates of Θ resulting from a recursive parameter estimation algorithm by

$$\widehat{\Theta}(t) = \left[\widehat{\alpha}_\omega(t) \ \widehat{\beta}_\omega(t)\right]'. \qquad (6.56)$$

The RLS estimator with the condition number monitoring (CNM) technique is employed in the ST-MV-TIMO strategy (see Appendix E). The CNM technique was proposed in [57] as a modification to the RELS estimator and it guarantees boundedness of the condition number of the estimate error covariance matrix.

The **ST-MV-TIMO controller** consists of the following steps which are performed at every sample instant t:

step 1. calculate the vector of estimates $\widehat{\Theta}(t)$ using the RLS-CNM algorithm (see eqns. E.5 to E.7, Appendix E), with the equation error defined as

$$e(t) = y(t) - \widehat{\Theta}(t-1)'\phi(t-k); \qquad (6.57)$$

step 2. calculate the control signal

$$u(t) = (\widehat{\beta}_0(t))^{-1} \begin{bmatrix} 1 & \widehat{\alpha}_\omega(t,p) & p^{-1}\left(\widehat{\beta}_\omega(t,p) - \widehat{\beta}_0\right) \end{bmatrix} \begin{bmatrix} r(t+k) \\ -y_F(t) \\ -u(t-1) \end{bmatrix}, \qquad (6.58)$$

where $\widehat{\alpha}_\omega(t,p) = \widehat{\alpha}_\omega(t)P_{n_p}$, and $\widehat{\beta}_\omega(t,p) = \widehat{\beta}_\omega(t)P_{n_p}$ ($P_{n_p} = [1 \ p \ \dots \ p^{n_p}]'$).

Comment 6.4 *In order to eliminate possibility of the estimate $\widehat{\beta}_0(t)$ becoming zero, one can implement the constrained estimation technique (as for the ST-MPE-SIMO strategy, see step 2, eqns. 6.19 and 6.20, p. 201).*

Comment 6.5 *The implementation of the RLS-CNM estimator requires calculation of the maximum eigenvalue* $\lambda_{\max}(P_m(t-k))$ *of the covariance matrix* $P_m(t-k)$ *at every sample instant* t *(see eqn. E.7, Appendix E). Since* $\lambda_{\max}(P_m(t-k)) < \mathrm{tr}(P_m(t-k))$*, where "tr" denotes trace, the covariance eigenvalue in eqn. E.7 can be replaced by the trace of covariance matrix, as proposed in [57].* [3] *Such a modification reduces the computational burden. The simulation studies confirm the applicability of the modification (see subsection 6.2.3).*

Convergence of the ST-MV-TIMO strategy.

The convergence analysis of the ST-MV-TIMO strategy is based on a Martingale convergence theory [87, chapter 7] [171, chapter 2]. For the purpose of analysis, the methodology developed in [55,57] [9, chapter 9 and 11] for establishing convergence of stochastic self-tuning prediction and control algorithms for systems having the feedback configuration FI is employed. This approach is applied here to establish convergence of the stochastic self-tuning control for systems having the feedback configuration FD.

We have the following global convergence result.

Theorem 6.2 *Consider a system satisfying Assumptions 3.7 to 3.10 (p. 91), 3.11 (p. 113), and 6.4 (p. 228). Assume that future value of the bounded reference sequence* $r(t+k)$ *is known (or is computable) at time* t*. If* $\widehat{\beta}_0(t) \neq 0$ *for all sample instants* t*, then the ST-MV-TIMO strategy (implemented with the RLS-CNM estimator with no forgetting) ensures with probability one*

$$\limsup_{N\to\infty} \frac{1}{N} \sum_{t=k}^{N} u(t)^2 < \infty, \tag{6.59}$$

$$\limsup_{N\to\infty} \frac{1}{N} \sum_{t=k}^{N} \|y_F(t)\|^2 < \infty, \tag{6.60}$$

$$\limsup_{N\to\infty} \frac{1}{N} \sum_{t=k}^{N} y(t)^2 < \infty, \tag{6.61}$$

[3] This method can be applied to the adaptive TIMO-type predictor as well (see eqns. 3.143 to 3.145, p. 112).

$$\lim_{N\to\infty}\frac{1}{N}\sum_{t=k}^{N}E\{[y(t)-r(t)]^2\,|\mathcal{F}_{t-k}\}=FF'\sigma^2. \tag{6.62}$$

Proof: see Appendix F.

Comment 6.6 *The global convergence of the ST-MV-TIMO algorithm means that for all initial states of the system and the algorithm, the (conditional) mean-square tracking error (see eqn. 6.62) is minimized with probability one by applying a sample mean-square bounded input sequence $\{u(t)\}$ (see eqn. 6.59), and the feedback sequence $\{y_F(t)\}$ is sample mean-square bounded (see eqn. 6.60). Furthermore, it follows from eqn. 6.43 that $FF'\sigma^2$ is the minimal possible mean-square tracking error achievable with any linear causal control law, including that designed for a system with known parameters.*

Comment 6.7 *The parameter estimates difference convergence, i.e.,*

$$\lim_{N\to\infty}\sum_{t=1}^{N}\|\widehat{\Theta}(t)-\widehat{\Theta}(t-i)\|^2<\infty \quad \text{for} \quad t\geq i \quad a.s,$$

for any finite i, can be readily established as in [57]. This ensures that the estimates remain bounded with probability one for all t.

The properties of the MV-TIMO control law (see eqn. 6.39 for $\lambda = 0$) which are relevant to self-tuning control are discussed below. These properties are compared with the corresponding features of the MPE self tuners developed for TISO systems, i.e., systems having the feedback configuration FI.

Comparison of the expected convergence rate of self-tuning controllers for TISO and TIMO systems.

The (weighted) minimum variance control law 6.51 for TISO systems involves optimal prediction of the controlled output, $y^o(t+k|t)$, filtered by the polynomial $c_\omega(p)-1$ (unless $c_\omega(p)=1$). Therefore, in the self-tuning implementation of the WMV-TISO control law (ST-WMV-TISO strategy) the coefficients of $c_\omega(p)$ must be estimated (see eqn. E.12, Appendix E). Furthermore, in the ST-WMV-TISO strategy the (unknown) optimal prediction $y^o(t+k|t)$ is *approximated* by the *a priori* output prediction (see e.g., [55,125], and eqn. E.14, Appendix E)

$$\widehat{y}(t+k) = \widehat{\Theta}_s(t)'\phi_s(t),$$

or by the *a posteriori* output prediction if $k=1$ (see e.g., [56,57,172,69,134]) [4]

$$\tilde{y}(t+1) = \widehat{\Theta}_s(t+1)'\phi_s(t).$$

The regression vector $\phi_s(t)$ of the estimator employed in the ST-WMV-TISO strategy involves output predictions (see eqn. E.13, Appendix E), which depend on the parameter estimate vector. Therefore, the regression vector depends on the parameter estimate vector. Such estimation algorithms are said to be of the *pseudo-linear regression* type, due to the *approximation* resulting from ignoring the dependence of $\phi_s(t)$ on Θ_s [173] [9, Remark 8.4.2, p. 320]. The following pseudo-linear regression estimation methods (or their variants) have been commonly employed in self-tuning controllers for TISO systems: stochastic gradient (SG) [53], recursive extended least squares (RELS), and recursive maximum likelihood (RML) (see Appendix E).

It is well known that the convergence rate of parameter estimates for the SG algorithm, which is a scalar gain estimator, is inferior to the convergence rates for the *matrix gain* estimators, such as the RELS or RML [56,57,69]. Therefore it is desirable to employ the matrix gain estimators in self-tuning control, although they are computationally more involved.

The RELS and RML estimators were developed using two different approaches [174,175]. The RML is an *approximation* to the nonrecursive version [176] due to the simplifications made in its derivation (see e.g., [175] [160, p. 376]). As pointed out in [53], the relation between the RELS and RML estimators can be seen by comparison of the calculation of the gradient vector $\psi(t)$ (see eqn. E.17, Appendix E)

$$D(t,p)\psi(t) = \phi_s(t), \tag{6.63}$$

[4] The control law is sometimes formulated in such a way that *a priori* or *a posteriori* prediction errors replace the corresponding output predictions [172,134].

where $D(t,p)$ is a filter defined as

$$D(t,p) = 1 \quad \text{for the RELS algorithm;} \tag{6.64}$$

$$D(t,p) = \widehat{c}_\omega(t,p) \quad \text{for the RML algorithm} \tag{6.65}$$

($\widehat{c}_\omega(t,p)$ is the estimate of $c_\omega(p)$). The RELS method can be interpreted as *approximation* to the RML method since the fixed filter 6.64 is used in place of 6.65 [177]. The convergence rate of the RML method is superior to that of the RELS at the expense of additional computations [53].

The *pseudo-linear* regression estimators SG, RELS, and RML, involve the above mentioned *approximations*; it is therefore desirable to consider the *linear regression* estimators, such as the RLS, for self-tuning control of TISO systems. As already mentioned, the coefficients of the noise polynomial $c_\omega(p)$ must be estimated in a general case of $c_\omega(p) \neq 1$ [57, Remark 3.1]. However, in the case of the *minimum variance regulation*, i.e., $r(t) =$ constant and $\lambda = 0$, some simplifications are possible. Namely,

(i) if $r(t) = 0$, then the MV control law implies that $\widehat{y}(t) = 0$ and the output predictions and coefficients of $c_\omega(p)$ can be removed from the regression and parameter estimate vectors, respectively [22];

(ii) if $r(t) =$ constant ($\neq 0$), then it suffices to replace past predictions in the regression vector by a constant, and estimates of the coefficients of $c_\omega(p)$ by a scaled sum of coefficients in the parameter estimate vector [9, p. 445].

Hence, for TISO systems with $c_\omega(p) \neq 1$, the RLS estimator can be employed only for self-tuning *minimum variance regulation.*

On the other hand, for the ST-MV-TIMO *controller* the output predictions are *not included* in the regression vector 6.55, and the coefficients of the noise polynomial $c_\omega(p)$ are *not estimated* (see eqn. 6.56). (In fact, the same can be said about the self-tuning *weighted* MV strategy based on the control law 6.39.) Therefore, one can apply the *linear regression* estimator, such as the RLS, to obtain unbiased parameter estimates for a *stochastic system with coloured noise* ($c_\omega(p) \neq 1$) in a general *reference* *tracking* (rather than regulation) problem. (Furthermore, in the case of the self-tuning *weighted* MV strategy for TIMO systems, the linear regression estimator can be used for $\lambda \geq 0$ rather than only for $\lambda = 0$ as for TISO systems.) This is a powerful result with significant practical implications in terms of the convergence rate of parameter estimates and modification of system assumptions required for convergence.

Comment 6.8 *It can be expected that the convergence rate of parameter estimates of the ST-MV-TIMO strategy will be superior to that of self-tuning controllers developed for TISO systems. This is due to the use of the linear regression estimator (RLS-CNM) which is free of the approximations involved in the SG, RELS, and RML methods, which are pseudo-linear regression estimators.*

The superior convergence rate of the ST-MV-TIMO strategy is confirmed by simulation studies described in subsection 6.2.3.

Aside 6.1 *Note also that superior convergence rate is usually observed for the plant parameter estimates, i.e.,* $\widehat{A}(p)$ *and* $\widehat{B}(p)$ *(or estimates involving plant parameters), in comparison with that of the noise polynomial estimates* $\widehat{C}(p)$ *in the ARMAX model for the RELS or RML methods [128] [160, p. 384]. The difference in convergence rate results form the fact that* $\widehat{C}(p)$ *is estimated by the pseudo-linear regression terms in the regression vector [9, p. 445]. This supports Comment 6.8.*

Furthermore, it is known that

Comment 6.9 *In a practical situation, growth of the condition number of the covariance matrix of the estimate error might lead to numerical instability of the estimator. This, in turn, may result in instability of the closed-loop system [6]. The pseudo-linear regression estimators are susceptible to the growth of the condition number of the covariance matrix [57]. However, the numerical robustness of the estimator in the ST-MV-TIMO strategy is enhanced as a result of employing the ordinary linear regression algorithm (see example 6.2.2, p. 240).*

Finally, note that both the ST-MV-TIMO strategy for *stochastic* systems and the ST-MPE-SIMO strategy of subsection 6.1.2 for *deterministic* systems employ the RLS estimator (cf. Comment 6.3, p. 224).

Comparison of assumptions concerning the noise polynomial $c_\omega(z^{-1})$ in self-tuning control of TISO and TIMO systems.

The next important aspect of the ST-MV-TIMO strategy is the modification of assumptions required by self tuners developed for TISO systems and concerning the noise polynomial $c_\omega(z^{-1})$.

The first modification was introduced for the nonadaptive case by permitting the roots of the noise polynomial $c_\omega(z^{-1})$ to lie *on the unit circle*, in contrast to the TISO schemes, which require all roots of $c_\omega(z^{-1})$ to be located strictly inside the unit circle [5]. In the light of the Spectral Factorization Theorem (SFT), however, the assumption about roots of $c_\omega(z^{-1})$ lying *strictly inside the unit circle* is a mild one, especially if there are no uncontrollable system modes on the unit circle [139, p. 373] [178] [9, p. 264]. (Recall that the open-loop system is said to be controllable, see Assumption 3.8, p. 92.)

The second modification of system assumptions is related to the convergence of self tuners. It is well known that for the (global or parameter) convergence of self-tuning controllers for TISO systems, certain transfer function involving the noise polynomial $c_\omega(z^{-1})$ is required to be *strictly positive real* [53] (see discussion on pp. 324-325, Appendix G). For example, convergence of self-tuning controllers employing the RELS estimator requires that

$$\frac{1}{c_\omega(z^{-1})} - \frac{1}{2} \tag{6.66}$$

is strictly positive real (i.e., $c_\omega(z^{-1})$ has all roots inside the unit circle and $c_\omega^{-1}(e^{i\varphi}) + c_\omega^{-1}(e^{-i\varphi}) - 1 > 0$, for $0 \leq \varphi \leq 2\pi$ [179,134]). On the other hand, note from Theorem 6.2 (see also Comment F.5, p. 313) that

Comment 6.10 *The (global) convergence of the ST-MV-TIMO strategy does not require any positive real condition, in contrast to self-tuning controllers developed for TISO systems without any special precautions being taken (see Appendix G)* [6].

[5] Alternatively, controllers based on the time-varying or restricted complexity predictors can be used, see discussion on pp. 222-224 and [9, p. 450].

[6] This property is illustrated in example 6.2.2, p. 240.

The *removal of the strictly positive real (SPR) condition* is the most significant feature of the ST-MV-TIMO strategy apart of the replacement of the pseudo-linear regression estimator by the linear regression estimator. Note, however, that the ST-MV-TIMO strategy requires stability of the open-loop system (see Assumption 6.4 (iii), p. 228). For self-tuning applications such stability can be verified more easily than the SPR condition which is dependent upon unknown parameters of the noise polynomial.

A survey of self-tuning MPE controllers for TISO systems with respect to convergence analysis and methods of overcoming the strictly positive real condition.

In order to assess the significance of the global convergence of the ST-MV-TIMO strategy and the significance of system assumptions under which it was established, it seems desirable to present a survey of stochastic self-tuning controllers developed for TISO systems. Such a survey, with respect to convergence analysis and methods of overcoming the SPR condition, is presented in Appendix G.

The first self tuner which was shown to be globally convergent employs the scalar gain estimator (stochastic gradient SG) [55]. It is known, however, that the convergence rate of parameter estimates of the *matrix* gain estimators is markedly superior to that of the *scalar* gain estimators [56,57,69]. Therefore, considerable research was devoted to the development of convergent self-tuning strategies employing matrix gain estimators rather than scalar gain estimators, for the purpose of improving the convergence rate. Three approaches to the development of such strategies are identified (see Appendix G): (i) estimator modification (method E), (ii) dither injection (method D), and (iii) estimator modification with dither injection (method E-D).

The ST-MV-TIMO strategy relies on the method E (because of the CNM technique) to guarantee global convergence of the self-tuning controller employing a matrix gain estimator. The CNM technique implies that a full least-squares step in parameter estimate update is taken only if this is consistent with the convergence criterion related to the condition number of the covariance matrix; otherwise, the gain of the algorithm is reduced (see eqns. E.8 and E.10, Appendix E). It was suggested in [57] that the estimator could revert to a full least-squares update if temporary convergence difficulties had been overcome.

Some authors have expressed their objections in relation to the matrix gain estimator combined with the CNM technique as a means for improving the convergence rate of a self tuner for TISO systems. In particular, it was pointed out in [69] that the analysis of the CNM technique presented in [57] does not establish whether the estimator approaches the full LS update asymptotically [7]. Similarly, the superiority of the matrix gain estimator with the CNM technique over the scalar gain estimator was questioned in [180].

It should be noted, however, that the importance of the above comments on the CNM technique decreases in the application of CNM to self-tuning control of TIMO systems. This is because a significant improvement in the convergence rate is expected due to the replacement of the pseudo-linear regression estimators employed for self-tuning control of TISO systems (i.e., estimators for which the objections about the CNM technique were expressed) by the linear regression estimator. Nevertheless, some other modifications to the matrix gain estimators proposed for TISO systems (e.g., the weighted least squares estimator introduced in [56]) could be considered for the development of a globally convergent self-tuning controller for TIMO systems.

Furthermore, in comparison with other approaches to the development of globally convergent self tuners employing matrix gain estimators for TISO systems (see Appendix G) we can conclude that

Comment 6.11 *The ST-MV-TIMO strategy leads to an optimal closed-loop system performance (see eqn. 6.62 and Comment 6.6). On the other hand, the schemes based on methods D and E-D for TISO systems yield suboptimal performance (i.e., with increased tracking error variance) due to white noise dither injected into the closed-loop system.*

In addition to the approaches to the development of strategies employing matrix

[7] Some insight into these aspects of the CNM technique is provided in example 6.2.2, p. 240.

gain estimators, the methods of *overcoming the SPR condition* for TISO systems are also discussed in Appendix G. Three methods relying on (a) dither injection (method D), (b) model overparameterization (method O), and (c) model overparameterization and dither injection (method O-D), are identified. We can conclude that

Comment 6.12 *The utilization of additional system outputs for feedback, which forms the basis of the ST-MV-TIMO strategy, is seen as an effective method of overcoming the SPR condition* [8]. *In particular, such a method avoids drawbacks of the methods of overcoming the SPR condition proposed for TISO systems, which involve*

- *additional computational effort which is required by self-tuning strategies employing two estimators and two controllers, on-line spectral factorization, prefiltering of the input-output data, etc. (method D);*
- *dither injection into the closed-loop system which leads to suboptimal closed-loop system performance unless some other precautions are being taken (methods D and O-D);*
- *model overparameterization which affects the convergence rate of the estimator and might lead to estimator ill-conditioning (method O);*
- *regulation about zero level (i.e., $r(t) = 0$) as the only achievable control objective (method O-D);*
- *certain prior system knowledge concerning the noise polynomial (methods O and O-D).*

The survey in Appendix G reviews the considerable research effort involved in (i) improving the convergence rate of self-tuning controllers (by employing the matrix gain estimators), (ii) overcoming the SPR condition for TISO systems. Both goals (i) and (ii) have been achieved in the present work by utilization of additional system outputs for feedback.

[8] This method of overcoming the SPR condition is not, as it might seem, based on the model overparameterization because the number of estimated parameters for the ST-MV-TIMO scheme may be the same as for the corresponding ST-MV-TISO strategy (see example 6.2.2, p. 240).

6.2.3 Performance improvements in self-tuning control of stochastic systems by utilization of additional outputs for feedback – simulation studies.

Simulation studies which illustrate properties of the self-tuning, minimum prediction error strategy developed for TIMO systems are presented in this subsection. The purpose of the simulation studies is to show benefits resulting from the utilization of additional system outputs for feedback in the design of self-tuning controllers for stochastic systems. Therefore, the results obtained for systems having the feedback configuration FD are compared with those obtained for systems having the feedback configuration FI. In particular, the following aspects of self-tuning control are compared for the ST-MV-TIMO and ST-MV-TISO strategies:

- convergence rate of parameter estimates (see Comment 6.8, p. 234);
- numerical robustness of the estimation algorithms (see Comment 6.9, p. 234);
- elimination of the SPR condition involving the noise polynomial $c_\omega(z^{-1})$ (see Comment 6.10, p. 235);
- performance for systems with the noise polynomial $c_\omega(z^{-1})$ having roots on the unit circle.

Furthermore, some insight is given into properties of the CNM technique and the choice of the values of its parameters.

In order to eliminate any possible differences in the convergence rate due to the number of estimated parameters (cf. Comment 6.1, p. 199), models of the simulated systems were chosen such that the number of the estimated parameters is the same for both ST-MV-TIMO and ST-MV-TISO strategies.

The following measures of the convergence rate of the self-tuning strategy and the closed-loop system performance were employed for the purpose of quantitative comparison of results

- tracking error variance (or controlled output variance σ_y^2 if $r(t) = 0$)

$$\sigma_{e_r}^2 = \frac{1}{N} \sum_{t=1}^{N} [y(t) - r(t)]^2 , \tag{6.67}$$

  where N is a total number of samples;

- on-line tracking error variance (or on-line controlled output variance if $r(t) = 0$) calculated at each sample instant t

$$\sigma_{e_r}^2(t) = \frac{1}{t} \sum_{\tau=1}^{t} [y(\tau) - r(\tau)]^2 ; \tag{6.68}$$

- cumulative loss function

$$\mathrm{CL}(t) = \sum_{\tau=1}^{t} [y(\tau) - r(\tau)]^2 . \tag{6.69}$$

Moreover, the condition number of the estimate error covariance matrix $P(t)$ was calculated. The condition number is defined as $\lambda_{\max}/\lambda_{\min}$, where $\lambda_{\max}$ ($\lambda_{\min}$) is the maximum (minimum) eigenvalue of the covariance matrix.

All simulations were performed on VAX 11/780 digital computer using FORTRAN software and single precision arithmetic. The covariance matrix condition number was calculated using the MATLAB package [110]. The RELS, RML, RLS and RLS-CNM estimators (see Appendix E) were implemented with the U-D factorization method [164] (see also discussion on page 174).

Example 6.2.2. System with a noise polynomial $c_\omega(z^{-1})$ not satisfying the SPR condition.

The purpose of this example is to show that the ST-MV-TIMO strategy is convergent for a system with a noise polynomial $c_\omega(z^{-1})$ such that $c_\omega^{-1}(z^{-1}) - 1/2$ is not strictly positive real.

Let us consider a second order ($n = 2$) system given by the state-space model 2.8

(p. 22), where $y(t) = y_1(t)$ and $y_F(t) = [y_1(t)\ y_2(t)]'$, and

$$A_s = \begin{bmatrix} -0.9 & -0.95 \\ 1 & 0 \end{bmatrix}, \qquad B_s = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad K = \begin{bmatrix} 0.6 \\ 0.2105 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad C_F = \begin{bmatrix} 1 & 0 \\ 0.8 & -0.4 \end{bmatrix}, \qquad C_S = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

The observability index of the pair $(C_F,\, A_s)$ is $\nu_F = 1$. The corresponding RDO representation is given by (see eqns. 3.89 to 3.94, p. 92)

$$\begin{aligned} d(p) &= [1\ \ 0.9\ \ 0.95]\, P_2, \\ N(p) &= [0\ \ 1\ \ 0]\, P_2, \\ c_\omega(p) &= [1\ \ 1.5\ \ 0.75]\, P_2, \\ N_F(p) &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.8 & -0.4 \end{bmatrix} P_2, \\ C_\omega(p) &= \begin{bmatrix} 1 & 1.5 & 0.75 \\ 1 & 1.2958 & 0.4742 \end{bmatrix} P_2. \end{aligned}$$

The delay $k = 1$ (see eqn. 3.91, p. 93).

The problem of the MV control of the output $y(t) = y_1(t)$ is considered in the following example. For the purpose of implementation of the ST-MV-TIMO strategy it is assumed that outputs $y_F(t) = [y_1(t)\ y_2(t)]'$ are available for feedback ($f = 2$). The ST-MV-TIMO strategy involves estimation of coefficients of the 1×2 polynomial vector $\alpha_\omega(p)$ and polynomial $\beta_\omega(p)$ (see eqns. 6.56 (p. 229) and 3.97, 3.98 (p. 94)). Assuming knowledge of the observability index ν_F, the degree of the polynomials is selected according to eqn. 3.133 (p. 105) as $n_p = 1$. The coefficients of the polynomials $\alpha_\omega(p)$ and $\beta_\omega(p)$, calculated from eqn. 3.113 (p. 99) for $k_p = k = 1$, are given by (cf. example 3.3.1, p. 105)

$$\alpha_\omega = [\alpha_{10}\ \alpha_{20}\ \alpha_{11}\ \alpha_{21}] = [7.7081\ \ -7.1081\ \ 4.3873\ \ -6.9388]\,, \tag{6.70}$$

$$\beta_\omega = [\beta_0\ \beta_1] = [1\ \ -2.9216]\,. \tag{6.71}$$

On the other hand, if the only output which is available for feedback is the controlled output (i.e., the system has the feedback configuration FI), then the RDO model of the corresponding TISO system is given by eqn. 3.89 (p. 92) with $y_F(t) \equiv y(t)$. Note that the model of this TISO system can be alternatively written in the following ARMAX form

$$d(p)y(t) = u(t-1) + c_\omega(p)\omega(t). \tag{6.72}$$

The noise polynomial $c_\omega(z^{-1})$ does not satisfy the SPR condition required for convergence of the RELS estimator (see fig. G.1 (p. 325) and note that the point $c_1 = 1.5$, $c_2 = 0.75$ lies outside the shaded area). The above ARMAX model was considered in [181, example 1] to show that the RELS estimator is not convergent for such a system [9].

If only the controlled output is available for feedback, i.e., $y_F(t) = y_1(t)$, then the ST-MV-TISO strategy implementing the control law 6.51 (p. 222) can be employed. The parameters which are estimated in the ST-MV-TISO strategy are the coefficients of the noise polynomial $c_\omega(p)$, and coefficients of polynomials $\alpha_{s\omega}(p)$ and $\beta_{s\omega}(p)$ resulting from eqns. 3.138 and 3.139 (p. 108)

$$\alpha_{s\omega}(p) \;=\; \alpha_{s\omega}P_1 = [\alpha_{s0}\;\;\alpha_{s1}]\,P_1 = [0.6\;\;-0.2]\,P_1, \tag{6.73}$$

$$\beta_{s\omega}(p) \;=\; \beta_{s\omega}P_1 = [\beta_{s0}\;\;\beta_{s1}]\,P_1 = [1\;\;0]\,P_1. \tag{6.74}$$

We shall consider the ST-MV-TISO strategy presented in [9, p. 444]. This self tuner employs the RELS method for estimation of coefficients of the polynomials $\alpha_{s\omega}(p)$, $\beta_{s\omega}(p)$, and $c_\omega(p)$. Therefore, this strategy is referred to as the ST-MV-TISO-RELS scheme. Alternatively, the coefficients can be estimated using the RML method (see Appendix G p. 325, for the discussion of the RML estimator with respect to the methods of overcoming the SPR condition). Then the resulting strategy is referred to as the ST-MV-TISO-RML scheme (see eqns. E.12 to E.17, Appendix E) [10].

We shall consider two control tasks: regulation problem ($r(t) = 0$), and tracking the reference sequence.

[9] Similar examples were presented in [48,58,135].

[10] The ST-MV-TISO-RML strategy was implemented *without* projection facility for the estimates of the noise polynomial $c_\omega(p)$ into the stability region (see discussion on page 325). The estimated polynomial $\widehat{c}_\omega(t,p)$ was stable for all sample instants t.

Regulation problem.

The variance of the noise $\omega(t)$ was set to $\sigma^2 = 1$. For the closed-loop system under MV regulation the minimal variance of the controlled output is $\sigma_y^2 = 1$, since delay $k = 1$ [83, p. 175]. The limit on the control signal magnitude was set to $|u(t)| \leq 10$.

In the case of regulation by the ST-MV-TISO-RELS strategy with $r(t) = 0$, the coefficients of the noise polynomial $c_\omega(p)$ need not be estimated (see case (i), p. 233). However, for the purpose of this simulation the noise polynomial was estimated to compare the results with those obtained with the ST-MV-TISO-RML scheme [11].

All estimators were initialized with $P(-1) = 100I$, the initial parameter estimates were set to zero except for $\widehat{\beta}_0(0) = \widehat{\beta}_{s0}(0) = 1$; no forgetting was used.

The sequences of the control input $u(t)$ and controlled output $y(t)$ resulting from the use of the ST-MV-TISO-RELS strategy are shown in fig. 6.12. The variance of the controlled output (i.e., the tracking error variance, see eqn. 6.67 for $N = 2000$) was $\sigma_y^2 = 1.12$. The estimates of the parameters given by eqns. 6.73 and 6.74, and estimates of coefficients of $c_\omega(p)$ are depicted in fig. 6.13. It can be seen that the estimates converged to biased values. This is because the noise polynomial $c_\omega(z^{-1})$ does not satisfy the SPR condition.

Next the ST-MV-TISO-RML strategy was simulated. The variance of the controlled output was $\sigma_y^2 = 1.12$, i.e., the same as for the scheme based on the RELS estimator. This is due to the fact that parameter estimates, which are depicted in fig. 6.14, were very similar to those obtained with the RELS method. The estimates of the noise polynomial were only slightly different from those of the RELS estimator.

Finally, the ST-MV-TIMO strategy was applied. Two cases were considered: without and with CNM. The parameters to be estimated are given by eqns. 6.70 and 6.71. The control input and controlled output sequences resulting form the ST-MV-TIMO strategy without CNM are shown in fig. 6.15 and the corresponding estimates in fig. 6.16. The variance of the controlled output was $\sigma_y^2 = 1.02$, i.e., it was close to the minimal value which would have resulted if the controller for a system

[11] The RML based strategy requires estimation of $c_\omega(p)$ for calculation of the gradient vector (see eqns. E.15 to E.17, Appendix E).

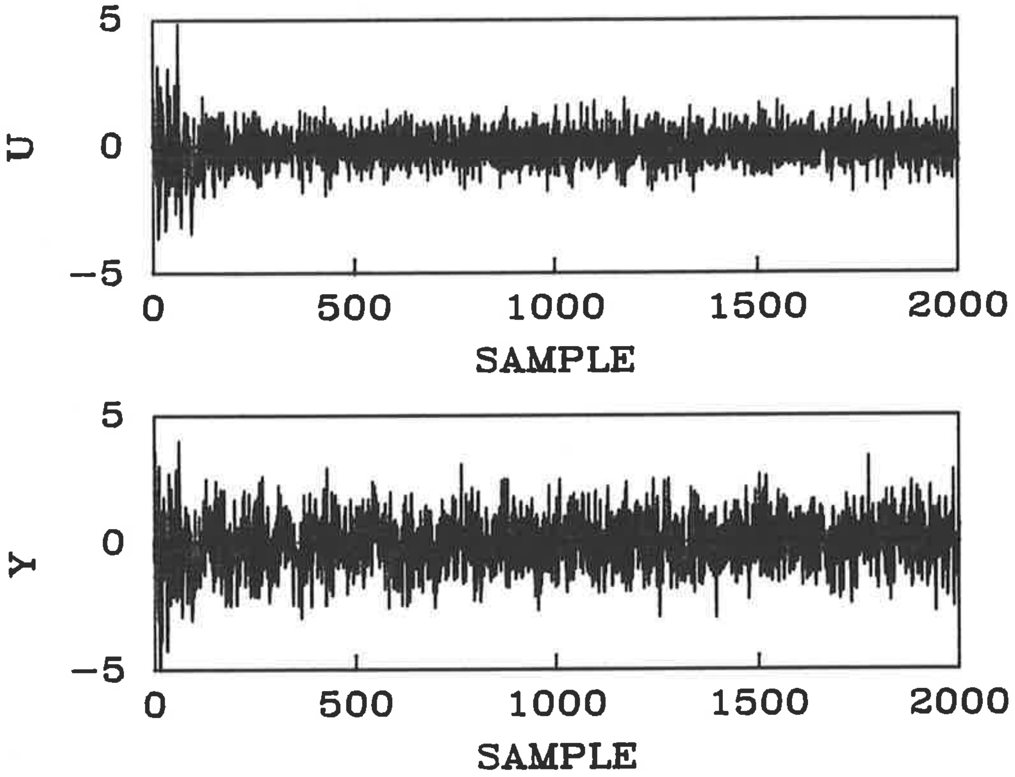

Figure 6.12: The sequences of the control input $u(t)$ and controlled output $y(t)$ resulting from the ST-MV-TISO-RELS strategy with $c_\omega(z^{-1})$ not satisfying the SPR condition (regulation).

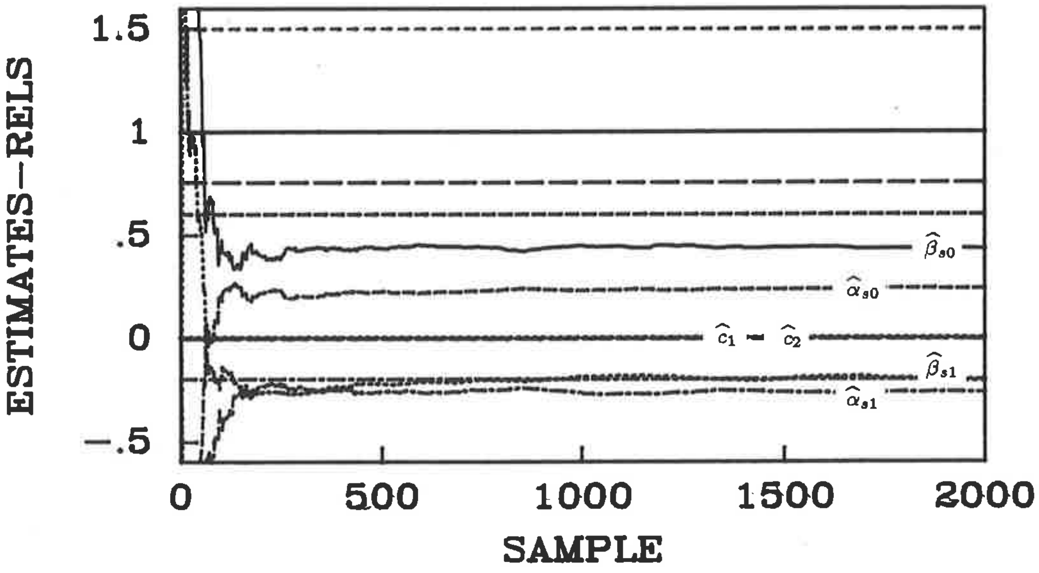

Figure 6.13: The parameter estimates and their true values for the ST-MV-TISO-RELS strategy with $c_\omega(z^{-1})$ not satisfying the SPR condition (regulation).

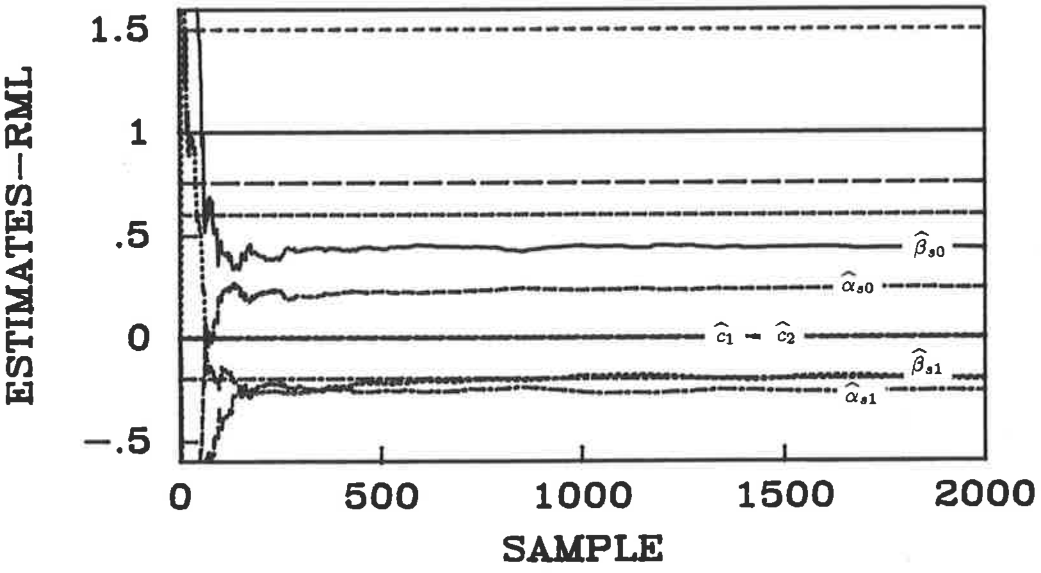

Figure 6.14: The parameter estimates and their true values for the ST-MV-TISO-RML strategy with $c_\omega(z^{-1})$ not satisfying the SPR condition (regulation).

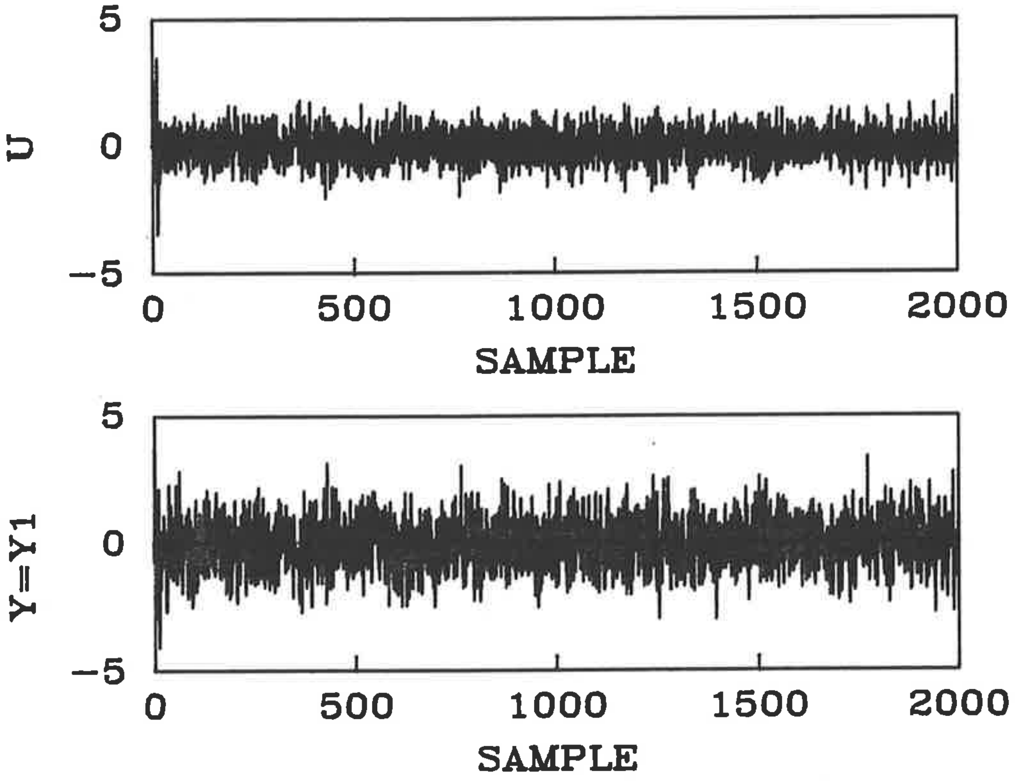

Figure 6.15: The sequences of the control input $u(t)$ and controlled output $y(t) = y_1(t)$ resulting from the ST-MV-TIMO strategy (without CNM) with $c_\omega(z^{-1})$ not satisfying the SPR condition (regulation).

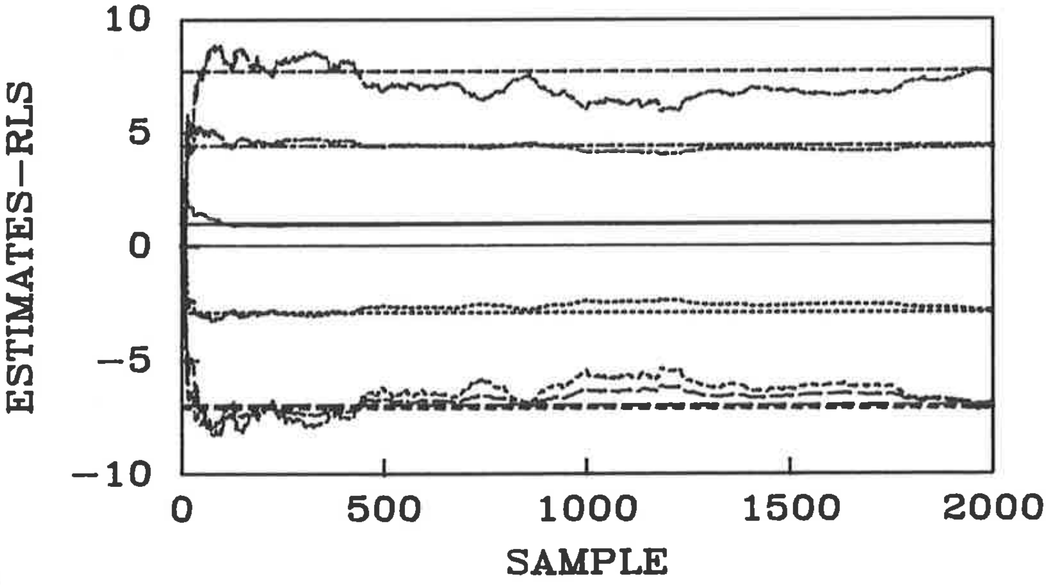

Figure 6.16: The parameter estimates and their true values for the ST-MV-TIMO strategy (without CNM) with $c_\omega(z^{-1})$ not satisfying the SPR condition (regulation).

with known parameters had been applied. Note that the estimates converged close to their true values after about 100 samples, thus overcoming the SPR condition. The estimates remained close to their true values for approximately 300 samples, after which they drifted around true values for the rest of the simulation.

It will be shown now that the drifting of estimates after the initial convergence phase can be effectively eliminated by the use of the CNM technique. For this purpose, the CNM was initialized with $\chi(-1) = 10^{-7}$ and $C_\chi = 10^{14}$. The variance of the controlled output resulting from the ST-MV-TIMO strategy with CNM was $\sigma_y^2 = 1.02$, i.e., it was not affected by the CNM. Furthermore, the estimate drift was eliminated, as shown in fig. 6.17. The initial convergence rate (first 100 samples) was not affected by the CNM, and the estimates remained close to their true values for the rest of simulation.

The estimate drift was eliminated by reducing the gain of the estimator (see eqn. E.8, Appendix E) with the scaling factor $\zeta(t) < 1$ (see eqn. E.10, Appendix E). The scaling factor is depicted in fig. 6.18. The scaling factor $\zeta(t)$ became less than one after 106 samples, i.e., after estimates converged close to true values (the corresponding value of the condition number of the matrix $P_m(t)$ was 2038). After initially small value of $\zeta(t)$ of about 0.95, it gradually increased towards 1, remaining at the

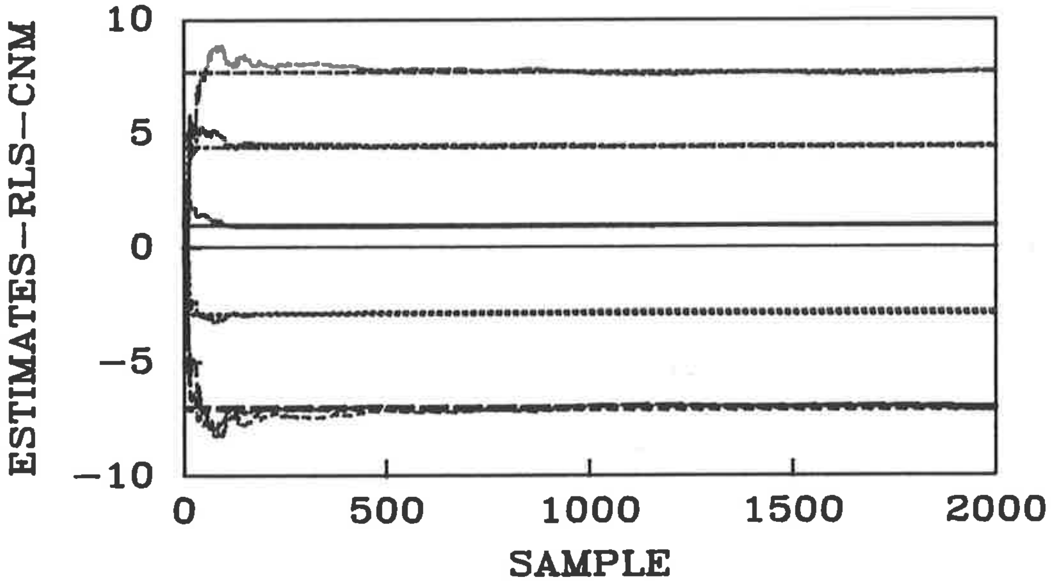

Figure 6.17: The parameter estimates and their true values for the ST-MV-TIMO strategy with CNM with $c_\omega(z^{-1})$ not satisfying the SPR condition (regulation).

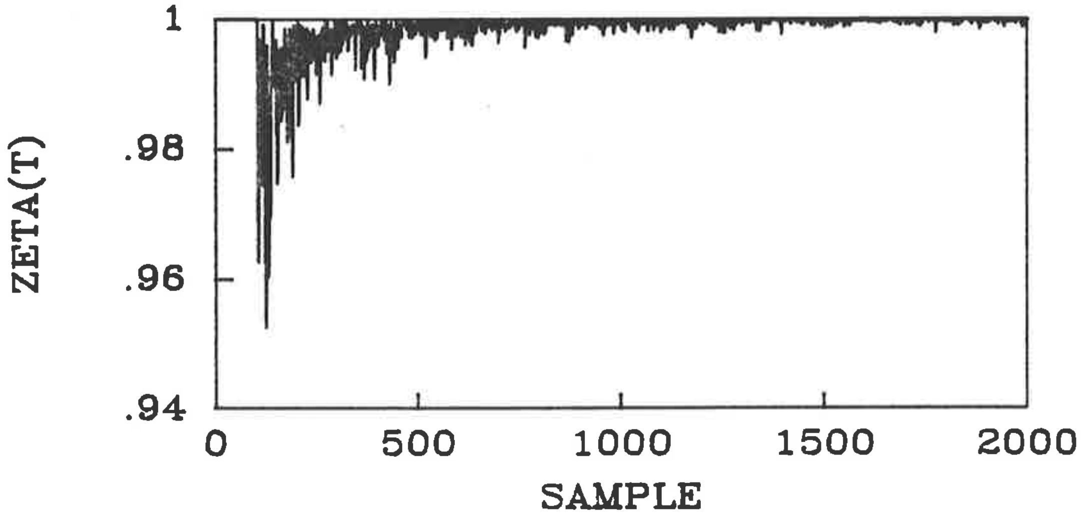

Figure 6.18: The scaling factor $\zeta(t)$ for the ST-MV-TIMO strategy with CNM with $c_\omega(z^{-1})$ not satisfying the SPR condition (regulation).

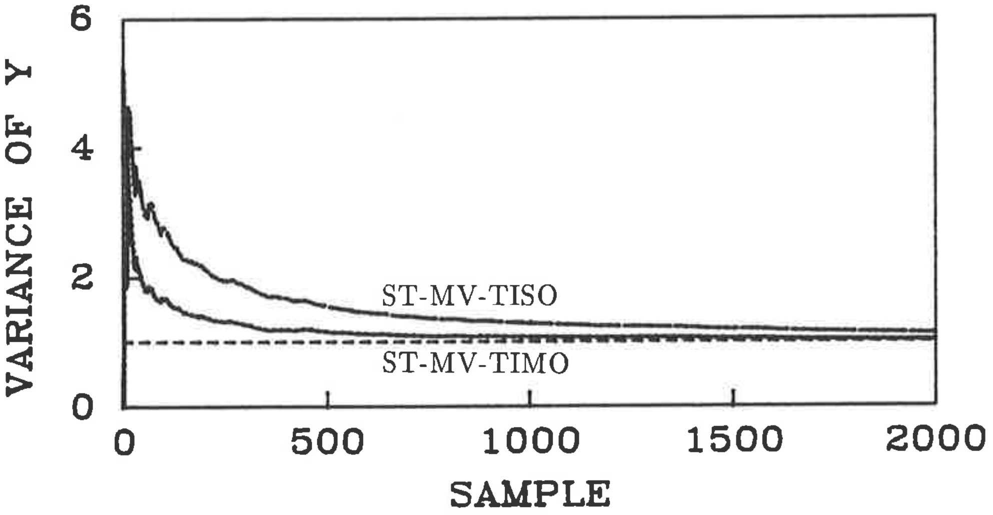

Figure 6.19: The on-line variance of the controlled output resulting from the ST-MV-TISO-RELS and -RML schemes (upper curve), and from the ST-MV-TIMO scheme without and with CNM (lower curve).

average value of 0.999 during last 1000 of iterations.

The performance of the simulated self-tuning strategies can be compared in terms of the on-line tracking error variance (i.e., in the considered case of regulation, the on-line variance of the controlled output), defined by eqn. 6.68 and depicted in fig. 6.19. Note the improvement in performance of self tuners due to utilization of additional outputs for feedback in the ST-MV-TIMO (without and with CNM) strategy, in comparison with both ST-MV-TISO-RELS and -RML schemes. There is also no degradation in performance due to the CNM technique employed in the ST-MV-TIMO algorithm. Although in this example the output variances tend to 1 for all self tuners, the convergence is guaranteed only for the ST-MV-TIMO with CNM scheme (see Theorem 6.2, p. 230). In general, the estimates for the ST-MV-TISO-RELS strategy may converge to biased values or may not converge at all (since the noise polynomial does not satisfy the SPR condition), possibly leading to an unstable closed-loop system. The convergence of the ST-MV-TISO-RML strategy is not guaranteed due to lack of the appropriate projection method for the estimate of the noise polynomial into the stability region (see p. 325, Appendix G).

The condition number of the estimate error covariance matrix $P(t)$ must be

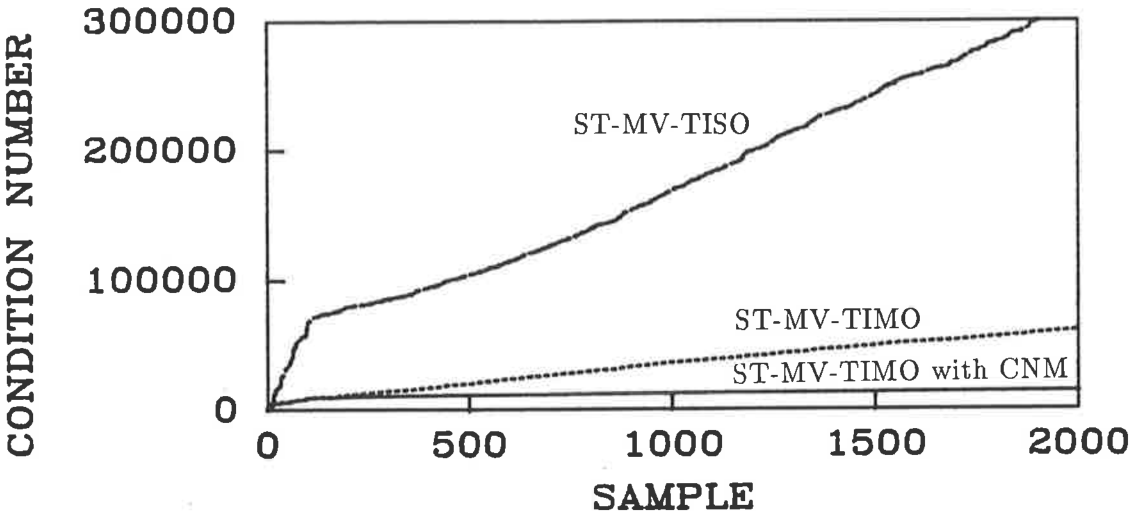

Figure 6.20: The condition number of $P(t)$ for the ST-MV-TISO strategy with the RELS and RML estimators (upper curve), and for the ST-MV-TIMO strategy without CNM (middle curve) and with CNM (lower curve) with $c_\omega(z^{-1})$ not satisfying the SPR condition (regulation).

bounded in order to avoid numerical instability of the estimator. The evolution in time of the condition number of $P(t)$ for all simulated strategies is shown in fig. 6.20 (upper curve for both ST-MV-TISO-RELS and -RML strategies, middle curve for the ST-MV-TIMO without CNM strategy, and lower curve for the ST-MV-TIMO with CNM scheme). It can be seen that the condition number grows rapidly for both ST-MV-TISO strategies. However, for the ST-MV-TIMO scheme without CNM, the condition number increases at a much slower rate than for both ST-MV-TISO strategies. This is due to the use of the linear regression estimator in the ST-MV-TIMO scheme instead of the pseudo-linear regression estimator (see Comment 6.9, p. 234). Further improvement is observed for the ST-MV-TIMO scheme with CNM, for which the condition number increases at a very slow rate.

It can be seen in this example that the CNM technique eliminates drifts of estimates and the rapid growth of the condition number of the covariance matrix, while not affecting the closed-loop system performance. The latter feature is achieved by a suitable choice of the initial values of the parameters of the CNM technique. Although $\chi(-1) = 10^{-7}$ and $C_\chi = 10^{14}$ were selected in this example, the choice of

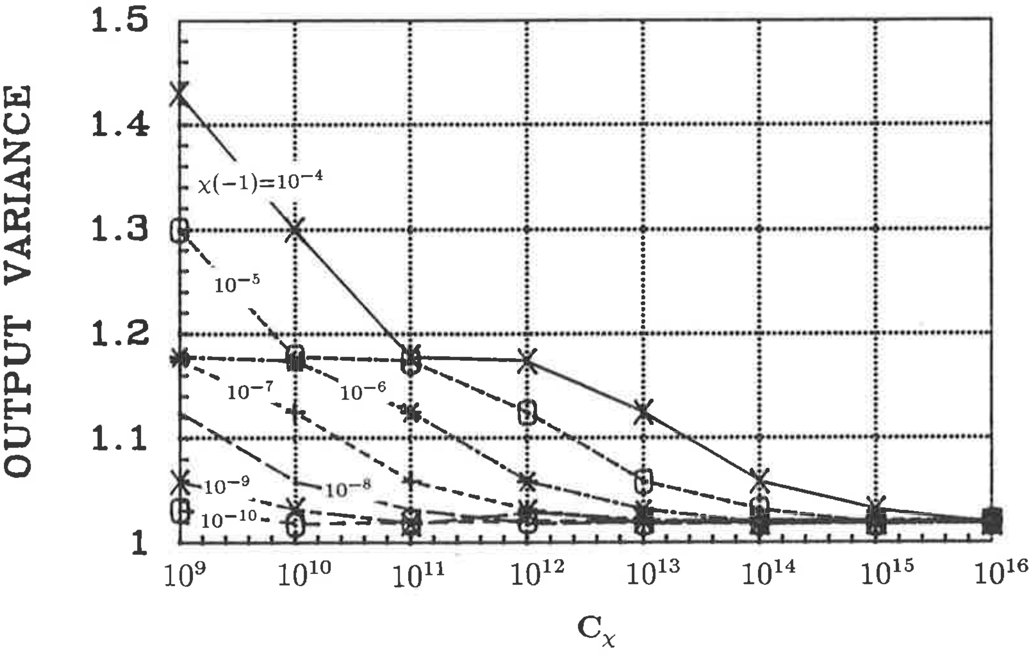

Figure 6.21: The controlled output variance vs. C_χ with $\chi(-1)$ as parameter for the ST-MV-TIMO strategy with CNM with $c_\omega(z^{-1})$ not satisfying the SPR condition (regulation).

these constants is somewhat arbitrary, as pointed out in [69]. In order to assess the influence of the choice of $\chi(-1)$ and C_χ on the closed-loop system performance, the variance of the controlled output was calculated from data obtained with the ST-MV-TIMO strategy for various choices of the CNM initial parameters: $\chi(-1)$ in the range from 10^{-4} to 10^{-10}, and C_χ in the range from 10^9 to 10^{16}. The results are shown in fig. 6.21 (output variance versus C_χ, with $\chi(-1)$ as a parameter). The CNM technique was implemented with the test based on the trace of $P_m(t)$ rather than on the maximum eigenvalue of $P_m(t)$ (see Comment 6.5, p. 230). Note that the closed-loop system performance can be significantly affected by an improper choice of $\chi(-1)$ and C_χ. Furthermore, in this example for lower values of $\chi(-1)$ the output variance is more or less independent of C_χ.

Tracking problem.

Let us now consider the reference tracking problem. The reference sequence was a square wave with unit amplitude and period of 20 samples. The input noise variance was set to $\sigma^2 = 0.01$.

In the regulation problem it was found that the use of the ST-MV-TISO-RML strategy did not improve the closed-loop system performance in comparison with the ST-MV-TISO-RELS strategy. Moreover, the trajectories of parameter estimates were very similar for both strategies. In the tracking case it was found out that in order to highlight possible difference in the estimation for both schemes, a forgetting factor $\alpha(t) = 0.995$ can be used [12]. The remaining initial parameters of the estimator were set as for the regulation problem.

In the case of tracking with the ST-MV-TISO-RELS strategy the noise polynomial must be estimated, unlike in the regulation problem (see cases (i) and (ii), p. 233). The tracking error variance (see eqn. 6.67) resulting from the ST-MV-TISO-RELS strategy was $\sigma^2_{e_r} = 1.49\ 10^{-2}$. The corresponding estimates are depicted in fig. 6.22. Note that only estimates of parameters β_{s0} and β_{s1} converged to the true values (see eqn. 6.74).

Next the ST-MV-TISO-RML strategy was simulated. The tracking error variance was $\sigma^2_{e_r} = 1.314\ 10^{-2}$, i.e., it was smaller than for the ST-MV-TISO-RELS scheme. The estimates converged close to true values after about 1000 samples (see fig. 6.23). This illustrates that the RML estimator employed for self-tuning control *may* overcome the SPR condition.

Finally, the ST-MV-TIMO strategy was applied. Again, schemes without and with CNM were simulated. For the case without CNM, the tracking error variance was $\sigma^2_{e_r} = 1.115\ 10^{-2}$, i.e., it was smaller than for both ST-MV-TISO strategies. The estimates converged close to their true values after about 500 samples (see fig. 6.24). This illustrates the superior convergence rate of the ordinary linear regression estima-

[12] In the regulation case described above, the forgetting factor $\alpha(t) = 0.995$ did not improve convergence rate of the RML scheme, and led to the divergence of estimates ("burst phenomenon") for the RELS scheme.

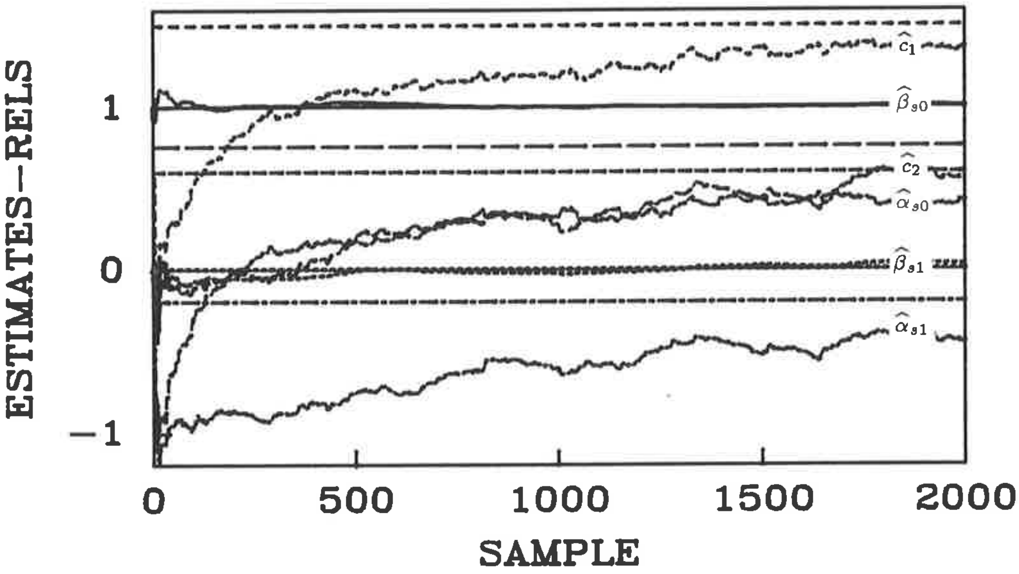

Figure 6.22: The parameter estimates and their true values for the ST-MV-TISO-RELS strategy with $c_\omega(z^{-1})$ not satisfying the SPR condition (tracking).

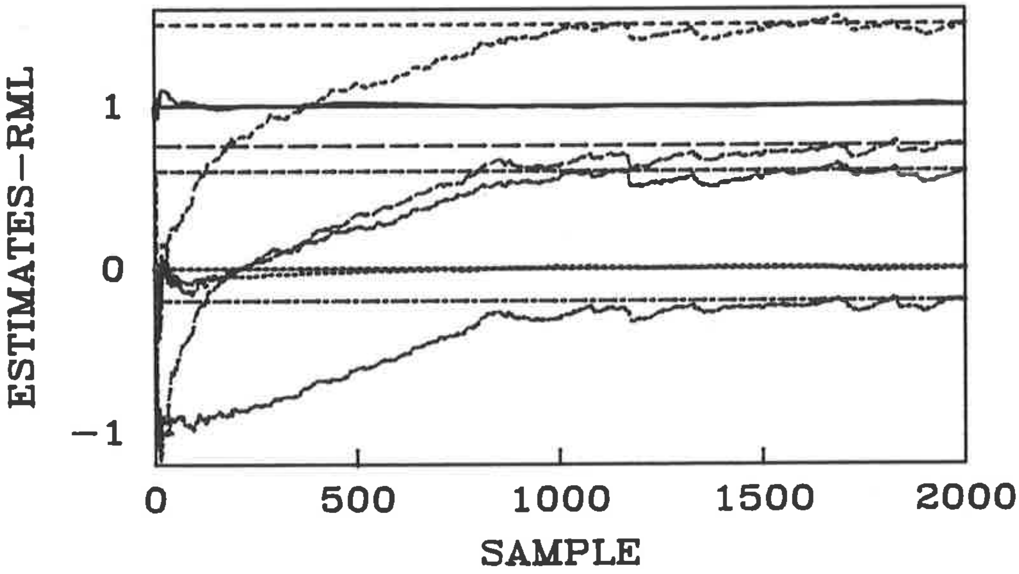

Figure 6.23: The parameter estimates and their true values for the ST-MV-TISO-RML strategy with $c_\omega(z^{-1})$ not satisfying the SPR condition (tracking).

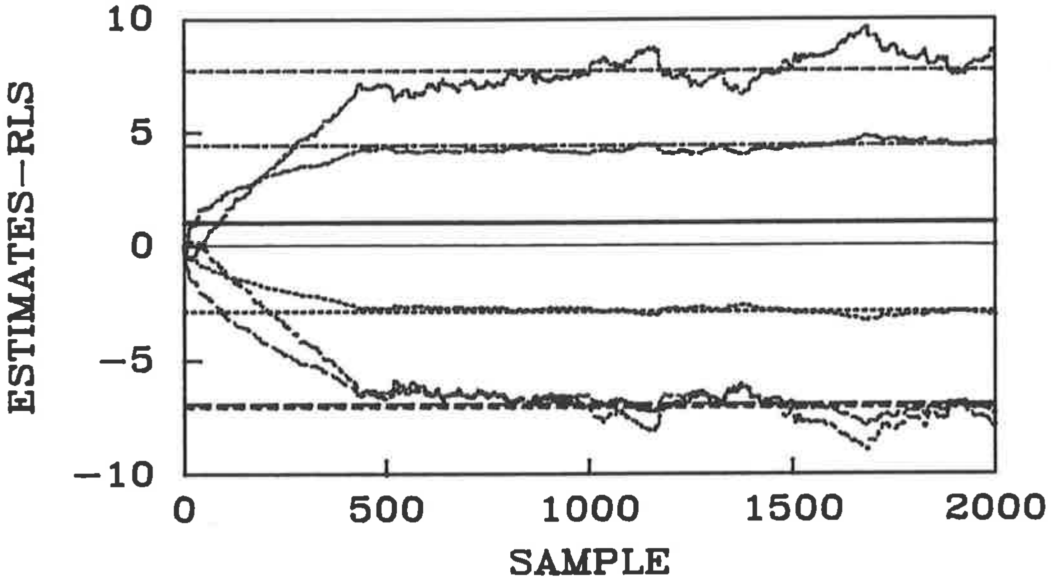

Figure 6.24: The parameter estimates and their true values for the ST-MV-TIMO strategy (without CNM) with $c_\omega(z^{-1})$ not satisfying the SPR condition (tracking).

tor over the pseudo-linear regression estimators. In fact, it was found by simulation studies not reported here that an even more significant improvement in convergence rate can be expected in comparison with the ST-MV-TISO-RELS strategy (applied for systems satisfying the SPR condition).

Certain fluctuations of parameter estimates for the ST-MV-TIMO strategy without CNM can be observed after about 1000 samples (see fig. 6.24). These fluctuations can be effectively eliminated by the use of the CNM technique, as shown in fig. 6.25. The CNM was initialized with $\chi(-1) = 10^{-10}$ and $C_\chi = 10^{18}$. After the rapid initial convergence of estimates during first 500 samples, a period of smooth convergence followed, due to the CNM which was invoked after about 600 samples. The CNM technique did not affect the tracking error variance which was the same as without CNM.

For the comparison of performance of all strategies, the cumulative loss functions (see eqn. 6.69) are depicted in fig. 6.26. It may be observed that the cumulative loss increases more rapidly for the ST-MV-TISO-RELS scheme than for the remaining methods. The ST-MV-TISO-RML strategy, which overcomes the SPR condition in this simulation study, leads to optimal performance (in contrast to the ST-MV-TISO-RELS scheme) but at a much slower rate than both ST-MV-TIMO schemes.

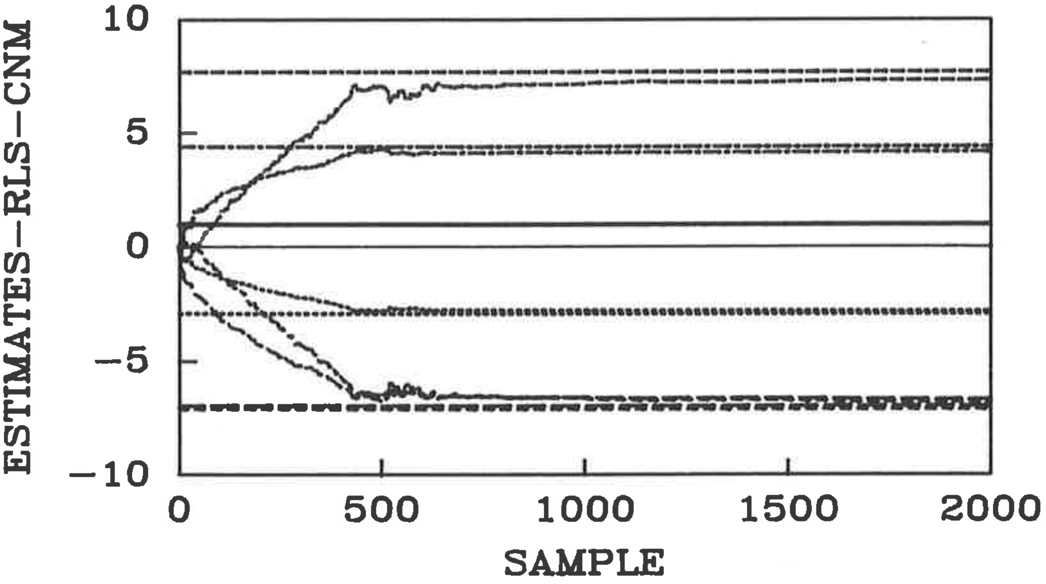

Figure 6.25: The parameter estimates and their true values for the ST-MV-TIMO strategy (with CNM) with $c_\omega(z^{-1})$ not satisfying the SPR condition (tracking).

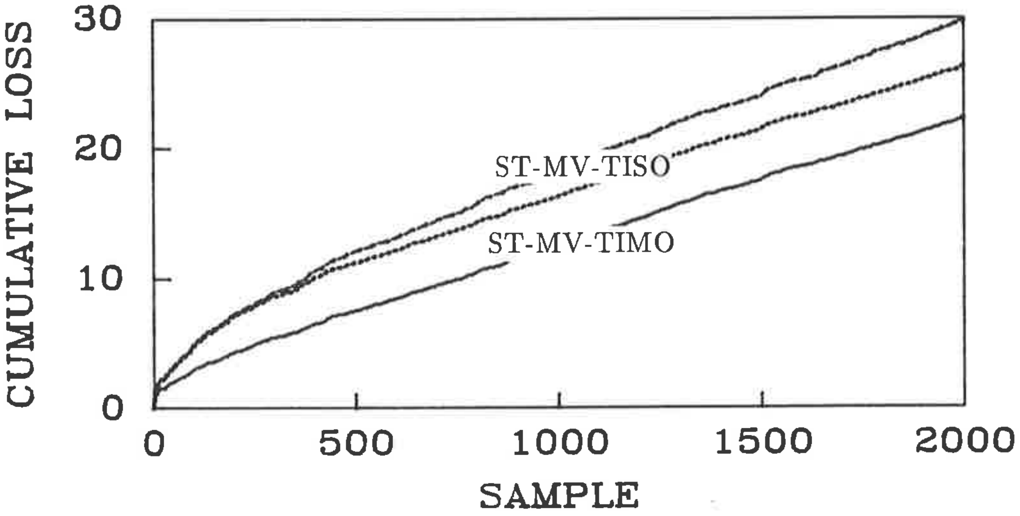

Figure 6.26: The cumulative loss functions resulting from the ST-MV-TISO-RELS strategy (upper curve), ST-MV-TISO-RML strategy (middle curve), and from the ST-MV-TIMO strategies without and with CNM (lower curve) with $c_\omega(z^{-1})$ not satisfying the SPR condition (tracking).

Let us now summarize the results of the above simulation studies in which the noise polynomial does not satisfy the SPR condition:

- the parameter estimates of the ST-MV-TISO-RELS strategy failed to converge to parameters of the optimal control law;
- the parameter estimates of the ST-MV-TISO-RML strategy converged to parameters of the optimal control law for the tracking problem but not for the regulation problem; in the regulation problem the performance of the ST-MV-TISO-RML scheme was as for the ST-MV-TISO-RELS scheme;
- the parameter estimates of the ST-MV-TIMO strategy converged to the parameters of the optimal control law for both regulation and tracking problems with the convergence rate significantly superior to that of the ST-MV-TISO-RML scheme (when the latter strategy converged);
- the CNM technique did not affect the performance of the ST-MV-TIMO strategy (i.e., the minimization of the tracking error variance) provided the initial values of parameters $\chi(-1)$ and C_χ of the CNM were suitably chosen;
- the CNM technique eliminated undesirable drifts of the parameter estimates observed after the estimates converged close to their true values when the CNM was not employed;
- the numerical robustness of the estimator was enhanced firstly by using the linear regression estimator instead of the pseudo-linear regression estimator, and secondly by the CNM technique.

Example 6.2.3. System with a noise polynomial $c_\omega(z^{-1})$ having roots on the unit circle.

The purpose of this example is to demonstrate that the ST-MV-TIMO strategy can cope with systems with the noise polynomial $c_\omega(z^{-1})$ having roots on the unit circle without sacrificing its asymptotic optimality (see discussion on pp. 222-224). For the purpose of comparison the ST-MV-TISO strategy based on the restricted complexity (suboptimal) predictor will be considered for the corresponding TISO system.

Let us consider a second order ($n = 2$) system given by the state-space model 2.8 (p. 22), where $y(t) = y_1(t)$ and $y_F(t) = [y_1(t)\ y_2(t)]'$, and

$$A_s = \begin{bmatrix} 1.5 & -0.95 \\ 1 & 0 \end{bmatrix}, \qquad B_s = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad K = \begin{bmatrix} 3.0641 \\ 0.8718 \end{bmatrix},$$
$$C = \begin{bmatrix} 1 & 0.5 \end{bmatrix}, \qquad C_F = \begin{bmatrix} 1 & 0.5 \\ 0.3 & -0.1 \end{bmatrix}, \qquad C_S = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

The observability index of the pair (C_F, A_s) is $\nu_F = 1$. The corresponding RDO representation is given by (see eqns. 3.89 to 3.94, p. 92)

$$\begin{aligned} d(p) &= [1\ \ -1.5\ \ 0.95]\, P_2, \\ N(p) &= [0\ \ 1\ \ 0.5]\, P_2, \\ c_\omega(p) &= [1\ \ 2\ \ 1]\, P_2, \\ N_F(p) &= \begin{bmatrix} 0 & 1 & 0.5 \\ 0 & 0.3 & -0.1 \end{bmatrix} P_2, \\ C_\omega(p) &= \begin{bmatrix} 1 & 2 & 1 \\ 1 & -0.6679 & 0.5259 \end{bmatrix} P_2. \end{aligned} \tag{6.75}$$

The delay $k = 1$. Note that the noise polynomial $c_\omega(z^{-1})$ has a double root on the unit circle at $z = -1$.

The problem of the self-tuning MV control of the output $y(t) = y_1(t)$ is considered in this example.

For this purpose one can implement the ST-MV-TIMO strategy (feedback from outputs $y_F(t) = [y_1(t)\ y_2(t)]'$) which involves estimation of coefficients of the 1×2 polynomial vector $\alpha_\omega(p)$ and polynomial $\beta_\omega(p)$ (see eqns. 6.56 (p. 229) and 3.97, 3.98 (p. 94)). Assuming knowledge of the observability index ν_F, the degree of the polynomials is chosen as $n_p = 1$ according to eqn. 3.133 (p. 105). The coefficients of the polynomials $\alpha_\omega(p)$ and $\beta_\omega(p)$, calculated from eqn. 3.113 (p. 99) for $k_p = k = 1$, are given by (cf. example 3.3.1, p. 105)

$$\alpha_\omega \;=\; [\alpha_{10}\ \alpha_{20}\ \alpha_{11}\ \alpha_{21}] = [0.6719\ \ 2.8281\ \ -0.6603\ \ 1.2555], \tag{6.76}$$

$$\beta_\omega = [\beta_0 \ \beta_1] = [1 \ 0.4797]. \tag{6.77}$$

On the other hand, if the only output which is available for feedback is the controlled output (i.e., the system has the feedback configuration FI), then the RDO model of the corresponding TISO system is given by eqn. 3.89 (p. 92) with $y_F(t) = y(t) = y_1(t)$. Note that the model of this system can be alternatively written in the following ARMAX form

$$d(p)y(t) = N(p)u(t) + c_\omega(p)\omega(t). \tag{6.78}$$

The ST-MV-TISO strategy for systems with the noise polynomial $c_\omega(z^{-1})$ having roots on the unit circle proposed in [9, p. 450] will be considered. For this purpose, the polynomial $c_\omega(z^{-1})$ is approximated by a first order, stable polynomial $c_A(z^{-1})$ (see [9, example 11.3.1, p. 450, and pp. 272-274])

$$c_A(z^{-1}) = 1 + c_a z^{-1}.$$

The estimate $\widehat{c}_a$ of the coefficient c_a is restricted to $|\widehat{c}_a| \leq c_r < 1$. If $\widehat{c}_a > c_r$ then the estimate value is set to $\widehat{c}_a = c_r$, and if $\widehat{c}_a < c_r$ then it is set to $c_a = -c_r$.

We shall consider two control tasks: regulation problem ($r(t) = 0$) and tracking the reference sequence.

Regulation problem.

The input noise variance was set to $\sigma^2 = 1$. Since $k = 1$, the minimal variance of the controlled output is $\sigma_y^2 = 1$.

The estimators were initialized with $P(-1) = 100I$, the initial parameter estimates were set to zero except for $\widehat{\beta}_0(0) = \widehat{\beta}_{s0}(0) = 1$.

The choice of the initial values of the parameters of the CNM technique for the ST-MV-TIMO strategy was $\chi(-1) = 10^{-10}$ and $C_\chi = 10^{18}$ (trace of the covariance matrix was used in the test in eqn. E.7 (Appendix E) rather then covariance matrix maximum eigenvalue, see Comment 6.5, p. 230). No forgetting was used, i.e., $\alpha(t) = 1$. The control signal magnitude was limited to $|u(t)| \leq 10$. The sequences of the control input $u(t)$, the controlled output $y(t) = y_1(t)$, and the second feedback output $y_2(t)$ are shown in fig. 6.27.

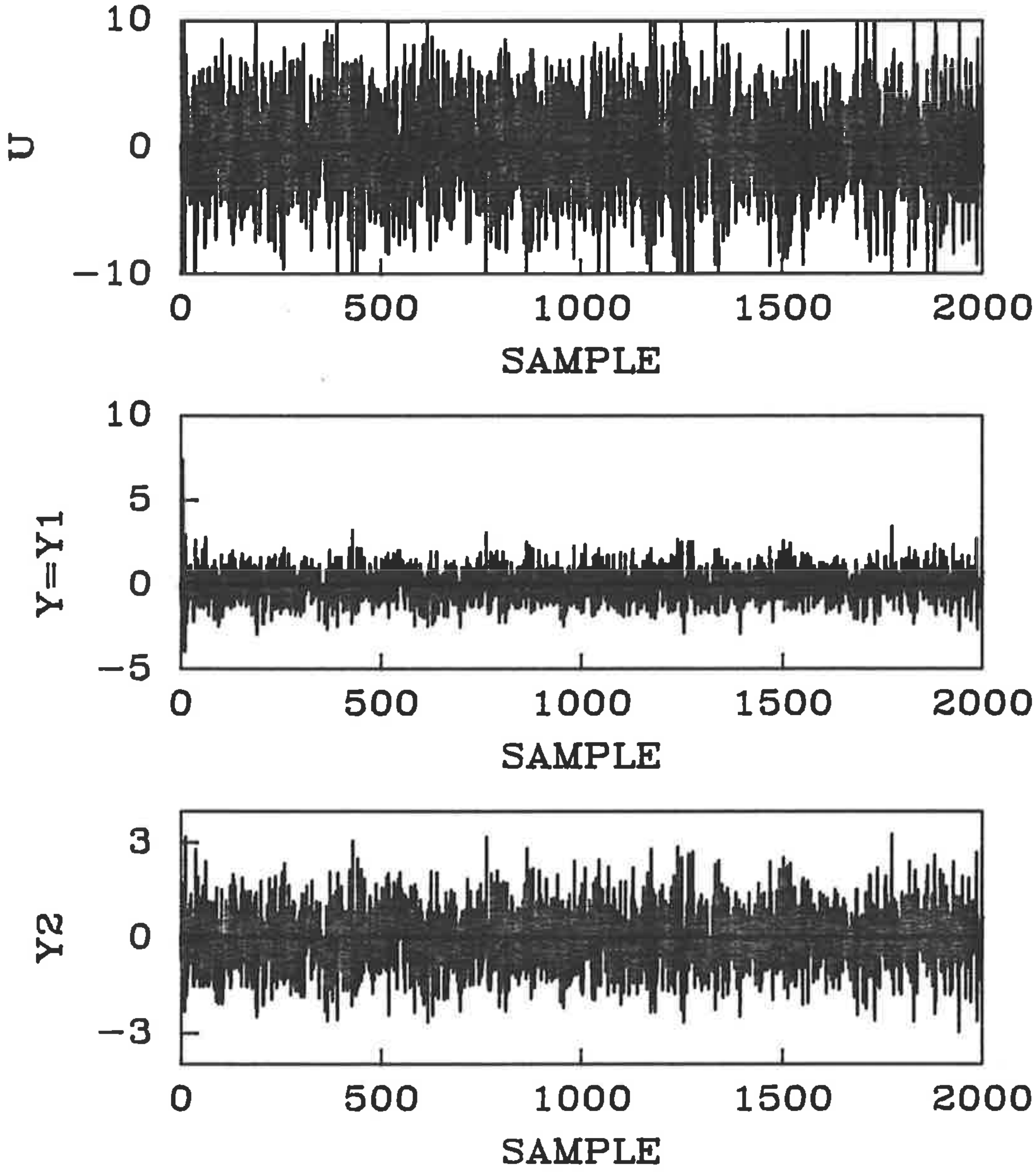

Figure 6.27: The sequences of the control input $u(t)$, controlled output $y(t) = y_1(t)$, and the second feedback output $y_2(t)$ resulting from the ST-MV-TIMO strategy with $c_\omega(z^{-1})$ having roots on the unit circle (regulation).

The variance of the controlled output (see eqn. 6.67 for $N = 2000$) was $\sigma_y^2 = 1.1$. The estimates of the parameters given by eqns. 6.76 and 6.77 are depicted in fig. 6.28.

For the purpose of comparison the (suboptimal) ST-MV-TISO strategy [9, p. 450]

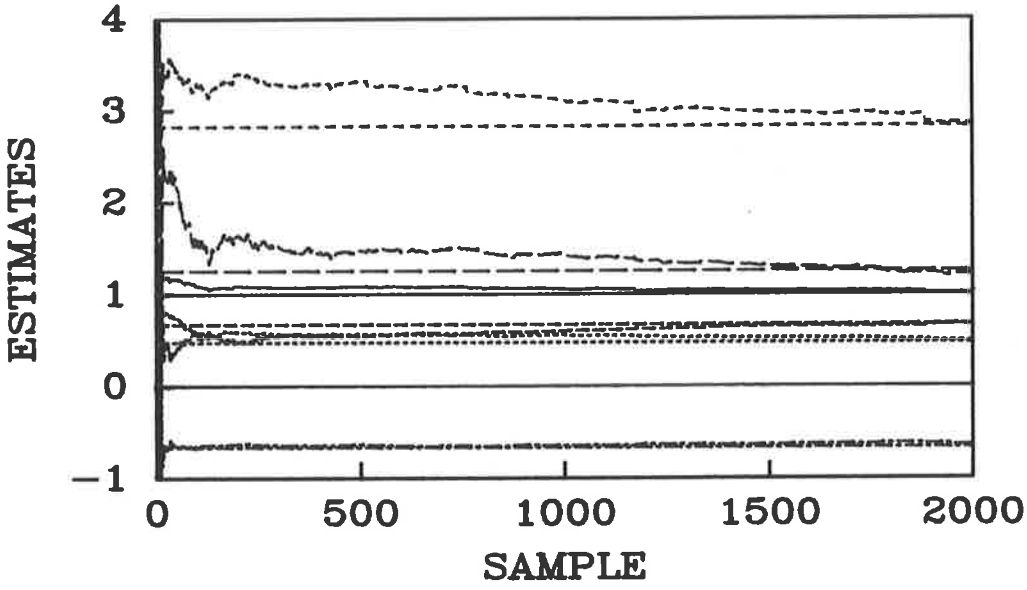

Figure 6.28: The parameter estimates and their true values for the ST-MV-TIMO strategy with $c_{\omega}(z^{-1})$ having roots on the unit circle (regulation).

was simulated. In order to observe the influence of the choice of the bound c_r on the closed-loop system performance, the simulation runs were repeated for the values of c_r from 0.4 to 0.95 with increment 0.05. For each value of c_r, four values of forgetting factor were chosen $\alpha(t) = 1$; 0.998; 0.996; and 0.994. The results are shown in fig. 6.29, where the variance of the regulated output $y(t)$ is depicted versus bound c_r with forgetting factor $\alpha(t)$ as a parameter. It may be observed that the influence of the bound c_r on the closed-loop system performance was insignificant for the values of the forgetting factor $\alpha(t) \geq 0.996$. For the value $\alpha = 0.994$ the performance deteriorated and the increase of output variance is observed.

In comparison with the closed-loop system performance resulting from the ST-MV-TIMO strategy (recall that $\sigma_y^2 = 1.1$ for the ST-MV-TIMO scheme), fig. 6.29 shows a significant increase of the minimized variance resulting from the ST-MV-TISO scheme.

Tracking problem.

Let us now consider the tracking problem. The reference sequence was a square wave lying between ±1 and with period 100 samples. The noise variance was set to $\sigma^2 = 0.01$. The estimators were initialized with $P(-1) = 100I$, the initial parameter

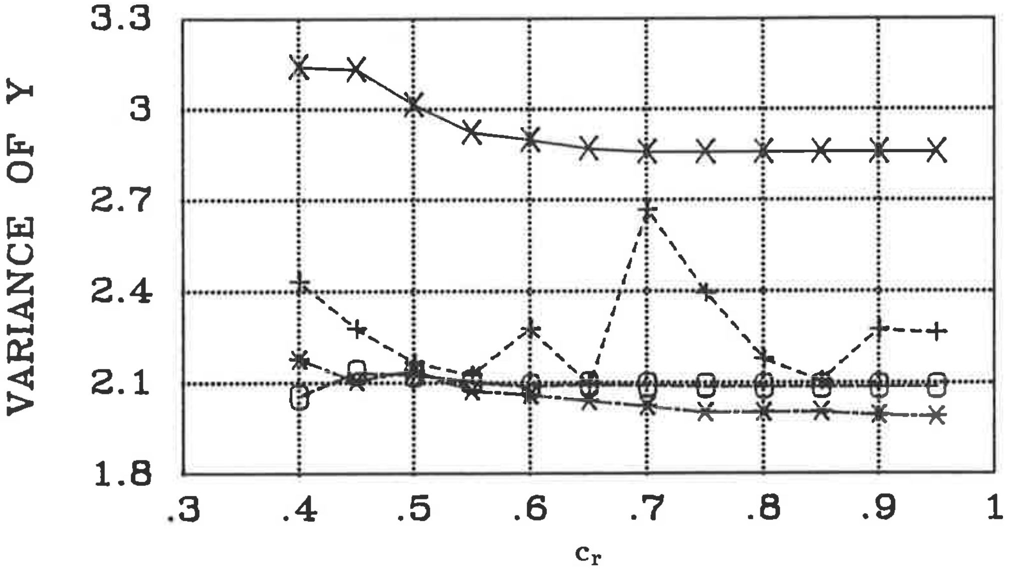

Figure 6.29: The controlled output variance vs. bound c_r for the ST-MV-TISO strategy based on the restricted complexity predictor (regulation): $\alpha(t) = 1$ (symbol "x"), 0.998 ("o"), 0.996 ("*"), and 0.994 ("+").

estimates were set to zero except for $\widehat{\beta}_0(0) = \widehat{\beta}_{s0}(0) = 1$.

The choice of the initial values of the parameters of the CNM technique of the ST-MV-TIMO strategy was $\chi(-1) = 10^{-8}$ and $C_\chi = 10^{10}$ (trace of the covariance matrix was used in the test in eqn. E.7, Appendix E). No forgetting was used, i.e., $\alpha(t) = 1$. The control signal magnitude was limited to $|u(t)| \leq 10$. The sequences of the control input $u(t)$, the controlled output $y(t) = y_1(t)$ and reference $r(t)$, and the second feedback output $y_2(t)$ are shown in fig. 6.30. The variance of the tracking error (see eqn. 6.67 for $N = 500$) was $\sigma^2_{e_r} = 1.77\ 10^{-2}$. The estimates of the parameters given by eqns. 6.76 and 6.77 are depicted in fig. 6.31.

For the purpose of comparison the (suboptimal) ST-MV-TISO strategy [9, p. 450] was simulated with the bound on the estimate of coefficient c_a of the polynomial $c_A(z^{-1})$ set to $c_r = 0.95$. In order to improve the convergence rate, the forgetting factor was set to $\alpha(t) = 0.99$. The control signal magnitude was limited to $|u(t)| \leq 15$.

The sequences of the control input $u(t)$, the controlled output $y(t)$ and reference $r(t)$ are shown in fig. 6.32. The variance of the tracking error was $\sigma^2_{e_r} = 2.73\ 10^{-2}$, i.e., greater than for the ST-MV-TIMO scheme (the difference in performance is actually

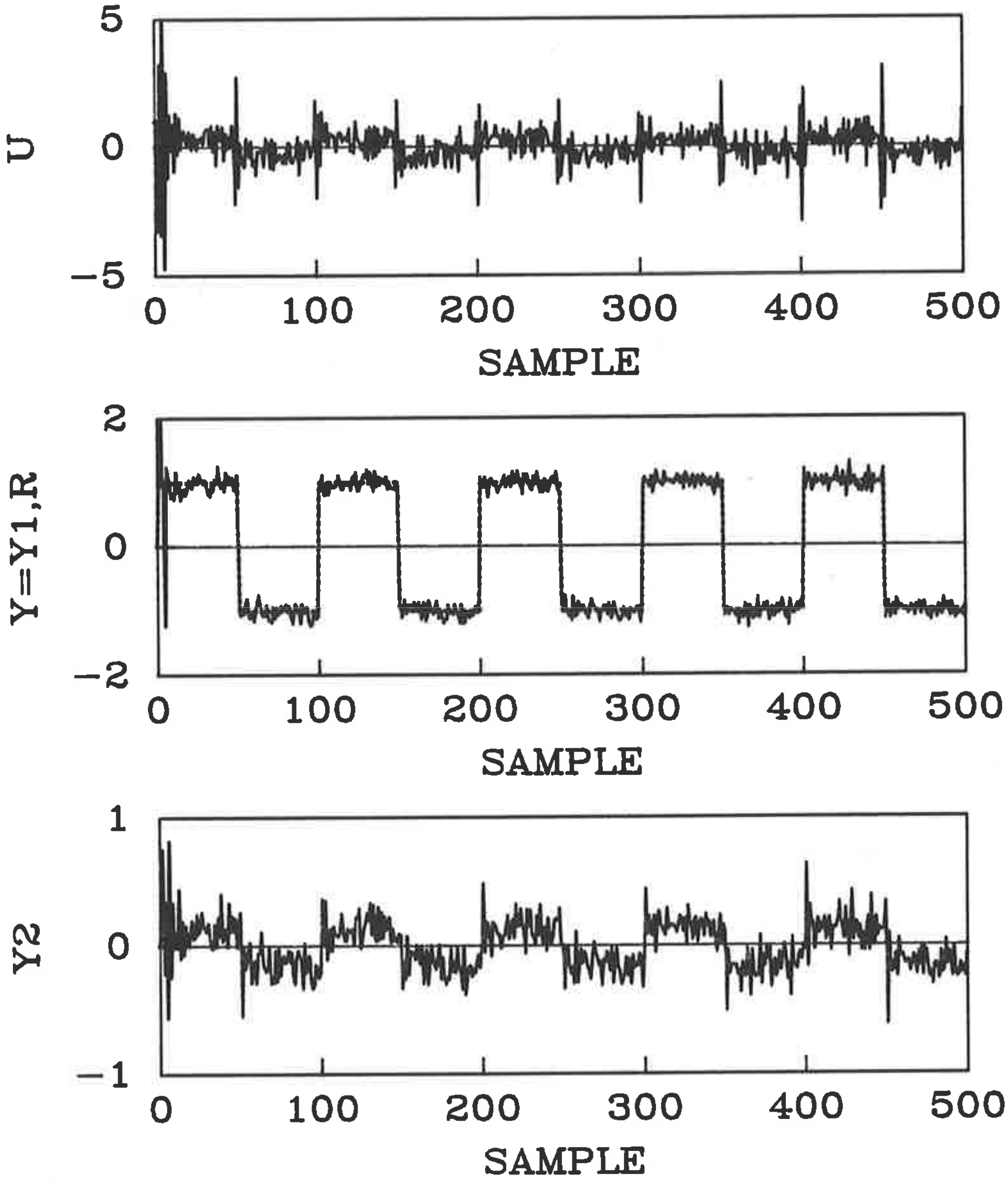

Figure 6.30: The sequences of the control input $u(t)$, controlled output $y(t) = y_1(t)$ and reference $r(t)$, and the second feedback output $y_2(t)$ resulting from the ST-MV-TIMO strategy with $c_\omega(z^{-1})$ having roots on the unit circle (tracking).

visible in output sequences, see figs. 6.30 and 6.32).

The polynomials $\alpha_{s\omega}(p)$ and $\beta_{s\omega}(p)$, which are estimated in the ST-MV-TISO scheme, are of the form [13] $\alpha_{s\omega}(p) = [\alpha_{s0} \;\; \alpha_{s1}]\, P_1$, and $\beta_{s\omega}(p) = [\beta_{s0} \;\; \beta_{s1}]\, P_1$. The

[13] For the system with known parameters and for the given approximating polynomial $c_A(p)$, poly-

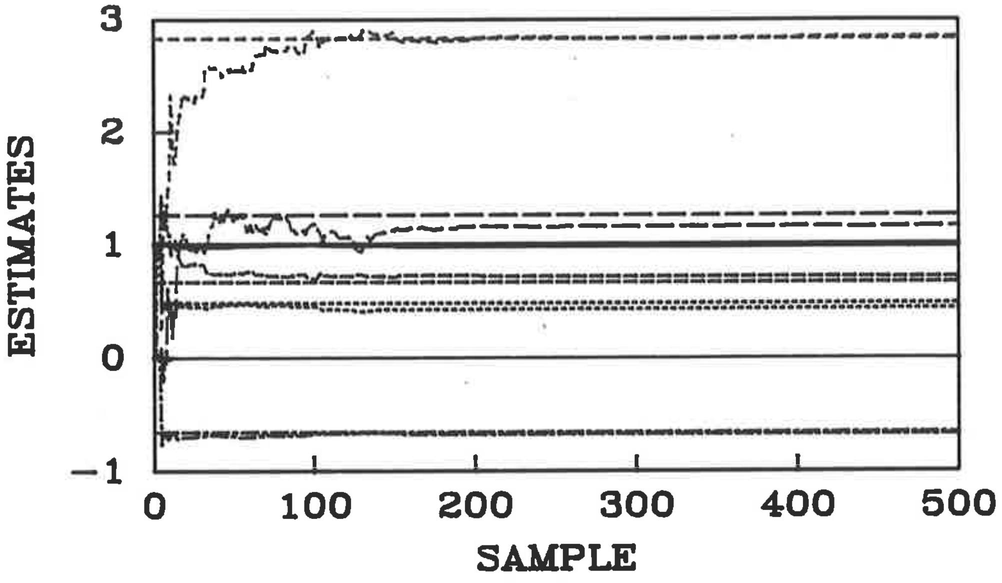

Figure 6.31: The parameter estimates and their true values for the ST-MV-TIMO strategy with $c_\omega(z^{-1})$ having roots on the unit circle (tracking).

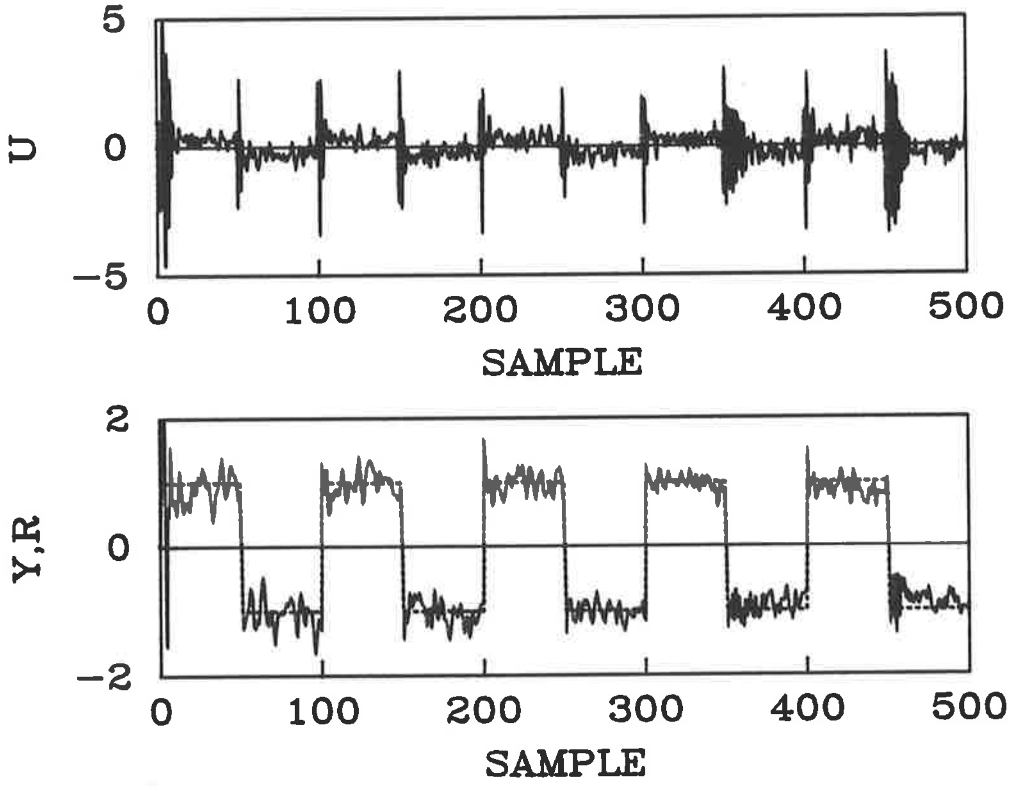

Figure 6.32: The sequences of the control input $u(t)$, controlled output $y(t)$ and reference $r(t)$ resulting from the ST-MV-TISO strategy based on the restricted complexity predictor applied to a system with $c_\omega(z^{-1})$ having roots on the unit circle (tracking).

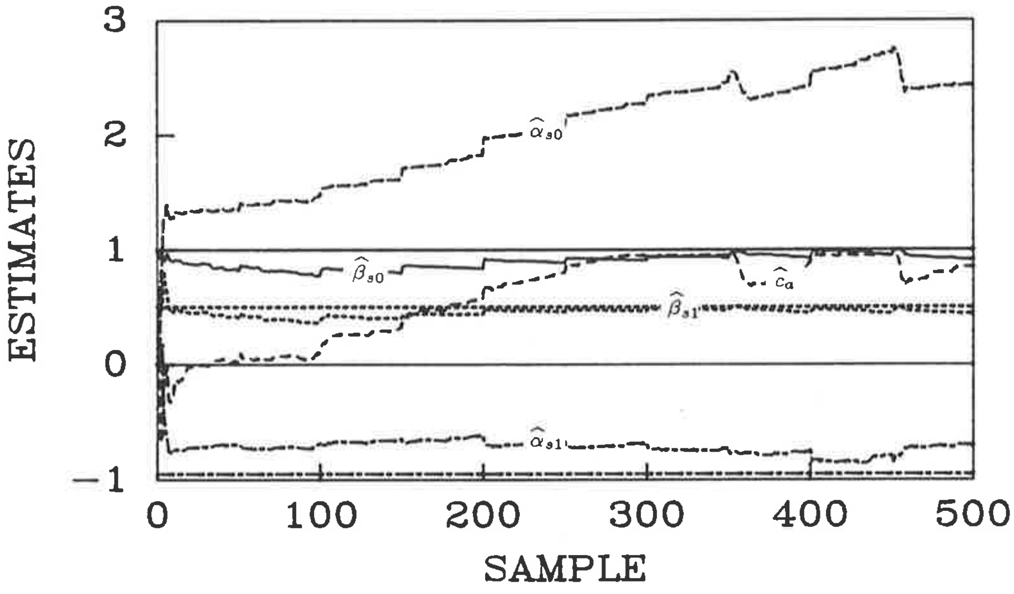

Figure 6.33: The parameter estimates for the ST-MV-TISO strategy based on the restricted complexity predictor applied to a system with $c_\omega(z^{-1})$ having roots on the unit circle (tracking).

estimates of $\alpha_{s\omega}(p)$, $\beta_{s\omega}(p)$, and $c_A(p)$ are shown in fig. 6.33. Note that the estimate $\widehat{c}_a$ approaches its bound $c_r = 0.95$, thus approximating the true noise polynomial with double root at $z = -1$ by the first order (stable) polynomial with root at $z = -0.95$.

The performance of both self-tuning controllers can be compared using the cumulative loss functions (see eqn. 6.69) depicted in fig. 6.34.

Finally, in order to compare the performance of the ST-MV-TIMO and ST-MV-TISO strategies under various conditions, the simulation runs were repeated for the input noise variance $\sigma^2 = 0.0001$; 0.0025; 0.01; 0.04; 0.09; and 0.16. In the case of the ST-MV-TISO strategy the forgetting factor was set to $\alpha(t) = 1$; 0.995; and 0.99 for each value of σ^2. For the ST-MV-TIMO strategy the forgetting factor was $\alpha(t) = 1$. The results are shown in fig. 6.35, where the tracking error variance is depicted versus the input noise variance σ^2 with the forgetting factor as a parameter. It can be seen that the ST-MV-TIMO scheme outperformed the ST-MV-TISO strategy regardless of the noise variance and forgetting factor used to improve the convergence rate of

nomials $\alpha_{s\omega}(p)$ and $\beta_{s\omega}(p)$ which define the MV controller of [9, Theorem 10.3.3, p. 416] for systems with the noise polynomial having roots on the unit circle, result from eqns. 3.138 and 3.139 (p. 108) with $c_A(p)$ substituted for $c_\omega(p)$.

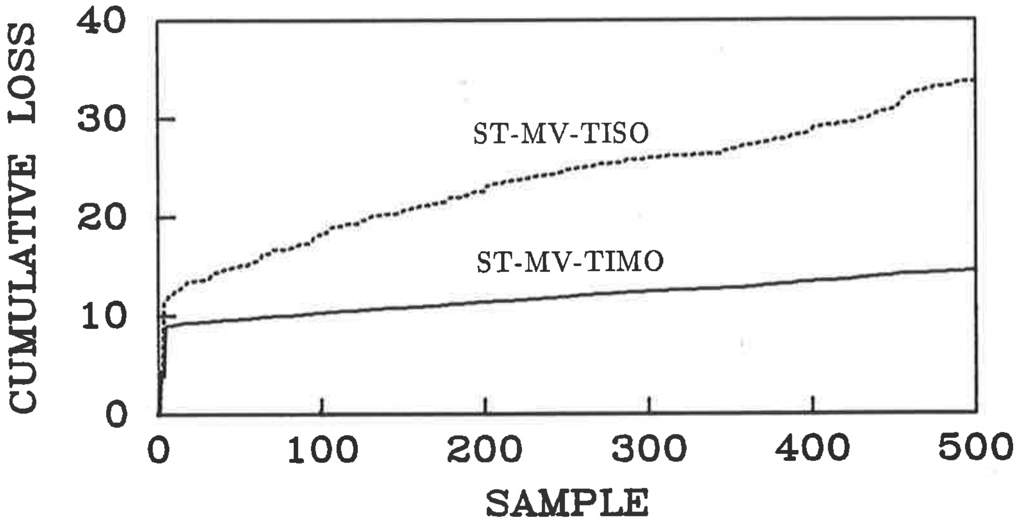

Figure 6.34: The cumulative loss functions resulting from the ST-MV-TIMO strategy (lower curve) and ST-MV-TISO strategy (upper curve) applied to a system with $c_\omega(z^{-1})$ having roots on the unit circle (tracking).

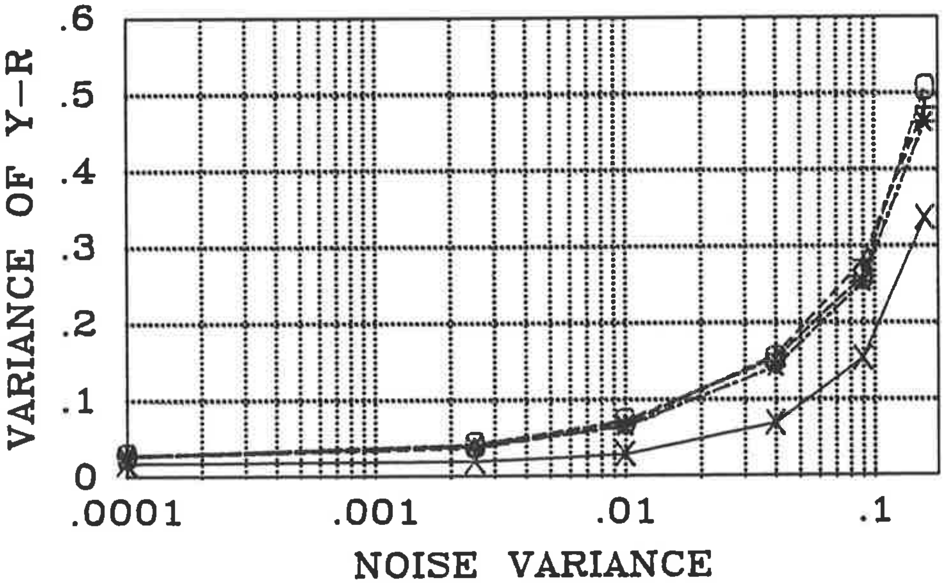

Figure 6.35: The tracking error variance vs. input noise variance for the ST-MV-TIMO ("x") and ST-MV-TISO schemes (tracking): $\alpha(t) = 1$ ("o"), 0.995 ("$*$"), and 0.99 ("+").

the latter scheme.

Let us now summarize the results of the above simulation studies in which the noise polynomial has roots on the unit circle:

- the parameter estimates of the ST-MV-TIMO strategy converged to parameters of the optimal control law;
- the ST-MV-TISO strategy based on the restricted complexity predictor leads to a suboptimal control law, with possibility that the performance in comparison with the ST-MV-TIMO scheme is significantly inferior.

6.3 Concluding remarks.

In this chapter the minimum prediction error, self-tuning control of deterministic and stochastic systems having the feedback configuration FD is considered. The predictors employed in the development of controllers were introduced in sections 3.2 and 3.3.

It is assumed that the system has one control input, and one output which is to be controlled. For a system subject to the stochastic disturbance, it is assumed that there is one white noise system input. If the only output which is available for feedback is the controlled output, then the system is said to have the feedback configuration FI. Such an approach has been commonly considered for the development of self-tuning controllers. We assume, however, that there are some *additional system outputs* available for feedback. The controlled output may or may not be used for feedback. Such a system is said to have the feedback configuration FD. For deterministic systems at least one output is required for feedback, and for stochastic systems more than one output is required for feedback.

The concept of a system having the feedback configuration FD was introduced in this thesis for the purpose of modeling systems with some additional output variables available for feedback.

The *new* minimum prediction error controller is developed for *deterministic* systems having the feedback configuration FD. Such a controller is likely to involve *less coefficients* and to be of *degree smaller* than the corresponding controller (i.e., the controller minimizing the same cost function) for the same system having the feedback configuration FI. Both features of the minimum prediction error controller for case FD are desirable in its application to self-tuning control. In such an application, *faster convergence rate* of parameter estimates can be expected for case FD than for FI due to the reduction in the number of estimated coefficients of the control law. Furthermore, *more recent input-output data* is involved in an estimator due to a shorter memory of the controller for case FD. The latter feature is especially desirable in self-tuning control of time-varying systems.

Two self-tuning strategies based on the minimum prediction error control law for deterministic systems are considered. The first strategy (ST-MPE-SIMO) involves some operations on the estimated parameters in order to determine parameters of the control law; the performance of this strategy is assessed by simulation studies. The second strategy (ST-MPE-L-SIMO) estimates parameters of the control law directly; global convergence of this strategy is proved. No persistent excitation condition is needed for the global convergence of the self tuner.

Simulation studies performed for the ST-MPE-SIMO strategy demonstrate an *improvement in the convergence rate* of the self tuner for case FD in comparison with case FI in the application to a *robot manipulator.*

For systems having the feedback configuration FD is possible to separate the controlled and feedback variables. Simulation studies show that if the measurement of the controlled output is contaminated with white noise, then such a separation improves significantly performance of the self-tuning closed-loop system in comparison with that of case FI.

The *new* weighted minimum variance controller is developed for *stochastic* systems having the feedback configuration FD. It is shown that the noise polynomial $c_\omega(z^{-1})$ is not a factor of the closed-loop system characteristic polynomial in case FD, although it is in case FI. The variance of the tracking errors resulting from the weighted minimum variance control law within the feedback configuration FD and

FI is identical. However, for case FD it is possible to achieve the weighted minimum variance control *without* feedback from the controlled output. In the nonadaptive case this implies that the measurement of the controlled output is not required for the implementation of the weighted minimum variance control law.

The minimum variance *self-tuning* strategy developed for stochastic systems having the feedback configuration FD employs the *linear regression* estimator (the recursive least squares with condition number monitoring). On the other hand, self-tuning controllers introduced for systems having the feedback configuration FI involve *pseudo-linear regression* estimators. A *significant improvement in the convergence rate* of parameter estimates of the self-tuning strategy can be expected for case FD in comparison with case FI as a consequence of using the linear regression estimator. Moreover, the use of the linear regression method enhances the *numerical robustness* of the estimator.

Furthermore, some system assumptions concerning the noise polynomial $c_\omega(z^{-1})$ are modified for case FD in comparison with case FI. Firstly, the roots of $c_\omega(z^{-1})$ are allowed to lie *on the unit circle*. On the other hand, self tuners for case FI require the roots of the noise polynomial to lie *strictly inside the unit circle*; otherwise, *suboptimal or time-varying* controllers must be considered for application in self-tuning control. Secondly, it is known that the convergence of self-tuning controllers for case FI was established subject to the *strictly positive real* (SPR) condition imposed on a transfer function involving the noise polynomial $c_\omega(z^{-1})$. There are only a few methods of side-stepping the SPR condition. These methods are computationally involved, often require dither injection into the closed-loop system, or require estimator overparameterization. On the other hand, the global convergence of the self-tuning controller for case FD *does not* require any SPR condition. Hence, the SPR condition is overcome by utilization of additional outputs for feedback. Furthermore, the complexity of the strategy for case FD is comparable with that of the strategies for case FI which are subject to the SPR condition. It is assumed, however, for the strategy developed for case FD that the open-loop system is stable. Furthermore, in the present work it is assumed that the system is controllable. (Consequently, the roots of the noise polynomial lying on the unit circle *cannot* arise due to the system uncontrollable modes lying on the unit circle, as they do for the system ARMAX model [9, chapter 7].)

The *removal of the SPR condition* is regarded as the most significant feature of the self-tuning controller for systems having the feedback configuration FD, apart of the *replacement of the pseudo-linear regression estimator by the linear regression estimator.*

Simulation studies demonstrate the improvements in the performance of the self-tuning strategy for case FD in comparison with the strategies for case FI (convergence rate, numerical robustness, optimality, strictly positive real condition, roots of the noise polynomial on the unit circle). Furthermore, some insight in given into the properties of the condition number monitoring (CNM) technique employed in the strategy for case FD to guarantee boundedness of the condition number of the estimator covariance matrix. In particular, it is demonstrated that the performance of the self-tuning strategy (i.e., the minimization of the tracking error variance) is not affected by the monitoring if appropriate initial values of parameters of the CNM are chosen. On the other hand, the CNM technique effectively eliminates the undesirable drifts of parameter estimates observed after the initial estimation phase when the monitoring is not employed.

The *global convergence* of the strategy developed for stochastic systems having the feedback configuration FD is established using the martingale approach. No persistent excitation condition is needed for the global convergence of the self tuner.

Chapter 7

Conclusions and suggestions for future research.

In this thesis two approaches to self-tuning predictive control have been considered. The approaches are classified in terms of the *feedback configuration* of the system. The first and conventional approach assumes that *only* the outputs which are to be controlled are available for feedback (feedback configuration FI). The second approach assumes that apart of the controlled outputs there are some *other* system outputs which are available for feedback (feedback configuration FD); the controlled output may or may not be used as a feedback variable.

Self-tuning control of systems having the feedback configuration FI.

Two approaches to self-tuning control of multi-input, multi-output, square deterministic systems having the feedback configuration FI have been considered in this thesis. The first approach is based on minimization of the single-stage cost function and leads to the *minimum prediction error* control law. For this approach a substantial *reduction of the prior system knowledge* required by the self tuner has been achieved in comparison with other self-tuning controllers. The second approach is based on minimization of the multi-stage cost function and leads to the *long-range predictive* control law.

Design of the above-mentioned controllers involves a characterization of the system delay structure. Unique left and right interactor matrices [33,35] were commonly employed as such characterizations for the development of self-tuning controllers. However, an assessment of (i) the requirements of an admissible control law, and (ii) the properties of the interactor matrices, led the author to develop *new characterizations of the multivariable linear system delay structure* called the *left and right nilpotent interactor matrices* (see subsections 2.3.2 and 2.3.3). The nilpotent interactor matrices are *nonunique, have a general, square rather than triangular structure, and possess properties essential for the design of minimum prediction error controllers.*

The nilpotent interactor matrix is evaluated as a *product of first degree polynomial matrices.* A *new* algorithm proposed in subsection 2.3.2 calculates the left nilpotent interactor matrix from the numerator *polynomial matrix* of the right matrix fraction system description (or of the right difference operator representation). For the purpose of comparison, the unique left interactor matrix can be calculated either from the *matrix of rational functions* [33] or from the *two polynomial matrices,* $A(p)$ and $B(p)$, of the system DARMA model [39] (in the latter case, *division* of polynomial matrices is required). Another advantage of the algorithm is that it operates on the *numerical representation* of the numerator polynomial matrix in the form of a matrix of coefficients. Hence, the algorithm operates on a *real matrix*; however, it requires decomposition (such as QR, SVD, or LDU) of real matrices of dimension of the system transfer matrix. Since the algorithm operates on the numerical representation of polynomial matrices it is *amenable to computer-based calculations* and can be easily implemented with any *matrix-oriented* software (see Appendix D). Furthermore, the algorithm can also be used for the calculation of the *right* nilpotent interactor matrix from the polynomial matrix $B(p)$ of the system *DARMA model;* this involves, however, additional transpositions of polynomial matrices as described in subsection 2.3.4.

The algorithms of the same type as the algorithm for the calculation of the left nilpotent interactor matrix were subsequently employed for *other* computational problems in linear systems theory in [81]. The latter results were adopted in the present work to introduce a *dual* algorithm for the calculation of the right nilpotent interactor matrix from the polynomial matrix $B(p)$ of the DARMA model; this obviates the necessity of additional transpositions of polynomial matrices required if the algorithm for the left nilpotent interactor is used to calculate the right nilpotent interactor matrix (see subsection 2.3.3).

In this thesis, nilpotent interactor matrices have been employed in the design of controllers for multivariable *square* deterministic systems. However, nilpotent interactor matrices, and algorithms developed for their calculation, could be used in self-tuning control of *nonsquare* systems with the on-line evaluation of the interactor matrix. This is facilitated by the *stability* of nilpotent interactor matrices which is guaranteed by the proposed algorithms for nonsquare as well as square systems. The nonuniqueness of nilpotent interactor matrices might also provide a *degree of freedom* in terms of minimization of the variances of individual outputs in the design of controllers for *stochastic* systems for which interactor matrices are nondiagonal.

The *right nilpotent interactor matrix* has been employed in the design of the minimum prediction error controller using the approach proposed for the (unique) right interactor matrix in [35] (see section 4.1). The advantage of the proposed minimum prediction error controller in comparison with that of [35] is the elimination of the effect of a deterministic disturbance, the generator model of which is incorporated in the system DARMA model.

A *new* indirect self-tuning controller based on the above minimum prediction error control law has been developed (see section 4.2). This self tuner involves the *on-line* calculation of the right nilpotent interactor matrix from the estimates of the DARMA model using the algorithm presented in subsection 2.3.3. The on-line calculation of the interactor matrix increases the computational burden in comparison with strategies based on the assumption of prior knowledge of the interactor matrix. However, the prior system knowledge required by this self tuner is reduced to the *upper bound on the degree of polynomial matrices of the DARMA model.* This is a *very significant* relaxation of the prior system knowledge required in self-tuning predictive control, since *no* knowledge of the interactor matrix is needed. The prior system knowledge is also relaxed in comparison with the indirect pole-placement self-tuning controller of [39], which involves estimation of the same number of parameters as the new self tuner; for the pole-placement self tuner prior knowledge of the system observability indexes and an upper bound on the controllability index is required [39].

Another (and the only one known to the author) self-tuning minimum prediction error strategy which requires the same (minimal) prior system knowledge was proposed in [39]. The first difference between the strategies is related to the minimized cost function. The cost function of [39] penalizes the tracking error between the controlled outputs and the reference sequence filtered by the (left) interactor matrix. On the other hand, the cost function associated with the new strategy penalizes the actual tracking error; furthermore, it penalizes the control signal filtered by the inverse of the (right) interactor matrix. One would need to examine which cost function is more meaningful for a particular application. For example, if future values of the reference sequence are unknown then, for systems with nondiagonal interactor matrices, undesirable transient behaviour of the controlled outputs can be observed for the self tuner of [39]. On the other hand, for the new strategy only delay in tracking is observed if zero control weighting is assumed. The second difference between the strategies is related to the operations performed on polynomial matrices in order to synthesize the control law. The new strategy involves *multiplication* of polynomial matrices in contrast to the self tuner of [39] which requires two *divisions* of polynomial matrices to be performed at each sample instant (one division is needed to calculate the interactor matrix).

Note that as a result of employing in the new self-tuning controller a predictive control law, which is robust to model overparameterization, (i) the *upper bound* rather than the exact value of the degree of polynomial matrices in the DARMA model is required, (ii) the effect of a *deterministic disturbance* (the generator model of which is incorporated in the system DARMA model) is eliminated. On the other hand, recall that the pole placement self tuners are sensitive to common factors resulting from the model overparameterization; furthermore, it is assumed that the open-loop system is controllable which implies that some *ad hoc* methods for elimination of the effect of deterministic disturbances must be employed [182,43].

Simulation studies demonstrate the applicability of the self tuner based on the right nilpotent interactor matrix to *robot control* (see section 4.3). However, it would be desirable to investigate (i) the robustness of this strategy when estimation is performed in a noisy environment, and (ii) the ability of the strategy to follow changes in the system delay structure.

The left interactor matrix (in particular the nilpotent interactor matrix) has been employed in the development of the *new* indirect self-tuning, long-range predictive, receding horizon controller (see section 5.1). The cost function, postulated for the development of the long-range predictive controller, involves the tracking error filtered by the interactor matrix.

The new long-range predictive control law *generalizes* the Dynamic Matrix Control scheme [30] and the Generalized Predictive Control strategy. The latter strategy was introduced in [29] for *scalar* systems as an extension to the Dynamic Matrix Controller and forms the basis for the long-range predictive controller developed in this thesis for *multivariable* systems. Furthermore, in relation to the Dynamic Matrix Controller, the new long-range predictive controller (i) is applicable to open-loop *unstable* systems, (ii) is applicable to systems with a *general* delay structure (characterized by the interactor matrix), (iii) has been developed for the system model involving smaller number of parameters which improves the *convergence rate* of estimates in self-tuning control, (iv) *eliminates* the effect of a deterministic disturbance, the generator model of which is incorporated in the system DARMA model.

The long-range predictive strategy has, in particular, two *time-oriented* tuning knobs: the *prediction horizon* P and the *control horizon* C. The tuning knobs P and C facilitate control of *nonminimum phase and/or unstable systems.* Moreover, the choice of their values is straight-forward in self-tuning control applications. The control weighting matrix Λ may be used to decrease the excessive control effort. Furthermore, this is its main role, in contrast to the minimum prediction error controllers for which the appropriate choice of $\Lambda \neq 0$ may be crucial to guarantee closed-loop system stability for nonminimum phase systems.

The strategy relies on prediction of system outputs (filtered by the interactor matrix) over the prediction horizon P. It is assumed that there are no changes in the control signal beyond the control horizon C. The choice $C = P$ and $\Lambda = 0$ results in the minimum prediction error (or one-step-ahead) control law which is sensitive to nonminimum phase systems. The stabilizing effect for *nonminimum phase* systems is achieved by setting $C < P$, which prevents the growth of the control signal, even if $\Lambda = 0$ (since an infinite penalty is placed on changes of the control signal beyond the control horizon). Furthermore, the choice of $C < P$ reduces the computational burden.

It has been found in the simulation studies that P should be chosen such that $(k + P)T_s$ (where k is the degree of the interactor matrix and T_s is the sampling interval) covers a significant part of the open-loop system step response. Furthermore, the choice of $C = 0$ leads to satisfactory performance for many systems, even nonminimum phase and unstable systems such as the robot manipulator considered in simulation studies. In general, the greater the difference $P - C$ is the more sluggish is the output response and the less control effort is involved; furthermore, for systems with nondiagonal interactor matrices the undesirable effect of unknown future values of the reference sequence becomes less significant. A larger value of C permits more changes in the control signal, which leads to faster output response.

The long-range predictive strategy is well-suited for *programmed control* (i.e., if future values of the reference sequence are known) since it introduces an *anticipatory* control action which can improve the transient response of sluggish systems or systems with large delay. It has been found in the simulation studies that for the programmed control the time required for the outputs to reach new steady-state values, after a change in the reference sequence, is largely independent of P. Furthermore, for systems with diagonal interactor matrices, programmed control reduces the *control effort* markedly and without degradation of performance in comparison with the fixed reference sequence model control (i.e., with unknown future values of the reference sequence approximated by the present value).

The long-range predictive self tuner is *robust* to model overparameterization. Its drawback is, however, the requirement of the *prior knowledge* of the left interactor matrix.

New computational algorithms have been developed for the calculation of the *matrices of coefficients* of polynomial matrices required to synthesize the long-range predictive control law (see section 3.1). These algorithms can be viewed as closed or *contracted* forms of iterative algorithms which yield *coefficients* of the polynomial matrices rather than the matrices of coefficients. The new algorithms are amenable to *computer-based calculations* using matrix-oriented software. In particular, a *recursive* algorithm has been developed for the calculation of matrices of coefficients of the predictor polynomials for subsequent values of the prediction horizon; this reduces the computational burden required to solve a polynomial equation for each value of the prediction horizon.

Comparison of a number of properties of the long-range predictive and minimum prediction error self-tuning controllers suggests that a combination of both approaches (see approach (iv) on page 60) could be considered for the development of a self-tuning strategy. These properties are (a) the required prior system knowledge, (b) cost functions penalizing outputs filtered by the left interactor matrix or the control inputs filtered by the inverse of the right interactor matrix, (c) tuning knobs and their effect on the closed-loop system performance, and (d) sensitivity of the closed-loop system performance to the choice of the values of the tuning knobs when applied to nonminimum phase and/or unstable systems. The resulting self-tuning long-range predictive controller would involve the *on-line* calculation of the right nilpotent interactor matrix from the estimates of the system DARMA model, thus eliminating the requirement of its prior knowledge.

Self-tuning long-range predictive control for *stochastic* systems having the feedback configuration FI is considered in section 5.2. The long-range predictive control law has been developed for systems modeled by the ARMAX model, using the approach proposed in [29] for the Generalized Predictive Control based on the CARIMA model. The latter strategy was found in [29] to possess a number of properties desirable in self-tuning control; however, no general *closed-loop system analysis* has yet been presented. Such an analysis, involving the *stability criterion*, is presented in this thesis for the long-range predictive control law. As a result, closed-loop system stability can now be verified easily for a system with known parameters. Furthermore, the influence of the control weighting coefficient λ and the two time-oriented tuning knobs, P and C, on the closed-loop system *pole location* can now be examined.

It was found in simulation studies that the time-oriented tuning knobs of the long-range predictive control law facilitate control of dynamically complex systems. Superior *robustness* properties were observed with respect to (i) variable delay in comparison with the generalized minimum variance self tuner, and (ii) with respect to underparameterization in comparison with the pole-placement self tuner [150]. Furthermore, the long-range predictive self tuner *outperforms* the Generalized Predictive Controller in elimination of the effect of stochastic disturbances which are adequately represented by the ARMAX model. (In [94] the ARMAX model was reported to be more suitable than CARIMA for self-tuning control of robot manipulators.)

Self-tuning control of systems having the feedback configuration FD.

The second approach to self-tuning control concerns systems having the feedback configuration FD (see chapter 6).

It is assumed that the system has one control input, and one output which is to be controlled. There are, however, some *additional system outputs* which can be utilized for feedback. For a system subject to a stochastic disturbance it is assumed that there is one white noise input. Hence, single-input, multi-output (SIMO) deterministic systems, and two-input, multi-output (TIMO) stochastic systems are considered.

For the development of minimum prediction error controllers for the feedback configuration FD, the *SIMO-type predictor* for deterministic systems and the *TIMO-type optimal predictor* for stochastic systems have been introduced (see sections 3.2 and 3.3). These *new* predictors utilize measurement of *additional system outputs*. The prediction error of the optimal TIMO-type prediction is the same as that of the corresponding optimal prediction for the same system within the feedback configuration FI.

Computational methods based on a numerical representation of Diophantine equations have been proposed for the *off-line* calculation of predictor parameters for systems with known parameters. Such an approach translates the problem of solving a polynomial equation into a problem of solving a set of linear algebraic equations. Furthermore, the dimension of the set of linear algebraic equations is reduced by separation into more than one set of equations.

For systems with unknown parameters, the *adaptive* SIMO- and TIMO-type predictors have been introduced. The properties of adaptive predictors correspond to those of the self-tuning minimum prediction error controllers discussed below.

New minimum prediction error control laws have been developed for deterministic and stochastic systems having the feedback configuration FD.

The *same* control objective can be achieved for a system having the feedback configuration FD and FI. However, the (physical) *separation* of the controlled output from the feedback outputs is possible for systems having the feedback configuration FD. The separation implies, for instance, that the nonadaptive (weighted) minimum variance control can be achieved *without measurement* of the controlled output.

There are many advantages of utilization of additional system outputs for feedback in self-tuning control.

For deterministic systems, the minimum prediction error controller developed for the feedback configuration FD is likely to involve *less parameters* and to be of *degree smaller* than the controller aiming at the same control objective for case FI. Both features of the control law for case FD are desirable in direct self-tuning control since *less parameters* is to be estimated than in case FI. Furthermore, *more recent input-output data* is used for the estimation of controller parameters. As a result, *faster convergence rate* of parameter estimates has been observed in the simulation studies for self tuners within the feedback configuration FD than for case FI.

For stochastic systems having the feedback configuration FI, the weighted minimum variance control law involves predictions of the controlled output filtered by the noise polynomial, in contrast to the corresponding new controller for systems having the feedback configuration FD. This difference implies a number of important features in self-tuning control.

Firstly, the *linear regression* estimator is used in case FD in place of the *pseudo-linear regression* estimators employed in self tuners for systems having the feedback configuration FI. As a result, the *convergence rate* of parameter estimates and the *numerical robustness* of the estimator involved in the self tuner within the feedback configuration FD are *markedly superior* to those of estimators in case FI.

Secondly, some system assumptions required by self-tuning controllers within the feedback configuration FI are modified for case FD. Of particular benefit is the *removal of the strictly positive real condition* involving the noise polynomial associated with the controlled output. On the other hand, such a condition is required for convergence of self-tuning controllers within the feedback configuration FI. The methods of overcoming the strictly positive real condition proposed for systems having the feedback configuration FI involve significant *computational burden*, dither injection which leads to *suboptimal* performance unless some other precautions are taken, estimator overparameterization which affects the *convergence rate* of the parameter estimates, or restrict control objective to *regulation.* On the other hand, utilization of additional system outputs for feedback is a *simple* and *effective* method of overcoming the strictly positive real condition. Furthermore, in case FD the *optimal and time-invariant* control law is employed in self-tuning control of systems having the roots of the noise polynomial lying on the unit circle. In case FI, for such systems the time-invariant and *suboptimal* or optimal and *time-varying* control laws must be considered for application in self-tuning control.

However, it is assumed for case FD that the open-loop system is stable and controllable; such assumptions are not required for self-tuning controllers for systems having the feedback configuration FI.

The *global convergence* of the new self-tuning controllers for systems having the feedback configuration FD has been established. The self tuner for case FD is the *first* globally convergent strategy which employs the linear regression rather than pseudo-linear regression estimator for self-tuning control (tracking problem) of stochastic systems.

In order to guarantee global convergence of the self-tuning strategy for stochastic systems, the monitoring of the condition number of the estimate error covariance matrix has been employed. The condition number monitoring technique was proposed in [57] as a modification to the matrix gain, pseudo-linear regression estimator for self-tuning control of stochastic systems having the feedback configuration FI. The application of this technique to self-tuning control of systems having the feedback configuration FD is even more *feasible* due to the enhancement of the convergence rate by using the linear regression estimator. Nevertheless, other modified matrix gain estimators, such as the weighted least squares method proposed in [56], could also be adopted for case FD.

There is a number of ways of extending the results for self-tuning control of systems having the feedback configuration FD. Firstly, it would be desirable to develop self-tuning strategies for systems subject to deterministic disturbances. Secondly, self-tuning control of more than one output for systems having the feedback configuration FD could be considered. Since the number of estimated parameters grows rapidly for control of more than one output, the reduction of the size of the parameter estimation problem (or more generally an improvement in the convergence rate of parameter estimates) becomes the *critical* issue for application of self-tuning control [39]. The utilization of additional system outputs for feedback in self-tuning control has been shown in the present work to provide a means for achieving the improvement in the convergence rate of parameter estimates. Thirdly, self-tuning control of stochastic systems having the feedback configuration FD with an arbitrary number of independent noise sources (for example equal to the number of system outputs rather than to the number of the controlled outputs) could be considered.

In the author's future research, the developments of chapter 6 will be applied in the design of self-tuning power system stabilizers which utilize additional stabilizing system variables.

Appendix A

Forward- and backward-shift operator.

Let us denote the *forward shift operator* by q. Then

$$qx(t) \overset{\text{def}}{=} x(t+1), \qquad t \geq 0$$

where $x(t)$ is the value of the sequence $\{x(t)\} \overset{\text{def}}{=} x(0), x(1), \ldots$ defined at discrete, integer values of time $t \in \{0, 1, \ldots\}$. Furthermore, let us denote the *backward shift (delay) operator* by p, i.e.,

$$p = q^{-1}. \tag{A.1}$$

Then

$$px(t) \overset{\text{def}}{=} x(t-1), \qquad t > 0; \qquad px(0) \overset{\text{def}}{=} 0.$$

The shift operator is defined for a semi-infinite sequence. (Sometimes it is defined for doubly infinite sequences (i.e., $t \in \{\ldots - 1, 0, 1, \ldots\}$) [77] [59, p. 47].

Let us establish a link between the shift operator and the single-sided z-transform of a sequence $\{x(t)\}$, i.e.,

$$Z\{x(t)\} \overset{\text{def}}{=} X(z) = \sum_{t=0}^{\infty} x(t)z^{-t}.$$

For this purpose let us consider z-transform of the shifted sequence $\{x(t+1)\}$

$$Z\{x(t+1)\} = Z\{q\{x(t)\}\} = zX(z) - zx(0),$$

where $x(0)$ denotes the initial condition [183] [9, Appendix A.2]. Thus, the z-transform of the shifted sequence can be obtained by replacing the q-operator by the complex variable z (with the appropriate change of time function $x(t)$ to the z-transform $X(z)$ of $x(t)$) if initial conditions are zero.

Although the z-transform is defined above for a semi-infinite time sequence, sometimes it is defined for doubly infinite sequences [77].

Appendix B

Numerical representation of a polynomial matrix and multiplication of polynomial matrices.

Let us explain certain aspects of the numerical representation of polynomial matrices and multiplication of polynomial matrices as in [79, chapter 3]. Those results are often used throughout the thesis.

Let us consider the following $l \times m$ polynomial matrix $Q(x)$

$$Q(x) = Q_0 + Q_1 x + \cdots + Q_s x^s.$$

The polynomial matrix $Q(x)$ can be expressed as a product of a numerical block matrix Q containing only coefficients of polynomial elements of $Q(x)$, and a matrix of powers of the indeterminate x of $Q(x)$, as follows

$$Q(x) = Q(X_s \otimes I_m) \tag{B.1}$$

where

$$Q = \begin{bmatrix} Q_0 & Q_1 & \dots & Q_s \end{bmatrix}, \tag{B.2}$$

$$X_s = \begin{bmatrix} 1 & x & \dots & x^s \end{bmatrix}', \tag{B.3}$$

and I_m is the $m \times m$ identity matrix, $\otimes$ denotes the Kronecker product, and $'$ denotes transposition. The $l \times (s+1)m$ matrix Q is called the *matrix of coefficients* of the polynomial matrix $Q(x)$, in contrast to the $l \times m$ *coefficient matrix* Q_i $(i = 0, \ldots, s)$ of the polynomial matrix $Q(x)$. If $x = z^{-1}$ or $x = p$ (p is the backward shift operator), then vector B.3 is denoted Z_s or P_s.

It is said that the degree of a polynomial matrix $Q(x)$ is s, i.e., $s = \deg Q(x)$, if s is the highest power of the polynomial indeterminate for which the coefficient matrix is nonzero.

Let us now define the following $r \times c$ matrix

$$R_i^{r,c} \overset{\text{def}}{=} \begin{bmatrix} 0_{r \times i} & I_r & 0_{r \times (c-r-i)} \end{bmatrix} \tag{B.4}$$

containing ones along the $(i+1)$-th diagonal and zeros elsewhere $(c \geq r,\ 0 \leq i \leq c-r)$. Let us consider the numerical representation of a $m \times n$ polynomial matrix $D(x)$ of degree t by the matrix of coefficients $D = [D_0 \ldots D_t]$. Then the *resultant* $\langle D \rangle_s$ *of order* s of the polynomial matrix $D(x)$ is defined as a parallelogram block-matrix of polynomial coefficient matrices

$$\langle D \rangle_s \overset{\text{def}}{=} \begin{bmatrix} D(R_0^{t+1,s+t+1} \otimes I_n) \\ D(R_1^{t+1,s+t+1} \otimes I_n) \\ \vdots \\ D(R_s^{t+1,s+t+1} \otimes I_n) \end{bmatrix} = \begin{bmatrix} D_0 & \ldots & D_t & 0 & \ldots & 0 \\ 0 & D_0 & \ldots & D_t & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & D_0 & \ldots & D_t \end{bmatrix}, \tag{B.5}$$

having the dimension $(s+1)m \times (s+t+1)n$.

Finally, the multiplication of polynomial matrices can be represented by an equivalent operation on numerical block-matrices. Consider the $l \times m$ polynomial matrix $Q(x)$ of degree s and the $m \times n$ polynomial matrix $D(x)$ of degree t. Then the $l \times (s+t+1)n$ matrix of coefficients $M = [M_0 \ldots M_{s+t}]$ of the product $M(x) = Q(x)D(x)$ is given by

$$M = Q\langle D \rangle_s. \tag{B.6}$$

Appendix C

An algorithm for the calculation of a nilpotent interactor matrix for linear multivariable systems.

The following paper concerns a new characterization of the delay structure of a multivariable linear system. Such a characterization is called the (left) nilpotent interactor matrix. An algorithm for the calculation of the (left) nilpotent interactor matrix is presented. This paper appeared in the IEEE Transactions on Automatic Control in March 1987. Subsection 2.3.2 provides a brief summary of the main results.

Furthermore, an example of the calculation of the (left) nilpotent interactor matrix is presented. The example deals with details of the implementation of the algorithm in MATLAB (or MATRIX$_X$) software package, which are not covered in the paper.

The macro XLNI, which is written in MATLAB commands and implements the algorithm for calculation of the left nilpotent interactor matrix, is given in Appendix D.

Rogozinski, M., Paplinski, A. & Gibbard, M. (1987). An algorithm for the calculation of a nilpotent interactor matrix for linear multivariable systems. *IEEE Transactions on Automatic Control, 32*(3), 234-237.

Example of calculation of the left nilpotent interactor matrix using MATLAB software package (continuation of Example 2.1).

This example highlights details of implementation of the algorithm for the evaluation of the LNI matrix using MATLAB [110] (or MATRIX$_X$ [113]) software package, which were not covered in the original paper [109].

Consider system given by the transfer matrix 2.47, p. 44. This transfer matrix can be factored into the RMF description with monic denominator polynomial matrix, as follows

$$H_{y,u}(z) = \hat{N}(z)\hat{D}^{-1}(z) = (z^n N(z^{-1}))(z^n D(z^{-1}))^{-1} = N(z^{-1})D^{-1}(z^{-1}),$$

where

$$N(z^{-1}) = \begin{bmatrix} 1+3z^{-1} & 1+4z^{-1} \\ 1+z^{-1} & 1+2z^{-1} \end{bmatrix} \quad D(z^{-1}) = \begin{bmatrix} 1+4z^{-1}+3z^{-2} & 0 \\ 0 & 1+6z^{-1}+8z^{-2} \end{bmatrix}. \tag{C.1}$$

The numerator matrix can be expressed using eqn. B.1 (Appendix B) as

$$N(z^{-1}) = N(Z_2 \otimes I_2) = [N_0 \; N_1 \; N_2](Z_2 \otimes I_2) = \begin{bmatrix} 0 & 0 & 1 & 1 & 3 & 4 \\ 0 & 0 & 1 & 1 & 1 & 2 \end{bmatrix}(Z_2 \otimes I_2). \tag{C.2}$$

In the example given in the preceding paper the column block-matrix representation was chosen to illustrate the operations involved in the algorithm rather than to demonstrate the computer implementation of the algorithm. If the column block-matrix representation is used, then the result of premultiplication by the matrix $U_L^{(i)}(z)$ at step 3 of the algorithm is represented by shifting the rows of the column block-matrix upwards.

In this example the row block-matrix representation of the numerator polynomial matrix is chosen as more convenient for the implementation of the algorithm in MATLAB commands (see Appendix B). In the implementation of the algorithm, the rows of blocks of the row block-matrix (rather than rows of column block-matrix) representation are shifted upwards-left as a result of premultiplication by the matrix $U_L^{(i)}(z)$, according to the pattern depicted in fig. C.1.

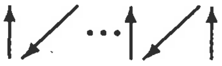

Figure C.1: The shift pattern of the rows of matrices of coefficients at step 3 of the algorithm.

The algorithm is implemented with the matrix orthogonal-triangular decomposition QR which, for a m-row matrix X, produces an upper quasi-triangular matrix T of full row rank and a unitary $m \times m$ matrix Q [110], so that

$$X = Q \begin{bmatrix} T \\ 0_k \end{bmatrix} = QR, \tag{C.3}$$

where 0_k is a $k = (m - \operatorname{rank} X)$-row zero matrix. Step 1 of the i-th iteration of the algorithm groups $k_i = m - \operatorname{rank} \mathcal{N}_0^{(i-1)} = m - r_i$ zero rows at the top of the postmultiplier of the decomposition of $\mathcal{N}_0^{(i-1)}$ (see eqn. 12 in the paper [109])

$$\mathcal{N}_0^{(i-1)} = (Q_L^{(i)})^{-1} \begin{bmatrix} 0_{k_i} \\ \mathcal{N}_{0D}^{(i)} \end{bmatrix}. ,$$

Thus it is necessary to reverse the order of columns of Q and rows of R resulting from QR factorization C.3 of $\mathcal{N}_0^{(i-1)}$ in the first step of the algorithm

$$X = \mathcal{N}_0^{(i-1)} = (QJ_m)(J_m^{-1}R) = (Q_L^{(i)})^{-1} \begin{bmatrix} 0_{k_i} \\ \mathcal{N}_{0D}^{(i)} \end{bmatrix}.$$

The unitary $m \times m$ matrix J_m, which performs the desired permutations of columns and rows of matrices Q and R, is defined as

$$J_m = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & & & . & 0 \\ \vdots & & . & & \vdots \\ 0 & . & & & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix}, \tag{C.4}$$

and is sometimes called the standard involuntary permutation matrix [184, p. 377], or the exchange matrix [119,185]. Multiplication of a matrix by the matrix J_m on the right or left side reverses the order of the matrix columns or rows, respectively.

Step 2 of the i-th iteration of the algorithm premultiplies the $\mathcal{N}^{(i-1)}(z^{-1})$ matrix by $Q_L^{(i)} = (QJ_m)^{-1} = J_m^{-1}Q^{-1} = J_mQ'$, ($J_m$ and Q are unitary and $J'_m = J_m$). Thus $Q_L^{(i)}$ is calculated as a transpose of Q resulting from decomposition of $\mathcal{N}_0^{(i-1)}$, and reverse of the order of rows of Q'. The row block-matrix of coefficients of the polynomial matrix $\overline{N}^{(i)}(z^{-1})$ resulting from step 2, has the following form

$$\overline{N}^{(i)} = \left[\overline{N}_0^{(i)}\,\overline{N}_1^{(i)}\ldots\overline{N}_n^{(i)}\right] = \begin{bmatrix} 0_{k_i} & \mathcal{N}_{1U}^{(i)} & \ldots & \mathcal{N}_{nU}^{(i)} \\ \mathcal{N}_{0D}^{(i)} & \mathcal{N}_{1D}^{(i)} & \ldots & \mathcal{N}_{nD}^{(i)} \end{bmatrix}. \tag{C.5}$$

In step 3 of the i-th iteration of the algorithm matrix $\overline{N}^{(i)}(z^{-1})$ of degree n is premultiplied by a first degree matrix $U_L^{(i)}(z)$

$$\mathcal{N}^{(i)}(z^{-1}) = U_L^{(i)}(z)\overline{N}^{(i)}(z^{-1}) = \left[\mathcal{N}_{-1}^{(i)}\,\mathcal{N}_0^{(i)}\,\ldots\,\mathcal{N}_n^{(i)}\right]\left(\left[z\;1\;\ldots\;z^{-n}\right]'\otimes I_m\right).$$

Note that $\mathcal{N}_{-1}^{(i)} = 0$ because

$$\mathcal{N}_{-1}^{(i)} = [U_0^{(i)}\,U_1^{(i)}]\begin{bmatrix}\overline{N}_0^{(i)} \\ 0\end{bmatrix} = U_0^{(i)}Q_L^{(i)}\mathcal{N}_0^{(i-1)} = \begin{bmatrix} 0_{r_i} & 0_{r_i} \\ I_{k_i} & 0_{k_i}\end{bmatrix}\begin{bmatrix} 0_{k_i} \\ \mathcal{N}_{0D}^{(i)}\end{bmatrix} = \begin{bmatrix} 0_{r_i} \\ 0_{k_i}\end{bmatrix} = 0,$$

where 0_{r_i} denotes r_i-row zero matrix. The row block-matrix of the coefficients of the polynomial matrix $\mathcal{N}^{(i)}(z^{-1})$ resulting from step 3, has the following form (cf. eqn. C.5)

$$\mathcal{N}^{(i)} = [\mathcal{N}_{-1}^{(i)}\,\mathcal{N}_0^{(i)}\ldots\mathcal{N}_n^{(i)}] = \begin{bmatrix} 0_{r_i} & \mathcal{N}_{0D}^{(i)} & \ldots & \mathcal{N}_{nD}^{(i)} \\ 0_{k_i} & \mathcal{N}_{1U}^{(i)} & \ldots & 0_{k_i}\end{bmatrix}. \tag{C.6}$$

The operation of multiplication of polynomial matrices is represented according to eqn. B.6 (Appendix B).

The algorithm is initialized with $\mathcal{N}^{(0)}(z^{-1}) = N(z^{-1})$ and $K_L^{(0)}(z) = K_0^{(0)} = I_2$. Then

i=1: $r_1 = 0$, $k_1 = 2$

step 1.

Since $r_1 = \operatorname{rank}\mathcal{N}_0^{(0)} = \operatorname{rank} N_0 = 0$ a decomposition of N_0 yields $T = 0$ (see eqn. C.3) and $Q_L^{(1)}$ can be chosen identity. Then

$$U_L^{(1)}(z) = \begin{bmatrix} z & 0 \\ 0 & z\end{bmatrix},\quad U_L^{(1)} = [U_0^{(1)}\,U_1^{(1)}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0\end{bmatrix};$$

step 2.

$$\overline{N}^{(1)} = \left[\overline{N}_0^{(1)}\,\overline{N}_1^{(1)}\,\overline{N}_2^{(1)}\right] = Q_L^{(1)}\mathcal{N}^{(0)} = I_2\mathcal{N}^{(0)} = [N_0\,N_1\,N_2]\,;$$

step 3.

$$\begin{aligned}\mathcal{N}^{(1)} &= [\mathcal{N}_{-1}^{(1)}\,\mathcal{N}_0^{(1)}\,\mathcal{N}_1^{(1)}\,\mathcal{N}_2^{(1)}] = U_L^{(1)}\langle\overline{N}^{(1)}\rangle_1 = \\ &= [U_0^{(1)}\,U_1^{(1)}]\begin{bmatrix} \overline{N}_0^{(1)} & \overline{N}_1^{(1)} & \overline{N}_2^{(1)} & 0 \\ 0 & \overline{N}_0^{(1)} & \overline{N}_1^{(1)} & \overline{N}_2^{(1)} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 & 3 & 4 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 2 & 0 & 0 \end{bmatrix},\end{aligned}$$

$$\begin{aligned}K_L^{(1)} &= [K_0^{(1)}\,K_1^{(1)}] = U_L^{(1)}\langle Q_L^{(1)}K_L^{(0)}\rangle_1 = \\ &= [U_0^{(1)}\,U_1^{(1)}]\begin{bmatrix} Q_L^{(1)}K_0^{(0)} & 0 \\ 0 & Q_L^{(1)}K_0^{(0)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.\end{aligned}$$

Since the system transfer matrix $H_{y,u}(z)$ is strictly proper, we have $N_0 = 0$ in eqn. C.2. Hence, this iteration can be omitted and the algorithm initialized with $\mathcal{N}^{(1)}(z^{-1}) = U_L^{(1)}(z)N(z^{-1})$ represented by $[\mathcal{N}_0^{(1)}\,\mathcal{N}_1^{(1)}\,\mathcal{N}_2^{(1)}]$, and $K_L^{(1)}(z) = zI_2$.

i=2: $r_2 = 1$, $k_2 = 1$,

step 1.

$$Q = \begin{bmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}, \quad R = \begin{bmatrix} -1.4142 & -1.4142 \\ 0 & 0 \end{bmatrix},$$

$$Q_L^{(2)} = J_2Q' = \begin{bmatrix} -0.7071 & 0.7071 \\ -0.7071 & -0.7071 \end{bmatrix},$$

$$U_L^{(2)}(z) = \begin{bmatrix} 0 & 1 \\ z & 0 \end{bmatrix}, \quad U_L^{(2)} = [U_0^{(2)}\,U_1^{(2)}] = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix};$$

step 2.

$$\begin{aligned}\overline{N}^{(2)} &= \left[\overline{N}_0^{(2)}\,\overline{N}_1^{(2)}\,\overline{N}_2^{(2)}\right] = Q_L^{(2)}\mathcal{N}^{(1)} = \\ &= \begin{bmatrix} 0 & 0 & -1.4142 & -1.4142 & 0 & 0 \\ -1.4142 & -1.4142 & -2.8284 & -4.2426 & 0 & 0 \end{bmatrix};\end{aligned}$$

step 3.

$$\begin{aligned}
\mathcal{N}^{(2)} &= [\mathcal{N}_{-1}^{(2)}\,\mathcal{N}_0^{(2)}\,\mathcal{N}_1^{(2)}\,\mathcal{N}_2^{(2)}] = U_L^{(2)}\langle\overline{N}^{(2)}\rangle_1 = \\
&= [U_0^{(2)}\,U_1^{(2)}]\begin{bmatrix} \overline{N}_0^{(2)} & \overline{N}_1^{(2)} & \overline{N}_2^{(2)} & 0 \\ 0 & \overline{N}_0^{(2)} & \overline{N}_1^{(2)} & \overline{N}_2^{(2)} \end{bmatrix} = \\
&= \begin{bmatrix} 0 & 0 & -1.4142 & -1.4142 & -2.8284 & -4.2426 & 0 & 0 \\ 0 & 0 & -1.4142 & -1.4142 & 0 & 0 & 0 & 0 \end{bmatrix},
\end{aligned}$$

$$\begin{aligned}
K_L^{(2)} &= [K_0^{(2)}\,K_1^{(2)}\,K_2^{(2)}] = U_L^{(2)}\langle Q_L^{(2)}K_L^{(1)}\rangle_1 = \\
&= [U_0^{(2)}\,U_1^{(2)}]\begin{bmatrix} Q_L^{(2)}K_0^{(1)} & Q_L^{(2)}K_1^{(1)} & 0 \\ 0 & Q_L^{(2)}K_0^{(1)} & Q_L^{(2)}K_1^{(1)} \end{bmatrix} = \\
&= \begin{bmatrix} 0 & 0 & -0.7071 & -0.7071 & 0 & 0 \\ -0.7071 & 0.7071 & 0 & 0 & 0 & 0 \end{bmatrix};
\end{aligned}$$

i=3: $r_3 = 1$, $k_3 = 1$,

step 1.

$$Q = \begin{bmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}, \quad R = \begin{bmatrix} 2 & 2 \\ 0 & 0 \end{bmatrix},$$

$$Q_L^{(3)} = J_2 Q' = \begin{bmatrix} -0.7071 & 0.7071 \\ -0.7071 & -0.7071 \end{bmatrix},$$

$$U_L^{(3)}(z) = \begin{bmatrix} 0 & 1 \\ z & 0 \end{bmatrix}, \quad U_L^{(3)} = [U_0^{(3)}\,U_1^{(3)}] = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix};$$

step 2.

$$\overline{N}^{(3)} = \left[\overline{N}_0^{(3)}\,\overline{N}_1^{(3)}\,\overline{N}_2^{(3)}\right] = Q_L^{(3)}\mathcal{N}^{(2)} = \begin{bmatrix} 0 & 0 & 2 & 3 & 0 & 0 \\ 2 & 2 & 2 & 3 & 0 & 0 \end{bmatrix};$$

step 3.

$$\begin{aligned}
\mathcal{N}^{(3)} &= [\mathcal{N}_{-1}^{(3)}\,\mathcal{N}_0^{(3)}\,\mathcal{N}_1^{(3)}\,\mathcal{N}_2^{(3)}] = U_L^{(3)}\langle\overline{N}^{(3)}\rangle_1 = \\
&= [U_0^{(3)}\,U_1^{(3)}]\begin{bmatrix} \overline{N}_0^{(3)} & \overline{N}_1^{(3)} & \overline{N}_2^{(3)} & 0 \\ 0 & \overline{N}_0^{(3)} & \overline{N}_1^{(3)} & \overline{N}_2^{(3)} \end{bmatrix} = \\
&= \begin{bmatrix} 0 & 0 & 2 & 2 & 2 & 3 & 0 & 0 \\ 0 & 0 & 2 & 3 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \text{(C.7)}
\end{aligned}$$

$$\begin{aligned} K_L^{(3)} &= [K_0^{(3)}\, K_1^{(3)}\, K_2^{(3)}\, K_3^{(3)}] = U_L^{(3)} \langle Q_L^{(3)} K^{(2)} \rangle_1 = \\ &= [U_0^{(3)}\, U_1^{(3)}] \begin{bmatrix} Q_L^{(2)} K_0^{(2)} & Q_L^{(2)} K_1^{(2)} & Q_L^{(2)} K_2^{(2)} & 0 \\ 0 & Q_L^{(2)} K_0^{(2)} & Q_L^{(2)} K_1^{(2)} & Q_L^{(2)} K_2^{(2)} \end{bmatrix} = \\ &= \begin{bmatrix} 0 & 0 & 0.5 & -0.5 & 0.5 & 0.5 & 0 & 0 \\ -0.5 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \end{bmatrix}; \end{aligned}$$

i=4: $r_4 = 2$ which terminates the algorithm and the left nilpotent interactor matrix is

$$K_L(z) = K_L^{(3)}(z) = \begin{bmatrix} 0.5z^2 + 0.5z & -0.5z^2 + 0.5z \\ -0.5z^3 + 0.5z^2 & 0.5z^3 + 0.5z^2 \end{bmatrix}.$$

Appendix D

Self-tuning control software for simulation studies.

Simulation studies, presented in chapter 4 and in section 5.1, have been carried out by means of matrix-oriented software packages MATLAB [110] and MATRIX$_\mathrm{X}$ [113]. Some of the macros developed for the purpose of simulation studies are described below.

MATLAB/MATRIX$_\mathrm{X}$ macros.

Global variables:

`a` – matrix $A = [I_m\ A_1\ \dots\ A_n]$, (see eqn. 3.2, p. 59);

`b` – matrix $B = [B_1\ \dots\ B_n]$, (see eqn. 3.3, p. 59);

`c` – control horizon C;

`cov` – covariance matrix $P(t)$ of the simplified multivariable RLS estimator;

`ea` – estimate of matrix A, `ea` $= [I_m\ \widehat{A}_1(t)\ \dots\ \widehat{A}_n(t)]$;

`eb` – estimate of matrix B, `eb` $= [\widehat{B}_1(t)\ \dots\ \widehat{B}_n(t)]$ (except for **`XLNI`**);

`est` – matrix of estimates $\widehat{\Theta}(t)$ of the simplified multivariable RLS estimator;

`m` – number of inputs and outputs of the MIMOS system;

`n` – degree of polynomial matrices in the DARMA model;

`nk` – degree of the interactor matrix;

`nos` – number of samples for the simulation;

`par` – matrix consisting of estimates of A and B;

`p` – prediction horizon P;

`res` – matrix consisting of control inputs, outputs, and reference sequence values;

`rper` – $2 \times$ `rper` is the period of the square wave reference sequence;

`umx` – limit on the control signal amplitude;

`up` – $(\mathtt{n}+1)\mathtt{m} \times 1$ vector of past inputs $\mathtt{up} = [u(t-1)' \ \ldots \ u(t-n-1)']'$;

`v` – $(\mathtt{nk}+1)\mathtt{m} \times 1$ vector of past auxiliary control signals $u_k(t)$;

`vmx` – limit on the auxiliary control signal amplitude;

`yp` – $(\mathtt{n}+1)\mathtt{m} \times 1$ vector of outputs $\mathtt{yp} = [y(t)' \ \ldots \ y(t-n)']'$.

Macros:

`XAB` – matrices B_F, B_P, A_P (eqns. 5.9 to 5.11 (p. 154), and Remark 3.2, p. 69);

input: `m, n, ea, eb, p` ($\mathtt{p} = P+1$),

`al,be` (from `XPRL`),

output: `al` ($\mathtt{al} = A_P$),

`be` ($\mathtt{be} = [B_F(J_{P+1} \otimes I_m) \ \ B_P]$, J_{P+1} is given by eqn. C.4, p. 286);

`XCON` – constants for simulation of the ST-LRP controller,

input: `m, nk, k, p, c,`

output: `yr` (reference sequence vector R given by eqn. 5.8, p. 154),

`cjp` ($\mathtt{cjp} = (J_{P+1} \otimes I_m)T$, where J_{P+1} is given by eqn. C.4 (p. 286) and T by eqn. 5.14, p. 155);

`XDIM` – constants for simulations,

input: `a,b,`

output: `m,n,`

`cuy` ($\mathtt{cuy} = S \otimes I_m$, where S is the $(\mathtt{n}+1) \times (\mathtt{n}+1)$ matrix given by eqn. 3.70, p. 81),

`cv` ($\mathtt{cv} = S \otimes I_m$, where S is the $(\mathtt{nk}+1) \times (\mathtt{nk}+1)$ matrix given by eqn. 3.70, p. 81);

XINI – initialization of simulations,

input: m,n,nk,nos,ea,eb,

output: up,yp,v,res,est,par,cov;

XLNI – left nilpotent interactor matrix $K_L(q)$ for a $m \times m$ strictly proper transfer matrix $H_{y,u}(z)$ (see Definition 2.3 (p. 43) and Theorem 1, Appendix C),

input: m,n,eb (eb $= N = [N_1 \ldots N_n]$ is the matrix of coefficients of the numerator polynomial matrix of the RMF description of $H_{y,u}(z)$ $= N(z^{-1})D^{-1}(z^{-1})$, with monic denominator matrix $D^{-1}(z^{-1})$),

output: k (matrix of coefficents of the LNI matrix k $= [K_0 \ldots K_k]$);

XLRP – control sequence U_T of the LRP controller, $P > 0$, $\Upsilon = I_m$, $S(p) = I_m$ (see eqn. 5.15, p. 156),

input: m,n,up,yp,cjp,p,c (p $= P + 1$, c $= C + 1$),

l ($\Lambda = lI_m$),

yr (from XCON),

al, be (from XAB),

output: xb (xb $= U_T$);

XMPE – control signal $u(t)$ of the MPE-RNI controller, $\Upsilon = I_m$, $\Lambda = 0$ (see eqns. 4.2 and 4.6, p. 120),

input: m,n,nk,yp,v,cv,vmx,k (k from XRNI),

yr ($m \times 1$ vector of the reference sequence value $r(t)$),

al, be (from XPRR),

output: xb (xb $= u(t)$);

XPRL – matrices $\alpha^{(0)}$, $\beta^{(0)}$ (see eqns. 3.10, 3.11 (p. 62), and Remark 3.1, p. 65),

input: m,n,nk,ea,eb,k (k from XLNI),

output: al,be (al $= \alpha^{(0)}$, be $= \beta^{(0)}$);

XPRR – matrices γ, δ (see eqns. 3.47, 3.48 (p. 72), and Remark 3.3, p. 74),

input: m,n,nk,ea,xp (xp $= \mathcal{B}^{(t)} = [\mathcal{B}_0^{(t)} \ldots \mathcal{B}_n^{(t)}]$ is the matrix of coefficients of $\mathcal{B}^{(t)}(z^{-1})$, see eqn. 2.66, p. 48),

output: `al,be` (`al` $= \gamma$, `be` $= \delta$);

`XRLS` – simplified multivariable RLS estimator with forgetting factor $\alpha(t) = 0.95$ (see eqns. E.18 to E.22, p. 301),

input: `m,n,up,yp`,

`cov,est` (`cov` $= P(t-2)$ or $P(-1)$, `est` $= \widehat{\Theta}(t-1)$ or $\widehat{\Theta}(0)$),

output: `cov,est` (`cov` $= P(t-1)$, `est` $= \widehat{\Theta}(t)$);

`XRNI` – right nilpotent interactor matrix $K_R(q)$ for a $m \times m$ strictly proper transfer matrix (see Definition 2.4 (p. 45) and Theorem 2.1, p. 47),

input: `m,n,eb`,

output: `k` (matrix of coefficents of the RNI matrix `k` $= [K_0 \ldots K_k]$),

`xp` (`xp` $= \mathcal{B}^{(t)} = [\mathcal{B}_0^{(t)} \ldots \mathcal{B}_n^{(t)}]$ is the matrix of coefficients of $\mathcal{B}^{(t)}(z^{-1})$, see eqn. 2.66, p. 48);

`XSAT` – limit on the control signal amplitude,

input: `m,umx,xb` (`xb` is either U_T from `XLRP`, or $u(t)$ from `XMPE` or `XWPE`),

output: `xb` ($u(t)$ such that $|u(t)|_i \leq$ `umx`, for $i = 1, \ldots,$ `m`);

`XSATV` – limit on the auxiliary control signal amplitude,

input: `m,vmx,xb` (`xb` $= u_k(t)$),

output: `xb` ($u_k(t)$ such that $|u_k(t)|_i \leq$ `vmx`, for $i = 1, \ldots,$ `m`);

`XSIM` – simulation of the system from the DARMA model,

input: `m,n,cuy,up,yp,a,b,xb` (`xb` $= U_T$ or $u(t)$),

output: `up,yp` (updated vectors of inputs and outputs);

`XSLR` – simulation of the ST-LRP controller, $P > 0$, $\Upsilon = I_m$, $S(p) = I_m$ (fixed reference sequence model for the square wave reference sequence),

input: `a,b,nk,p,c,umx,ea,eb,nos,rper`,

`k` (from `XLNI`),

`l` ($\Lambda = \mathtt{l} I_m$),

output: `res,par`;

`XSMP` – simulation of the ST-MPE-RNI controller, $\Upsilon = I_m$, $\Lambda = 0$ (fixed reference sequence model for the square wave reference sequence),

input: `a,b,umx,vmx,ea,eb,nos,rper,nk` (Note: `nk` is required *only* to set up the simulation),

output: `res,par`;

`XWPE` – control signal $u(t)$ of the MPE-RNI controller, $\Upsilon = I_m$, $\Lambda \neq 0$, $S(p) = I_m - pI_m$ (see eqns. 4.2 and 4.6, p. 120);

input: as for `XMPE`,

`l` ($\mathtt{l} = \Lambda$),

output: `xb` ($\mathtt{xb} = u(t)$);

`XXMP` – resultant of a polynomial matrix (macro developed in [79]),

input: `xn,xd` ($\mathtt{xn} = [s, t]$ and $\mathtt{xd} = D$, see eqn. B.5, p. 283),

output: `xd` ($\mathtt{xd} = \langle D \rangle_s$, see eqn. B.5, p. 283);

`XXRU` – right-upward shift block matrix (macro developed in [79]),

input: `xr` ($\mathtt{xr} = [i, j]$)

output: `xr` ($\mathtt{xr} = S' \otimes I_j$, where S is the $i \times i$ matrix given by eqn. 3.70).

Macro algorithms.

```
XAB='xd=[-ea(:,m+1:(n+1)*m);eye((n-1)*m) 0*ones((n-1)*m,m)];...
     xn=(diag(0*eye(p-1)))''.*.eye(m);xp=[xn,eb];be=[xn be];xb=al;...
     xn=be;xr=[n+p-1,m];]XXRU[;xr=xr''; for x=1:p-1,...
     xn=xn*xr+xb(:,1:m)*xp;be=[be;xn];xb=xb*xd;al=[al;xb];';
//
XCON='p=p+1;c=c+1;yr=diag(eye(p)).*.(k*diag(eye((nk+1)*m)));xp=...
      eye(p);xp=xp(:,p:-1:1);cjp=xp(:,c:p)*diag(eye(p-c+1)).*.eye(...
      m);if c>1,cjp=[xp(:,1:(c-1)).*.eye(m) cjp];';
//
XDIM='n=size(b);m=n(1);n=n(2)/n(1);xr=size(a);xr=[xr(2)/xr(1),...
```

```
          xr(1)];]XXRU[;cuy=xr'';xr=[nk+1,m];]XXRU[;cv=xr'';';
//
XINI='up=diag(0*eye((n+1)*m));yp=up;res=[0 0*diag(eye(3*m))''];...
          cov=10000*eye(2*n*m);est=[ea(:,m+1:(n+1)*m)'' ;eb''];...
          par=[0*diag(eye(m)) est''];nos=nos+1;v=0*diag(eye((nk+1)*m));';
//
XLNI='xp=0;k=eye(m,2*m);xj=eye(m);xj=xj(:,m:-1:1);xr=rank(eb(:,...
          1:m));while xr<m, [xd xn]=qr(eb(:,1:m));xd=xj*xd'';k=xd*k;...
          xa=[0*ones(xr,m);eye(2*m-xr,m)];xa=[xa(1:m,:) xa(m+1:2*m,...
          :)];xd=xd*eb;xn=[1,n-1];]XXMP[;eb=xa*xd;eb=eb(:,m+1:m*(n+1...
          ));xp=xp+1;xn=[1,xp];xd=k;]XXMP[;k=xa*xd;xr=rank(eb(:,1:m));';
//
XLRP='xa=be(:,1:p*m)*cjp;xb=inv(xa''*xa+eye(c*m)*l)*...
          xa''*(yr-be(:,p*m+1:(p+n-1)*m)*up(1:(n-1)*m)-al*yp(1:n*m));';
//
XMPE='xb=inv(be(:,1:m))*[yr-al*yp(1:n*m)-be(:,m+1:(n+nk)*m)*...
          v(1:(n+nk-1)*m)];]XSATV[;v=cv*v;v(1:m)=xb;xb=k*v;';
//
XPRL='xn=[nk-1,n];xd=ea;]XXMP[;xp=nk*m;be=k(:,1:xp)...
          *inv(xd(:,1:xp));al=-be*xd(:,xp+1:xp+n*m);xn=[nk-1,n-1];...
          xd=eb;]XXMP[;be=be*xd(:,xp-m+1:xp+(n-1)*m);';
//
XPRR='xn=[nk-1,n];xd=ea;]XXMP[;xa=inv(xd(:,1:nk*m));xa=xa(1:m,:);al...
          =-xa*xd(:,nk*m+1:(nk+n)*m);xn=[nk-1,n];xd=xp;]XXMP[;be=xa*xd;';
//
XRLS='xd=[-yp(m+1:(n+1)*m);up(1:n*m)];xn=1+xd''*cov*xd;est=est+...
          cov*xd*(yp(1:m)''-xd''*est)/xn;cov=(cov-cov*xd*xd''*cov/xn)...
          /0.95;ea=[eye(m) est(1:n*m,:)''];eb=est(n*m+1:2*n*m,:)'';';
//
XRNI='e=0;k=eye(m,2*m);jq=eye(m);jq=jq(:,m:-1:1);xp=[eb 0*eye(m)];...
```

```
        xr=rank(xp(:,1:m));while xr<m, [xd xn]=qr(xp(:,1:m)’’);...
        xa=xd*jq*[0*ones(m,xr) eye(m,2*m-xr)];xn=[n,1];xd=xa;]XXMP[;...
        xp=xp*xd;xp=xp(:,m+1:m*(n+2));e=e+1;xn=[e,1];xd=xa;...
        ]XXMP[;k=k*xd;xr=rank(xp(:,1:m));’;
//
XSAT=’for j=1:m, if abs(xb(j))>umx,xb(j)=xb(j)*umx/abs(xb(j));’;
//
XSATV=’for j=1:m, if abs(xb(j))>vmx,xb(j)=xb(j)*vmx/abs(xb(j));’;
//
XSIM=’up=cuy*up;up(1:m)=xb(1:m);xp=-a(:,(m+1):(n+1)*m)*yp(1:n*m)...
        +b*up(1:n*m);yp=cuy*yp;yp(1:m)=xp;’;
//
XSLR=’]XDIM[;]XINI[;]XCON[;v=ones(1,m);for i=1:nos, ]XPRL[;]XAB[;...
        ]XLRP[;]XSAT[;xa=[i-1 xb(1:m)’’ yp(1:m)’’ v],...
        res=[res;xa];]XSIM[;]XRLS[;par=[par;i*diag(eye(m)) est’’];...
        if round(i/rper)*rper=i,yr=-yr;v=-v;’;
//
XSMP=’]XDIM[;]XINI[;yr=diag(eye(m));for i=1:nos,]XRNI[;xr=size(k);...
        k=[0*ones(m,(nk+1)*m-xr(2)) k];]XPRR[;]XMPE[;]XSAT[;xa=[i-1...
        xb’’ yp(1:m)’’ yr’’],res=[res;xa];]XSIM[;]XRLS[;par=[par;...
        i*diag(eye(m)) est’’];if round(i/rper)*rper=i,yr=-yr;’;
//
XWPE=’xb=inv(l+be(:,1:m)’’*be(:,1:m))*[be(:,1:m)’’*(yr-al*...
        yp(1:n*m)-be(:,m+1:(n+nk)*m)*v(1:(n+nk-1)*m))+l*v(1:m)];...
        ]XSATV[;v=cv*v;v(1:m)=xb;xb=k*v;’;
//
XXMP=’xn=[xn,size(XD)];x=xn(2)+1;xr=[xn(1)+x,roun(xn(4)/x)];...
        XB=[XD,0*ones(xn(3),xn(1)*xr(2))];]XXRU[;XD=XB;...
        for x=1:xn(1),XB=XB*XR;XD=[XD;XB];’;
XXRU=’x=xr;XR=eye(x(1)+1);XR=XR(2:x(1)+1,1:x(1));XR=XR.*.eye(x(2));’;
```

Appendix E

Parameter estimation.

Recursive least squares estimator (RLS).

The recursive least-squares based estimator employed for adaptive prediction (subsection 3.2.2) and self-tuning control (subsection 6.1.2) is presented below.

Consider the regression model of a linear system $x(t) = \Theta'\phi(t-k)$, where $x(t)$ is the system (scalar) output [1], Θ is a vector of (unknown) system parameters [2], and $\phi(t-k)$ is the regression vector consisting of system inputs and outputs [3]. Denote the estimate of Θ by $\widehat{\Theta}(t)$, and define the equation error as [4]

$$e(t) = x(t) - \widehat{\Theta}(t-1)'\phi(t-k). \tag{E.1}$$

The recursive least squares RLS estimator (with exponential data weighting) is given by [9, chapters 3 and 4] (for $t \geq k$)

$$\begin{aligned}
\widehat{\Theta}(t) &= \widehat{\Theta}(t-1) + \frac{P(t-k-1)\phi(t-k)}{\alpha(t-1)+\phi(t-k)'P(t-k-1)\phi(t-k)}e(t), & \text{(E.2)}\\
P(t-k) &= \frac{1}{\alpha(t-1)} \times \\
&\times \left[P(t-k-1) - \frac{P(t-k-1)\phi(t-k)\phi(t-k)'P(t-k-1)}{\alpha(t-1)+\phi(t-k)'P(t-k-1)\phi(t-k)}\right] & \text{(E.3)}\\
P(-1) &= \epsilon I; \qquad 0 < \epsilon < \infty, & \text{(E.4)}
\end{aligned}$$

[1] See eqns. 3.81 (p. 86), 3.140 (p. 112), 6.14 (p. 201), 6.22 (p. 203), and 6.53 (p. 229).

[2] See eqns. 3.82 (p. 86), 3.141 (p. 112), 6.15 (p. 201), 6.23 (p. 203), and 6.54 (p. 229).

[3] See eqns. 3.83 (p. 86), 3.142 (p. 112), 6.16 (p. 201), 6.26 (p. 203), and 6.55 (p. 229).

[4] See eqns. 3.85 (p. 86), 3.144 (p. 113), 6.18 (p. 201), 6.30 (p. 204), and 6.57 (p. 229).

and $\widehat{\Theta}(0)$ is the initial parameter estimate vector. The choice of the forgetting factor (possibly time-varying) $0 < \alpha(t) \leq 1$ is discussed for instance in [146,11,147,148,13].

Recursive least squares estimator with condition number monitoring (RLS-CNM).

The RLS based estimator with condition number monitoring employed for adaptive prediction (subsection 3.3.4) and self-tuning control (subsection 6.2.2) is presented below.

The parameter estimate update equation of the RLS-CNM estimator (with exponential data weighting) is given by eqn. E.2 with the equation error defined by eqn. E.1. The following scheme, proposed in [57] [9, Remark 8.5.2, p. 333], is used for the covariance matrix update

$$\begin{aligned} P_m(t-k) &= \frac{1}{\alpha(t-1)} \times \\ &\quad \left[P(t-k-1) - \frac{P(t-k-1)\phi(t-k)\phi(t-k)'P(t-k-1)}{\alpha(t-1)+\phi(t-k)'P(t-k-1)\phi(t-k)}\right] \qquad \text{(E.5)} \\ \chi(t-k) &= \chi(t-k-1)\left[1+\phi(t-k)'P(t-k-1)\phi(t-k)\right], \qquad \text{(E.6)} \\ P(t-k) &= \begin{cases} P_m(t-k) & \text{if } \chi(t-k)\,\lambda_{\max}(P_m(t-k)) \leq C_\chi \\ \frac{C_\chi}{\chi(t-k)\,\lambda_{\max}(P_m(t-k))}P_m(t-k) & \text{otherwise} \end{cases} \qquad \text{(E.7)} \end{aligned}$$

where the constants $\chi(-1) > 0$, $0 < C_\chi < \infty$, and $\lambda_{\max}(P_m(t))$ denotes the maximum eigenvalue of matrix $P_m(t)$. This algorithm is referred to as the RLS with condition number monitoring because it ensures boundedness of the condition number of the parameter estimate error covariance matrix $P(t)$ for all t [9, Remark 8.5.2, p. 333].

For the purpose of convergence analysis, the covariance matrix update given by eqns. E.5 to E.7 (with no forgetting) can be expressed as follows [57, Lemma 3.1 (i)]

$$P(t-k) = \zeta(t-k)\left[P(t-k-1) - \frac{P(t-k-1)\phi(t-k)\phi(t-k)'P(t-k-1)}{1+\phi(t-k)'P(t-k-1)\phi(t-k)}\right], \qquad \text{(E.8)}$$

or

$$P(t-k)^{-1} = \zeta(t-k)^{-1}\left[P(t-k-1)^{-1} + \phi(t-k)\phi(t-k)'\right], \qquad \text{(E.9)}$$

where

$$\zeta(t-k) \overset{\text{def}}{=} \frac{C_\chi}{\max\left[C_\chi, \chi(t-k)\lambda_{\max}(P_m(t-k))\right]}, \tag{E.10}$$

or, for some $z(t)$ such that $0 \leq z(t) \leq 1$

$$\zeta(t-k) = \frac{1 + z(t-k)\phi(t-k)'P(t-k-1)\phi(t-k)}{1 + \phi(t-k)'P(t-k-1)\phi(t-k)}. \tag{E.11}$$

Note that if $\zeta(t-k) = 1$ then the ordinary RLS estimator results; if, however, the covariance matrix condition number becomes large, then the CNM scheme becomes active and $\zeta(t-k) < 1$.

Recursive extended least squares (RELS) estimator and recursive maximum likelihood (RML) estimator.

The estimation algorithm employed in the ST-MV-TISO-RELS and ST-MV-TISO-RML strategies (see subsection 6.2.3), is a version of the Pseudo Linear Regression algorithm [9, p. 319]. The parameter vector is defined as

$$\Theta_s = \left[\alpha_{s\omega}\ \ \beta_{s\omega}\ \ c_1\ \dots\ c_n\right]', \tag{E.12}$$

where $\alpha_{s\omega}$ and $\beta_{s\omega}$ are vectors of coefficients of polynomials of the k-step-ahead TISO-type predictor (see eqns. 3.137 to 3.139 (p. 108) for $k_p = k$) and c_i are coefficients of the noise polynomial $c_\omega(p)$ (eqn. 3.92, p. 93). The regression vector is defined as

$$\phi_s(t) = \left[y(t)\ \dots\ y(t-n+1)\ u(t)\ \dots\ u(t-n+1)\ \ -\widehat{y}(t+k-1)\ \dots\ -\widehat{y}(t+k-n)\right], \tag{E.13}$$

where the *a priori* output prediction $\widehat{y}(t)$ is given by

$$\widehat{y}(t) = \widehat{\Theta}_s(t-k)'\phi_s(t-k). \tag{E.14}$$

Define $e(t) = y(t) - \widehat{y}(t)$. Then the parameter estimate and covariance matrix update equations are given by (see [9, p. 444] for $D(t,p) = 1$)

$$\widehat{\Theta}_s(t) = \widehat{\Theta}_s(t-1) + \frac{P(t-k-1)\psi(t-k)}{\alpha(t-1) + \psi(t-k)'P(t-k-1)\psi(t-k)}e(t), \tag{E.15}$$

$$\begin{aligned} P(t-k) &= \frac{1}{\alpha(t-1)} \times \\ &\times \left[P(t-k-1) - \frac{P(t-k-1)\psi(t-k)\psi(t-k)'P(t-k-1)}{\alpha(t-1) + \psi(t-k)'P(t-k-1)\psi(t-k)}\right] \end{aligned} \tag{E.16}$$

where the gradient vector $\psi(t)$ is calculated from

$$D(t,p)\psi(t) = \phi_s(t). \tag{E.17}$$

If $D(t,p) = 1$, then the resulting algorithm is referred to as the RELS estimator [5]; if $D(t,p) = \widehat{c}_\omega(t,p)$, then the resulting algorithm is referred to as the RML estimator [6].

Simplified multivariable recursive least squares estimator.

The simplified multivariable recursive least squares (RLS) estimator is used in section 4.2 and subsection 5.1.2 for the estimation of parameters of the DARMA model given by eqns. 3.1 to 3.3, p. 59.

Let us define the $2mn \times m$ matrix of process parameters as

$$\Theta = \begin{bmatrix} A_1 & \dots & A_n & B_1 & \dots & B_n \end{bmatrix}'. \tag{E.18}$$

The $2mn \times 1$ regression vector is defined as

$$\phi(t) = \begin{bmatrix} -y(t)' & \dots & -y(t-n+1)' & u(t)' & \dots & u(t-n+1)' \end{bmatrix}'. \tag{E.19}$$

The $m \times 1$ equation error is

$$e(t) = y(t) - \widehat{\Theta}(t-1)'\phi(t-1), \tag{E.20}$$

where $\widehat{\Theta}(t)$ is a matrix of parameter estimates of Θ. Then the simplified RLS estimator (with exponential data weighting) is given by [9, p. 96]

$$\widehat{\Theta}(t) = \widehat{\Theta}(t-1) + \frac{P(t-2)\phi(t-1)}{\alpha(t-1) + \phi(t-1)'P(t-2)\phi(t-1)}e(t)', \tag{E.21}$$

$$P(t-1) = \frac{1}{\alpha(t-1)}\left[P(t-2) - \frac{P(t-2)\phi(t-1)\phi(t-1)'P(t-2)}{\alpha(t-1) + \phi(t-1)'P(t-2)\phi(t-1)}\right] \tag{E.22}$$

with the $2mn \times m$ initial parameter estimate matrix $\widehat{\Theta}(0)$, the initial covariance matrix $P(-1) = \epsilon I_{2mn}$, $(\epsilon > 0)$, and the forgetting factor $0 < \alpha(t) \leq 1$.

[5] The RELS method is also known under the following names: Panuska's method, extended matrix method, and RML1 [177]. Furthermore, if *a posteriori* output predictions replace the *a priori* predictions in the regression vector, the method is called AML [133].

[6] The RML method is also called the RML2 [175]. An interesting variant of the RML method with a posteriori prediction errors in the regression vector and the filter $D(t,p)$ modified to improve the convergence rate was proposed in [186].

Appendix F

Convergence proofs of the self-tuning minimum prediction error controllers for systems having the feedback configuration FD.

F.1 Proof of Lemma 6.2.

Comment F.1 *The proof of Lemma 6.2 (p. 205) is based on the methodology introduced in [32,34] [9, Theorem 6.3.2, p. 197] to prove global convergence of the MPE self tuners for systems having the feedback configuration FI.*

In the proof we shall make use of the so-called Key Technical Lemma *(KTL) [32, Lemma 3.1] [9, Lemma 6.2.1, p. 181]. In order to apply the KTL one needs to verify its assumptions. This is done as follows.*

Condition (1) ([32, eqn. 3.1] [9, eqn. 6.2.1, p. 181]). The analysis presented in the proof of [9, Theorem 6.3.2, p. 197] applies *mutatis mutandis* with

$$\mu(t) \overset{\text{def}}{=} y(t) + \overline{\lambda}\tilde{u}(t-k) - r(t). \tag{F.1}$$

Condition (1) of the KTL is satisfied with $s(t) = \mu(t)$, $b_1(t) = 1$, $b_2(t) = \lambda_{\max}(P(-1))$, where $\lambda_{\max}(P(-1))$ is the maximum eigenvalue of the initial covariance matrix $P(-1)$, and $\sigma(t) = \phi(t-k)$.

Condition (2): the uniform boundedness ([32, eqn. 3.2] [9, eqn. 6.2.2, p. 181]). This condition is satisfied since

$$0 < b_1(t) < C_1 < \infty \text{ and } 0 \leq b_2(t) < C_2 < \infty$$

for all $t \geq 1$.

Condition (3): linear boundedness ([32, eqn. 3.3] [9, eqn. 6.2.3, p. 181]). Multiplying both sides of eqn. F.1 by $t(p)d(p)$ one has

$$t(p)d(p)y(t) + \overline{\lambda}d(p)t(p)\tilde{u}(t-k) = t(p)d(p)\left[\mu(t) + r(t)\right].$$

Substituting from eqn. 3.61 (p. 79) for $y(t)$ and from eqn. 6.2 (p. 197) for $t(p)\tilde{u}(t)$ into the above equation one has

$$t(p)d(p)N(p)x_R(t) + \overline{\lambda}d(p)s(p)u(t-k) = t(p)d(p)\left[\mu(t) + r(t)\right].$$

Using eqns. 3.61 and 6.11 (p. 198) one has

$$w(p)u(t) = p^{-k}t(p)d(p)\left[\mu(t) + r(t)\right].$$

Similarly, starting with multiplication of eqn. F.1 by $t(p)N(p)$ one can show that

$$w(p)y(t) = p^{-k}t(p)N(p)\left[\mu(t) + r(t)\right];$$

and starting with multiplication by $t(p)N_F(p)$ one has

$$w(p)y_F(t) = p^{-k}t(p)N_F(p)\left[\mu(t) + r(t)\right].$$

It follows from Assumptions 3.5 (p. 79), 6.1 (iv) (p. 200), and [9, Lemma B.3.3, p. 486] that there exist finite nonnegative constants m_{u1}, m_{u2}, m_{y1}, m_{y2}, m_{y_F1}, and m_{y_F2} independent of N such that for all $1 \leq N \leq t$

$$\begin{aligned} u(N-k) &\leq m_{u1} + m_{u2} \max_{1\leq\tau\leq t} |\mu(\tau) + r(\tau)|, \\ y(N) &\leq m_{y1} + m_{y2} \max_{1\leq\tau\leq t} |\mu(\tau) + r(\tau)|, \\ y_F(N-k)_i &\leq m_{y_F1} + m_{y_F2} \max_{1\leq\tau\leq t} |\mu(\tau) + r(\tau)|, \end{aligned}$$

where $y_F(t)_i$ denotes the i-th element of the vector $y_F(t)$. Therefore, from eqn. 6.26 (p. 203) one has

$$\begin{aligned} \|\phi(t-k)\| &\leq c_3(m_{u1} + m_{u2} \max_{1\leq\tau\leq t} |\mu(\tau) + r(\tau)|) + \\ &(m_{y1} + m_{y2} \max_{1\leq\tau\leq t} |\mu(\tau) + r(\tau)|) + c_4(m_{y_F1} + m_{y_F2} \max_{1\leq\tau\leq t} |\mu(\tau) + r(\tau)|) \\ \leq\ & c_5(m_1 + m_2 \max_{1\leq\tau\leq t} |\mu(\tau) + r(\tau)|), \end{aligned}$$

where $c_3 = n_p + 1$, $c_4 = (n_p + 1)f$, $c_5 = c_3 + c_4 + 1$, $m_1 = m_{u1} + m_{y1} + m_{y_F1}$, and $m_2 = \max(m_{u2}, m_{y2}, m_{y_F2})$. Furthermore,

$$\begin{aligned} \|\phi(t-k)\| &\leq c_5[m_1 + m_2 \max_{1\leq\tau\leq t}(|\mu(\tau)| + |r(\tau)|)] \\ &\leq c_5[(m_1 + m_2 m_3) + m_2 \max_{1\leq\tau\leq t} |\mu(\tau)|] = c_1 + c_2 \max_{1\leq\tau\leq t} |\mu(\tau)|, \end{aligned}$$

since the reference sequence is bounded, i.e., $|r(\tau)| \leq m_3$ for $1 \leq \tau \leq t$. Hence, the linear boundedness condition of the KTL is satisfied.

We can now apply the KTL to conclude that the sequence $\{\|\phi(t)\|\}$ is bounded, which establishes part (a) of Lemma 6.2. From eqns. 6.22, 6.28 and 6.23 (p. 203) one has

$$u(t) - u^o(t) = \rho\mu(t+k),$$

and part (b) of Lemma 6.2 follows from the KTL part (i).

Q.E.D.

F.2 Proof of Theorem 6.2.

Comment F.2 *The proof presented below is based on the methodology introduced in [55,57] [9, chapter 9 and 11] for establishing the convergence of stochastic self-tuning control algorithms for systems having the feedback configuration FI. This approach is adopted here to prove the convergence of the stochastic self-tuning strategy (ST-MV-TIMO) for systems having the feedback configuration FD.*

Let us define some error quantities.

Prediction error for the k-step-ahead TIMO-type predictor, $\epsilon(t)$

$$\epsilon(t) \overset{\text{def}}{=} y(t) - y^o(t|t-k). \tag{F.2}$$

From eqn. 3.103 (p. 96)

$$\epsilon(t) = F(p)\omega(t) = \sum_{i=1}^{k} f_{i-1}\omega(t-i+1). \tag{F.3}$$

It follows from the smoothing property of conditional expectations [9, p. 498] using the fact $\mathcal{F}_{t-k} \subset \mathcal{F}_{t-i}$ for $i = 1, \ldots, k$, and from eqn. 3.95 (p. 93) that

$$E\{\epsilon(t)|\mathcal{F}_{t-k}\} = E\left\{\sum_{i=1}^{k} f_{i-1}E\{\omega(t-i+1)|\mathcal{F}_{t-i}\}|\mathcal{F}_{t-k}\right\} = 0 \quad a.s. \tag{F.4}$$

Furthermore, using eqn. 3.96 (p. 93) one has

$$E\{\epsilon(t)^2|\mathcal{F}_{t-k}\} = E\left\{E\{\left[\sum_{i=1}^{k} f_{i-1}\omega(t-i+1)\right]^2 |\mathcal{F}_{t-i}\}|\mathcal{F}_{t-k}\right\} = FF'\sigma^2 \quad a.s. \tag{F.5}$$

Finally,

$$\begin{aligned} \limsup_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} \epsilon(t)^2 &= \limsup_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N}\left[\sum_{i=1}^{k} f_{i-1}\omega(t-i+1)\right]^2 \\ \le \limsup_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} k\sum_{i=1}^{k} f_{i-1}^2\omega(t-i+1)^2 &< \infty \quad a.s., \end{aligned} \tag{F.6}$$

using Schwarz inequality and eqn. 3.146 (p. 113).

The *a posteriori* prediction error, $\eta(t)$

$$\eta(t) \overset{\text{def}}{=} y(t) - \tilde{y}(t) = y(t) - \widehat{\Theta}(t)'\phi(t-k), \tag{F.7}$$

where $\tilde{y}(t)$ denotes the *a posteriori* prediction.

The closed-loop system tracking error, $e_r(t)$

$$e_r(t) \overset{\text{def}}{=} y(t) - r(t), \tag{F.8}$$

which can be also written using the ST-MV-TIMO control law 6.58 (p. 229) as

$$e_r(t) = y(t) - \widehat{\Theta}(t-k)'\phi(t-k). \tag{F.9}$$

The parameter estimates error, $\overline{\Theta}(t)$

$$\overline{\Theta}(t) \stackrel{\text{def}}{=} \widehat{\Theta}(t) - \Theta. \tag{F.10}$$

Comment F.3 *In order to prove the global convergence of the ST-MV-TIMO strategy we shall make use of the* Stochastic Key Technical Lemma *(SKTL) [9, Lemma 8.5.3, p. 334]. In order to apply the SKTL one needs to*

- *establish certain relations between, and decompositions of, the error quantitics defined above (see eqns. F.12, F.15, F.16, F.18, to F.21);*
- *show that some quantities are measurable (see Lemma F.1, p. 308);*
- *establish some convergence results for the estimator (see Lemma F.2, p. 309);*
- *prove the Normalized Prediction Error Convergence (NPEC) result (see Lemma F.3, p. 314);*
- *establish certain boundedness properties of some input, outputs, and error quantities (see Lemma F.4, p. 315).*

Let us define

$$b(t) \stackrel{\text{def}}{=} -\overline{\Theta}(t)'\phi(t-k). \tag{F.11}$$

Subtracting $\tilde{y}(t)$ from both sides of eqn. F.2 and using eqn. 6.53 (p. 229) one has

$$y(t) - \tilde{y}(t) = \Theta'\phi(t-k) + \epsilon(t) - \tilde{y}(t).$$

This can be rewritten using eqn. F.7 as $\eta(t) - \epsilon(t) = \Theta'\phi(t-k) - \widehat{\Theta}(t)'\phi(t-k)$, or using eqns. F.10 and F.11

$$b(t) = \eta(t) - \epsilon(t). \tag{F.12}$$

Now the decompositions of $\overline{\Theta}(t)$, $\eta(t)$, and $b(t)$ will be introduced. Subtracting Θ for both sides of eqn. E.2 (p. 298) with the equation error $e(t)$ defined by eqn. 6.57 (p. 229), and using eqn. F.10 one has

$$\overline{\Theta}(t) = \overline{\Theta}(t-1) + \frac{P(t-k-1)\phi(t-k)}{1+\phi(t-k)'P(t-k-1)\phi(t-k)}\left[y(t) - \widehat{\Theta}(t-1)'\phi(t-k)\right]. \tag{F.13}$$

Using eqns. F.2 and 6.53 (p. 229), output $y(t)$ can be expressed as

$$y(t) = y^o(t|t-k) + \epsilon(t) = \Theta'\phi(t-k) + \epsilon(t).$$

Thus

$$\begin{aligned} \overline{\Theta}(t) &= \overline{\Theta}(t-1) + \frac{P(t-k-1)\phi(t-k)}{1+\phi(t-k)'P(t-k-1)\phi(t-k)}\left[-\phi(t-k)'\overline{\Theta}(t-1) + \epsilon(t)\right] \\ &= \overline{\Theta}(t-1) - \frac{P(t-k-1)\phi(t-k)\phi(t-k)'}{1+\phi(t-k)'P(t-k-1)\phi(t-k)}\overline{\Theta}(t-1) + \\ &\quad + \frac{P(t-k-1)\phi(t-k)}{1+\phi(t-k)'P(t-k-1)\phi(t-k)}\sum_{i=1}^{k} f_{i-1}\omega(t-i+1), \end{aligned} \tag{F.14}$$

using eqn. F.3. Let us now decompose $\overline{\Theta}(t)$ as follows:

$$\overline{\Theta}(t) = \sum_{i=1}^{k} \overline{\Theta}^{(i)}(t), \tag{F.15}$$

where from eqn. F.14

$$\begin{aligned} \overline{\Theta}^{(i)}(t) &\stackrel{\text{def}}{=} \overline{\Theta}^{(i)}(t-1) - \frac{P(t-k-1)\phi(t-k)\phi(t-k)'}{1+\phi(t-k)'P(t-k-1)\phi(t-k)}\overline{\Theta}^{(i)}(t-1) + \\ &\quad + \frac{P(t-k-1)\phi(t-k)}{1+\phi(t-k)'P(t-k-1)\phi(t-k)} f_{i-1}\omega(t-i+1). \end{aligned} \tag{F.16}$$

Similarly, from eqns. F.7 and F.2 one has

$$\begin{aligned} \eta(t) &= \Theta'\phi(t-k) + \epsilon(t) - \hat{\Theta}(t)'\phi(t-k) = -\overline{\Theta}(t)'\phi(t-k) + \epsilon(t) \\ &= -\left(\sum_{i=1}^{k} \overline{\Theta}^{(i)}(t)\right)'\phi(t-k) + \sum_{i=1}^{k} f_{i-1}\omega(t-i+1), \end{aligned} \tag{F.17}$$

using eqns. F.3 and F.15. Hence, denote

$$\eta(t) = \sum_{i=1}^{k} \eta^{(i)}(t), \tag{F.18}$$

where (from eqn. F.17)

$$\eta^{(i)}(t) \stackrel{\text{def}}{=} -\overline{\Theta}^{(i)}(t)'\phi(t-k) + f_{i-1}\omega(t-i+1). \tag{F.19}$$

Finally decompose $b(t)$ as

$$b(t) = \sum_{i=1}^{k} b^{(i)}(t), \tag{F.20}$$

where

$$b^{(i)}(t) \stackrel{\text{def}}{=} -\overline{\Theta}^{(i)}(t)'\phi(t-k). \tag{F.21}$$

We shall now establish the following result.

Lemma F.1 *For $i = 1, \ldots, k$*

(i) $\phi(t-k)$ *is* $\mathcal{F}_{t-k}$ *measurable;*

(ii) $\overline{\Theta}^{(i)}(t-1)$ *is* $\mathcal{F}_{t-i}$ *measurable;*

(iii) $\eta^{(i)}(t-1)$ *is* $\mathcal{F}_{t-i}$ *measurable;*

(iv) $b^{(i)}(t-1)$ *is* $\mathcal{F}_{t-i}$ *measurable.*

Proof.

Part (i): follows from definition of $\phi(t)$ (see eqn. 6.55, p. 229).

Part (ii): it follows from eqn. F.16 that

$$\begin{aligned}\overline{\Theta}^{(i)}(t-1) &= \overline{\Theta}^{(i)}(t-2) - \frac{P(t-k-2)\phi(t-k-1)\phi(t-k-1)'}{1+\phi(t-k-1)'P(t-k-2)\phi(t-k-1)}\overline{\Theta}^{(i)}(t-2) + \\ &\quad + \frac{P(t-k-2)\phi(t-k-1)}{1+\phi(t-k-1)'P(t-k-2)\phi(t-k-1)}f_{i-1}\omega(t-i). \qquad \text{(F.22)}\end{aligned}$$

Note from eqns. F.2 and F.3 that

$$\omega(t-i) = y(t-i) - y^o(t-i|t-i-1), \qquad \text{(F.23)}$$

where $y^o(t-i|t-i-1)$ is the one-step-ahead optimal prediction of output $y(t-i)$. If $k=1$, then the one-step-ahead optimal TIMO-type prediction can be expressed as $y^o(t-i|t-i-1) = \Theta'\phi(t-i-1)$, where $\Theta = [\alpha_\omega \; \beta_\omega]'$ (see Lemma 3.5 (p. 94) for $k_p = k = 1$). If $k > 1$, then the prediction can be expressed as $y^o(t-i|t-i-1) = \Theta_1'\phi(t-i-1)$, where $\Theta_1 = [\alpha_\omega^1 \; 0 \; \beta_\omega^1]'$, and α_ω^1 and β_ω^1 are vectors of coefficients of polynomials of the one-step-ahead TIMO-type predictor (see Lemma 3.5 for $k_p = 1 < k$). (Note that zero element in vector Θ_1 is introduced so that Θ_1 is conformable with $\phi(t)$.) Substituting from eqn. F.23 for $\omega(t-i)$ in eqn. F.22 with $y^o(t-i|t-i-1) = \Theta'\phi(t-i-1)$ if $k=1$ or $y^o(t-i|t-i-1) = \Theta_1'\phi(t-i-1)$ if $k>1$, one can see that $\overline{\Theta}^{(i)}(t-1)$ is $\mathcal{F}_{t-i}$ measurable, because $\phi(t-i-1)$ is $\mathcal{F}_{t-i-1}$ measurable, and $\mathcal{F}_{t-i-1} \subset \mathcal{F}_{t-i}$.

Part (iii): from eqns. F.19 and F.23 one has

$$\eta^{(i)}(t-1) = -\overline{\Theta}^{(i)}(t-i)'\phi(t-k-1) + f_{i-1}\left[y(t-i) - y^o(t-i|t-i-1)\right].$$

Substituting for $y^o(t-i|t-i-1) = \Theta'\phi(t-i-1)$ if $k=1$, or $y^o(t-i|t-i-1) = \Theta_1'\phi(t-i-1)$ if $k>1$, one can see that $\eta^{(i)}(t-1)$ is $\mathcal{F}_{t-i}$ measurable, because $\phi(t-i-1)$ is $\mathcal{F}_{t-i-1}$ measurable, and $\mathcal{F}_{t-i-1} \subset \mathcal{F}_{t-i}$, and $\overline{\Theta}^{(i)}(t-1)$ is $\mathcal{F}_{t-i}$ measurable (from part (ii)).

Part (iv): from eqn. F.21 one has

$$b^{(i)}(t-1) = -\overline{\Theta}^{(i)}(t-1)'\phi(t-k-1),$$

and $\phi(t-k-1)$ is $\mathcal{F}_{t-k-1}$ measurable, and $\mathcal{F}_{t-k-1} \subset \mathcal{F}_{t-i}$, and $\overline{\Theta}^{(i)}(t-i)$ is $\mathcal{F}_{t-i}$ measurable (from part (ii)). Hence, $b^{(i)}(t-1)$ is $\mathcal{F}_{t-i}$ measurable.

Q.E.D.

Comment F.4 *Note that in the proof of Lemma F.1, the optimal one-step-ahead TIMO-type predictor was employed. The existence of this predictor for $k \geq 1$ is guaranteed in view of Lemma 3.5 (p. 94).*

We need the following result in order to establish the NPEC.

Lemma F.2 *Subject to Assumptions 3.7 to 3.10 (p. 91), 3.11 (p. 113) and 6.4 (p. 228), the ST-MV-TIMO strategy ensures*

$$\lim_{N\to\infty} \sum_{t=k}^{N} \frac{\phi(t-k)'P(t-k-1)\phi(t-k)}{\chi(t-k-1)}\eta(t)^2 < \infty \quad a.s., \tag{F.24}$$

$$\lim_{N\to\infty} \sum_{t=k}^{N} \frac{b(t)^2}{\chi(t-k-1)} < \infty \quad a.s. \tag{F.25}$$

Proof.

We shall show at first that

$$\lim_{N\to\infty} \sum_{j=k}^{N} \frac{\phi(j-k)'P(j-k-1)\phi(j-k)}{\chi(j-k-1)}\eta^{(i)}(j)^2 < \infty \quad a.s., \tag{F.26}$$

$$\lim_{N\to\infty} \sum_{j=k}^{N} \frac{b^{(i)}(j)^2}{\chi(j-k-1)} < \infty \quad a.s. \tag{F.27}$$

From eqn. F.7 and E.2 (p. 298) one has

$$\begin{aligned}\eta(t) &= y(t)-\widehat{\Theta}(t)'\phi(t-k) = y(t)-\widehat{\Theta}(t-1)'\phi(t-k) - \\ &-\frac{\phi(t-k)'P(t-k-1)\phi(t-k)}{1+\phi(t-k)'P(t-k-1)\phi(t-k)}\left[y(t)-\widehat{\Theta}(t-1)'\phi(t-k)\right] \\ &= \frac{y(t)-\widehat{\Theta}(t-1)'\phi(t-k)}{1+\phi(t-k)'P(t-k-1)\phi(t-k)}.\end{aligned}$$

Substituting the above equation into eqn. F.13 one has

$$\overline{\Theta}(t) = \overline{\Theta}(t-1)+P(t-k-1)\phi(t-k)\eta(t).$$

Using decompositions introduced in eqns. F.15 and F.18 one has for $i = 1,\ldots,k$

$$\overline{\Theta}^{(i)}(t) = \overline{\Theta}^{(i)}(t-1)+P(t-k-1)\phi(t-k)\eta^{(i)}(t). \tag{F.28}$$

Multiplying both sides of the above equation by $\overline{\Theta}^{(i)}(t)'P(t-k-1)^{-1}$ one has

$$\overline{\Theta}^{(i)}(t)'P(t-k-1)^{-1}\overline{\Theta}^{(i)}(t) = \overline{\Theta}^{(i)}(t)'P(t-k-1)^{-1}\overline{\Theta}^{(i)}(t-1)+\overline{\Theta}^{(i)}(t)'\phi(t-k)\eta^{(i)}(t).$$

Substituting from eqn. F.28 for $\overline{\Theta}^{(i)}(t)'$ in the first term on the right hand side of the above equation and then for $\overline{\Theta}^{(i)}(t-1)$ from eqn. F.28, one has

$$\begin{aligned}&\overline{\Theta}^{(i)}(t)'P(t-k-1)^{-1}\overline{\Theta}^{(i)}(t) = \overline{\Theta}^{(i)}(t-1)'P(t-k-1)^{-1}\overline{\Theta}^{(i)}(t-1)+ \\ &+\phi(t-k)'\overline{\Theta}^{(i)}(t)\eta^{(i)}(t)-\phi(t-k)'P(t-k-1)\phi(t-k)\eta^{(i)}(t)^2+ \\ &+\overline{\Theta}^{(i)}(t)'\phi(t-k)\eta^{(i)}(t) \\ &= \overline{\Theta}^{(i)}(t-1)'P(t-k-1)^{-1}\overline{\Theta}^{(i)}(t-1)+2\overline{\Theta}^{(i)}(t)'\phi(t-k)\eta^{(i)}(t)- \\ &-\phi(t-k)'P(t-k-1)\phi(t-k)\eta^{(i)}(t)^2.\end{aligned}$$

Now substituting from eqn. E.9 (p. 299) for $P(t-k-1)^{-1}$ one has

$$\begin{aligned}&\zeta(t-k)\overline{\Theta}^{(i)}(t)'P(t-k)^{-1}\overline{\Theta}^{(i)}(t) = \overline{\Theta}^{(i)}(t-1)P(t-k-1)^{-1}\overline{\Theta}^{(i)}(t-1)+ \\ &+\overline{\Theta}^{(i)}(t)'\phi(t-k)\phi(t-k)'\overline{\Theta}^{(i)}(t)+2\overline{\Theta}^{(i)}(t)'\phi(t-k)\eta^{(i)}(t)- \\ &-\phi(t-k)'P(t-k-1)\phi(t-k)\eta^{(i)}(t)^2.\end{aligned} \tag{F.29}$$

Let us define the following (non-negative) Lyapunov function:

$$V^{(i)}(t) \overset{\text{def}}{=} \overline{\Theta}^{(i)}(t)'P(t-k)^{-1}\overline{\Theta}^{(i)}(t). \tag{F.30}$$

Eqn. F.29 can be written using eqns. F.30 and F.21 as

$$\begin{aligned}&\zeta(t-k)V^{(i)}(t)\\ =\;& V^{(i)}(t-1)+b^{(i)}(t)^2-2b^{(i)}(t)\eta^{(i)}(t)-\phi(t-k)'P(t-k-1)\phi(t-k)\eta^{(i)}(t)^2.\end{aligned}$$

It follows from eqn. F.12 using decompositions given by eqns. F.18 and F.20, and from eqn. F.3 that for $i=1,\ldots,k$

$$\eta^{(i)}(t)=b^{(i)}(t)+f_{i-1}\omega(t-i+1).$$

Using the above equation one has

$$\begin{aligned}\zeta(t-k)V^{(i)}(t) &= V^{(i)}(t-1)-b^{(i)}(t)^2-2f_{i-1}b^{(i)}(t)\omega(t-i+1)-\\ &\quad -\phi(t-k)'P(t-k-1)\phi(t-k)\eta^{(i)}(t)^2.\end{aligned}$$

Taking the conditional expectations for both sides of the above equation one has

$$\begin{aligned}&\zeta(t-k)E\left\{V^{(i)}(t)|\mathcal{F}_{t-i}\right\}=E\left\{V^{(i)}(t-1)|\mathcal{F}_{t-i}\right\}-E\left\{b^{(i)}(t)^2|\mathcal{F}_{t-i}\right\}-\\ &-2f_{i-1}E\left\{b^{(i)}(t)\omega(t-i+1)|\mathcal{F}_{t-i}\right\}-\\ &-E\left\{\phi(t-k)'P(t-k-1)\phi(t-k)\eta^{(i)}(t)^2|\mathcal{F}_{t-i}\right\}.\end{aligned}$$

We shall now use the following result [57, Lemma 3.2, (iii)]

$$E\left\{b^{(i)}(t)\omega(t-i+1)|\mathcal{F}_{t-i}\right\}=-\frac{\phi(t-k)'P(t-k-1)\phi(t-k)}{1+\phi(t-k)'P(t-k-1)\phi(t-k)}f_{i-1}\sigma^2,$$

which leads to

$$\begin{aligned}&\zeta(t-k)E\left\{V^{(i)}(t)|\mathcal{F}_{t-i}\right\}=E\left\{V^{(i)}(t-1)|\mathcal{F}_{t-i}\right\}-E\left\{b^{(i)}(t)^2|\mathcal{F}_{t-i}\right\}+\\ &+2\frac{\phi(t-k)'P(t-k-1)\phi(t-k)}{1+\phi(t-k)'P(t-k-1)\phi(t-k)}f_{i-1}^2\sigma^2-\\ &-E\left\{\phi(t-k)'P(t-k-1)\phi(t-k)\eta^{(i)}(t)^2|\mathcal{F}_{t-i}\right\}.\end{aligned}$$

Substituting from eqn. E.11 (p. 300) for $\zeta(t-k)$ into the above equation, dividing the above equation by $\chi(t-k-1)$, and substituting for $\chi(t-k-1)$ from eqn. E.6 (p. 299), one has

$$[1+z(t-k)\phi(t-k)'P(t-k-1)\phi(t-k)]\,E\left\{\frac{V^{(i)}(t)}{\chi(t-k)}|\mathcal{F}_{t-i}\right\}$$

$$\begin{aligned} = \quad & \frac{V^{(i)}(t-1)}{\chi(t-k-1)} - E\left\{\frac{b^{(i)}(t)^2}{\chi(t-k-1)}|\mathcal{F}_{t-i}\right\} \\ & -E\left\{\frac{\phi(t-k)'P(t-k-1)\phi(t-k)}{\chi(t-k-1)}\eta^{(i)}(t)^2|\mathcal{F}_{t-i}\right\} + \\ & +2\frac{\phi(t-k)'P(t-k-1)\phi(t-k)}{\chi(t-k)}f_{i-1}^2\sigma^2. \end{aligned} \tag{F.31}$$

Let us now define

$$\begin{aligned} X^{(i)}(t) \overset{\text{def}}{=} & \frac{V^{(i)}(t)}{\chi(t-k)} + \sum_{j=k}^{t}\frac{b^{(i)}(j)^2}{\chi(j-k-1)} + \\ & +\sum_{j=k}^{t}\frac{\phi(j-k)'P(j-k-1)\phi(j-k)}{\chi(j-k-1)}\eta^{(i)}(j)^2 + \frac{K}{\chi(t-k)} + \\ & +\sum_{j=k}^{t} z(j-k)\frac{\phi(j-k)'P(j-k-1)\phi(j-k)}{\chi(j-k)}V^{(i)}(j), \end{aligned} \tag{F.32}$$

for some constant K, $0 \leq K < \infty$. Taking conditional expectations one has

$$\begin{aligned} E\left\{X^{(i)}(t)|\mathcal{F}_{t-i}\right\} = & E\left\{\frac{V^{(i)}(t)}{\chi(t-k)}|\mathcal{F}_{t-i}\right\} + E\left\{\sum_{j=k}^{t}\frac{b^{(i)}(j)^2}{\chi(j-k-1)}|\mathcal{F}_{t-i}\right\} + \\ +E & \left\{\sum_{j=k}^{t}\frac{\phi(j-k)'P(j-k-1)\phi(j-k)}{\chi(j-k-1)}\eta^{(i)}(j)^2|\mathcal{F}_{t-i}\right\} + E\left\{\frac{K}{\chi(t-k)}|\mathcal{F}_{t-i}\right\} + \\ +E & \left\{\sum_{j=k}^{t} z(j-k)\frac{\phi(j-k)'P(j-k-1)\phi(j-k)}{\chi(j-k)}V^{(i)}(j)|\mathcal{F}_{t-i}\right\}. \end{aligned} \tag{F.33}$$

Consider the last term on the right hand side of eqn. F.33

$$\begin{aligned} & E\left\{\sum_{j=k}^{t} z(j-k)\frac{\phi(j-k)'P(j-k-1)\phi(j-k)}{\chi(j-k)}V^{(i)}(j)|\mathcal{F}_{t-i}\right\} \\ = \quad & E\left\{\sum_{j=k}^{t-1} z(j-k)\frac{\phi(j-k)'P(j-k-1)\phi(j-k)}{\chi(j-k)}V^{(i)}(j)|\mathcal{F}_{t-i}\right\} + \\ & +E\left\{z(t-k)\frac{\phi(t-k)'P(t-k-1)\phi(t-k)}{\chi(t-k)}V^{(i)}(t)|\mathcal{F}_{t-i}\right\} \\ = \quad & \sum_{j=k}^{t-1} z(j-k)\frac{\phi(j-k)'P(j-k-1)\phi(j-k)}{\chi(j-k)}V^{(i)}(j) + \\ & +z(t-k)\frac{\phi(t-k)'P(t-k-1)\phi(t-k)}{\chi(t-k)}E\left\{V^{(i)}(t)|\mathcal{F}_{t-i}\right\}, \end{aligned} \tag{F.34}$$

using the fact that $\mathcal{F}_{t-j} \subset \mathcal{F}_{t-i}$ and $\phi(t-j)$ is $\mathcal{F}_{t-j}$ measurable, and $\overline{\Theta}^{(i)}(t-1)$ is $\mathcal{F}_{t-i}$ measurable (Lemma F.1 (ii)), $i = 1, \ldots, k$, and $j = k, \ldots, t-1$.

Substituting eqn. F.34 into F.33 and using eqn. F.31 one has

$$\begin{aligned}
&E\left\{X^{(i)}(t)|\mathcal{F}_{t-i}\right\} = \frac{V^{(i)}(t-1)}{\chi(t-k-1)} + E\left\{\sum_{j=k}^{t-1}\frac{b^{(i)}(j)^2}{\chi(j-k-1)}|\mathcal{F}_{t-i}\right\} + \\
&+E\left\{\sum_{j=k}^{t-1}\frac{\phi(j-k)'P(j-k-1)\phi(j-k)}{\chi(j-k-1)}\eta^{(i)}(j)^2|\mathcal{F}_{t-i}\right\} + \\
&+\frac{K}{\chi(t-k)} + \sum_{j=k}^{t-1} z(j-k)\frac{\phi(j-k)'P(j-k-1)\phi(j-k)}{\chi(j-k)}V^{(i)}(j) + \\
&+2\frac{\phi(t-k)'P(t-k-1)\phi(t-k)}{\chi(t-k)}f_{i-1}^2\sigma^2.
\end{aligned}$$

Now using the fact that $b^{(i)}(t-1)$ and $\eta^{(i)}(t-1)$ are $\mathcal{F}_{t-i}$ measurable (Lemma F.1 (iv) and (iii)), and using eqn. F.32, and since $\{\chi(t)\}$ is a monotonic sequence (see eqn. E.6, p. 299), one has for $i = 1, \ldots, k$

$$E\left\{X^{(i)}(t)|\mathcal{F}_{t-i}\right\} \leq X^{(i)}(t-1) + 2\frac{\phi(t-k)'P(t-k-1)\phi(t-k)}{\chi(t-k)}f_{i-1}^2\sigma^2 \quad a.s. \quad (F.35)$$

Furthermore, it can be shown that [57, Lemma 3.3]

$$\sum_{t=k}^{\infty}\frac{\phi(t-k)'P(t-k-1)\phi(t-k)}{\chi(t-k)}f_{i-1}^2\sigma^2 < \infty \quad a.s. \quad \text{for} \quad i = 1, \ldots, k.$$

Hence, one can apply the Martingale Convergence Theorem in the form given in [9, Corollary D.5.1, p. 501] (see also [171, p. 34] [133]) to conclude that the sequence $\{X^{(i)}(t)\}$ converges, i.e.,

$$X^{(i)}(t) \rightarrow X^{(i)} < \infty \quad a.s. \quad \text{for} \quad i = 1, \ldots, k. \quad (F.36)$$

Then eqns. F.26 and F.27 follow from eqns. F.36 and F.32. Finally eqns. F.24 and F.25 follow from eqns. F.26 and F.27 using Schwarz inequality.

Q.E.D.

Comment F.5 *The essential difference in convergence analysis of the ST-MV-TIMO strategy and self-tuning strategies for systems having the feedback configuration FI is revealed in the above proof of Lemma F.2. Namely, the relations F.24 and F.25 are established without any assumption concerning strict positive realness of a transfer function involving the noise polynomial $c_\omega(z^{-1})$ (see also Comment 6.10, p. 235).*

Now the NPEC result will be proved.

Lemma F.3 *For the ST-MV-TIMO strategy, subject to Assumptions 3.7 to 3.10 (p. 91), 3.11 (p. 113), and 6.4 (p. 228), the following Normalized Prediction Error Convergence (NPEC) result holds*

$$\lim_{N\to\infty} \sum_{t=k}^{N} \frac{[e_r(t) - \epsilon(t)]^2}{\overline{\chi}(t-k)} < \infty \quad a.s., \tag{F.37}$$

where

$$\overline{\chi}(t-k) = \overline{\chi}(t-k-1) + \phi(t-k)'\phi(t-k), \tag{F.38}$$

with $\overline{\chi}(-1) = (C_\chi)^{-1}\chi(-1)$, *where* C_χ *and* $\chi(-1)$ *are the initial parameters of the CNM scheme (see Appendix E).*

Proof.

The analysis given in the proof of [9, Theorem 8.5.1, p. 326] applies *mutatis mutandis* here. First, the following result is established

$$\lim_{N\to\infty} \sum_{t=k}^{N} \frac{[e_r(t) - \epsilon(t)]^2}{\chi(t-k)} < \infty \quad a.s.$$

using Schwarz inequality, triangle inequality and Lemma F.2. In order to establish eqn. F.37 from the above equation, one needs the relation between $\overline{\chi}(t)$ and $\chi(t)$ given in [57, Lemma 3.1, part (ii) and (iii)].

Q.E.D.

Note that the NPEC result (eqn. F.37) verifies the first assumption of the Stochastic Key Technical Lemma (SKTL) (see [9, eqn. 8.5.70, p. 333]). (It is the proof of the NPEC which requires the boundedness of the condition number of the estimator covariance matrix $P(t)$ for all t. This is guaranteed by the condition number monitoring (CNM) technique (see eqns. E.5 to E.7, Appendix E).)

Finally we need to establish certain input, outputs and error boundedness conditions given in the following lemma.

Lemma F.4 *There exist finite positive constants K_1 to K_8, N_1 to N_3 and $\overline{N}$, such that*

$$\begin{aligned}
\frac{1}{N}\sum_{t=1}^{N-k} u(t)^2 &\leq \frac{K_1}{N}\sum_{t=k}^{N} y(t)^2 + K_2, \quad \text{for} \quad N > N_1 \quad a.s., & \text{(F.39)}\\
\frac{1}{N}\sum_{t=k}^{N} y(t)^2 &\leq \frac{K_3}{N}\sum_{t=k}^{N} [e_r(t) - \epsilon(t)]^2 + K_4, \quad \text{for} \quad N > N_2 \quad a.s., & \text{(F.40)}\\
\frac{1}{N}\sum_{t=1}^{N} \|y_F(t)\|^2 &\leq \frac{K_5}{N}\sum_{t=0}^{N} u(t)^2 + K_6, \quad \text{for} \quad N > N_3 \quad a.s., & \text{(F.41)}\\
\frac{\overline{\chi}(N-k)}{N} &\leq \frac{K_7}{N}\sum_{t=k}^{N} [e_r(t) - \epsilon(t)]^2 + K_8, \quad \text{for} \quad N > \overline{N} \quad a.s., & \text{(F.42)}
\end{aligned}$$

where $\overline{\chi}(N-k)$ is defined by eqn. F.38.

Proof.

The boundedness of the control sequence $\{u(t)\}$ (eqn. F.39) and of the controlled output sequence $\{y(t)\}$ (eqn. F.40) can be proved using Assumptions 6.4 (iii) (p. 228) and 3.11 (p. 113) as in [55,57] [9, Lemma 11.3.1, p. 461].

In order to prove the relation F.41, let us consider the system state-space model 2.8 (p. 22). Since the controlled system is asymptotically stable (Assumption 6.4 (iii)), it follows from [9, Lemma B.3.3, p. 486] using superposition that there exist finite positive constants K_{51}, K_{61}, and K_{62} such that

$$\sum_{t=1}^{N} \|y_F(t)\|^2 \leq K_{51}\sum_{t=0}^{N} u(t)^2 + K_{61}\sum_{t=0}^{N} \omega(t)^2 + K_{62},$$

for $N > 0$. Furthermore, since the noise $\omega(t)$ is mean-square bounded (Assumption 3.11, p. 113), there exist constants K_6 and N_3 such that (after division by N)

$$\frac{1}{N}\sum_{t=1}^{N} \|y_F(t)\|^2 \leq \frac{K_5}{N}\sum_{t=0}^{N} u(t)^2 + K_6 \quad \text{for} \quad N > N_3 \quad a.s.,$$

which proves the relation F.41.

Next to prove the relation F.42, let us consider eqn. F.38

$$\overline{\chi}(N-k) = \sum_{t=k}^{N} \phi(t-k)'\phi(t-k) + \overline{\chi}(-1).$$

Using definition of the regression vector (see eqn. 6.55, p. 229) one has

$$\overline{\chi}(N-k) = K_{71}\sum_{t=0}^{N-k} u(t)^2 + K_{81}\sum_{t=1}^{N-k} \|y_F(t)\|^2 + K_{91},$$

where K_{71}, K_{81}, and K_{91} are some finite positive constants. Dividing by N and using eqn. F.41 one has

$$\frac{\overline{\chi}(N-k)}{N} \leq \frac{K_{71}}{N} \sum_{t=0}^{N-k} u(t)^2 + \frac{K_{82}}{N} \sum_{t=0}^{N-k} u(t)^2 + K_{92},$$

and from eqns. F.39 and F.40

$$\frac{\overline{\chi}(N-k)}{N} \leq \frac{K_{72}}{N} \sum_{t=k}^{N} y(t)^2 + K_{93} \leq \frac{K_7}{N} \sum_{t=k}^{N} [e_r(t) - \epsilon(t)]^2 + K_8 \quad a.s.,$$

for $N > \overline{N} = \max(N_1, N_2, N_3)$.

Q.E.D.

Note that eqn. F.42 verifies the second assumption of the SKTL (cf. [9, eqn. 8.5.75, p. 334]).

Furthermore, eqns. F.4, F.5, 3.146 (p. 113), F.8, 6.58 (p. 229), and F.38, imply that the remaining assumptions of the SKTL are satisfied (cf. [9, eqns. 8.5.71 to 8.5.74, p. 334]. Then it follows from the SKTL [1] [9, Lemma 8.5.3, p. 334] that

$$\limsup_{N\to\infty} \frac{1}{N} \overline{\chi}(N-k) < \infty \quad a.s. \tag{F.43}$$

Thus eqns. 6.59 and 6.60 (p. 230) follow from the above equation and from eqns. F.38 and 6.55 (p. 229).

In order to show that eqn. 6.61 is satisfied, let us consider the state-space model 2.8 (p. 22). Since the controlled system is asymptotically stable (Assumption 6.4 (iii), p. 228) it follows from [9, Lemma B.3.3, p. 486] using superposition and Assumption 3.11 (p. 113) that there exist finite positive constants K_9, K_{10}, and N_4 such that

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} y(t)^2 \leq \lim_{N\to\infty} \frac{K_9}{N} \sum_{t=0}^{N} u(t)^2 + K_{10} \quad \text{for } N > N_4 \quad a.s.$$

Then eqn. 6.61 follows from the above equation using eqn. 6.59.

Finally, eqn. 6.62 follows from the SKTL [9, eqn. 8.5.78, p. 334] using eqns. 6.58 (p. 229) and 3.145 (p. 113).

Q.E.D.

[1] Alternatively, the relation given by eqn. F.43 can be established without referring to the SKTL using the Kronecker lemma [187, Lemma D.1.2, p. 236] as in [57].

Appendix G

A survey of self-tuning control of stochastic systems with respect to convergence analysis and methods of overcoming the strictly positive real condition.

For the purpose of assessing the significance of the properties of the ST-MV-TIMO strategy introduced in subsection 6.2.2, we shall discuss stochastic, discrete-time, self-tuning, minimum prediction error (MPE) controllers for two-input, single-output (TISO) systems [1]. These self-tuning controllers were developed for TISO systems described by the ARMAX model 2.26 (p. 29). The ARMAX model of the controllable and observable TISO system having the feedback configuration FI follows from eqn. 3.89 (p. 92) with $A(p) = d(p)$, $B(p) = N(p)$, and $C(p) = c_\omega(p)$.

[1] In this appendix the term "self-tuning controller" is used in the context of TISO systems unless otherwise stated.

Methods of convergence analysis of self-tuning controllers.

There are two methods (tools) of analysis of convergence properties of self-tuning controllers: (i) the averaging method, and (ii) the method based on a Martingale convergence theory.

The averaging method was used in [54] to determine possible convergence points for the parameter estimates of the self-tuning MV regulator. This method forms the basis of the so-called ODE (ordinary differential equation) approach developed in [52] for analysis of convergence properties of recursive stochastic algorithms. The ODE method was used in [53] to establish sufficient conditions for convergence of the minimum variance (MV) self tuning regulator employing the stochastic gradient (SG) estimator to the MV control law.

The drawback of the ODE method is that it assumes boundedness of the input-output data (i.e., stability of the system). This is not a severe restriction in the analysis of the open-loop estimation algorithms. However, in the analysis of self-tuning control schemes any *a priori* assumptions regarding closed-loop system stability are undesirable. The reason is that there might be periods of unstable behaviour of the adaptive system since the parameters of the control law are estimated or are based on the plant estimates. Certain combinations of parameter estimates, which are obtained during the normal operation of the self-tuning system or during undesirable behaviour of the estimator (such as "burst" of estimates due to the lack of persistent excitation), might lead to the closed-loop system instability. Hence, in general, data boundedness cannot be verified *a priori*.

The second method of analysis, which is based on a Martingale convergence theory, overcomes the above drawback of the ODE method. The martingale approach is used to establish the closed-loop system stability and convergence simultaneously (global convergence), i.e., no *a priori* assumption is required about boundedness of the control signal and system output sequences. Furthermore, the stability and convergence can be established under relatively weak assumptions; for example, no persistent excitation condition is needed. The martingale method was firstly employed in analysis of open-loop recursive parameter estimation algorithms [132,133], and secondly in analysis of self-tuning control strategies, the initial one being that of [55].

Sometimes, in order to strengthen the results, the ODE and martingale methods are combined in the analysis of self-tuning controllers [172,188,134].

The martingale method is employed in analysis of the ST-MV-TIMO strategy in subsection 6.2.2.

Types of convergence properties of self-tuning controllers.

The above methods of convergence analysis are usually employed to establish (i) parameter estimates convergence, and (ii) global convergence (see Comment 6.6, p. 231) of self-tuning controllers.

The analysis of the open-loop estimation algorithms using the ODE method [53,52] or martingale method [132,133] concentrates on the *parameter estimate convergence* (consistency of estimates) and asymptotic properties of the parameter estimate errors. As pointed out by many authors, convergence of parameter estimates is desirable in self-tuning control [56,69,188]. This is because the input signal is generated by feedback and, therefore, it is likely that there will be long periods during which the system is not sufficiently excited. The lack of persistently exciting input signal might lead to the divergence of estimates ("burst" phenomenon) and instability of the closed-loop system [6]. So far, in order to establish parameter estimates convergence it is necessary to *assume* [68,56] or *ensure* [189,188,134,190] some form of *persistency of excitation* of the system (see also [172,134] for a review of various persistency of excitation conditions).

Furthermore, it is claimed by some authors that the *exponential* rate of convergence (of the parameter estimates and/or of the controlled output sequence to the reference sequence) must be guaranteed for self-tuning controllers designed for time-invariant systems [191,14]. For example, according to [191]:

> It is in fact doubtful that a proof of convergence without exponential

> convergence of an adaptive algorithm should be regarded as providing justification in itself for use of the algorithm.

The reason for this is twofold [191,14,192]. Firstly, self-tuning controllers should cope with slowly time-varying systems. It is rather doubtful that a convergence rate slower than exponential for a time-invariant system can lead to satisfactory performance for a time-varying system. Secondly, the exponentially convergent self-tuning strategies are robust to modeling errors.

An alternative approach to that establishing parameter convergence was undertaken in [55,125,57] using martingale method to focus on the overall system performance (*global convergence*) resulting from the estimated parameters, rather than properties of the estimated parameters themselves. Such an approach to convergence analysis is justified in practice since in applications of self-tuning control the parameter estimates convergence, although important, is not necessarily the most essential aspect of the design. Furthermore, it is well known that self-tuning controllers often yield satisfactory performance after the initial phase of tuning, despite rather poor parameter estimates.

Global convergence can be established *without* any persistent excitation condition. The convergence of the parameter estimates difference can be readily established as well (see Comment 6.7, p. 231).

Global convergence is established for the ST-MV-TIMO strategy in subsection 6.2.2.

Although our discussion is limited to time-invariant systems, it is perhaps worth mentioning the convergence properties established for time-varying systems. In [15,17] global convergence was established for the MPE control law combined with the SG estimator for a system with converging martingale parameters. The martingale method of analysis was employed in [18] to show that the minimum variance/stochastic gradient (MV/SG) self-tuning strategy is globally convergent for systems described by a state-space model corresponding to a nonsteady-state innovations representation. The latter result is of particular interest, since it illustrates the robustness of the MV/SG self tuner, which was originally proposed for time-invariant systems in [55], when applied to a certain class of time-varying systems. This poses an counterexample to the remark cited from [191] on page 319, relating exponential convergence to robustness of self tuners developed for time-invariant systems but applied to time-varying systems.

Development of globally convergent self-tuning controllers employing matrix gain estimators.

Global convergence of a self-tuning controller was firstly established for the MV/SG strategy (using martingale method) in the breakthrough article [55]. Initially two special cases were considered: (1) delay $k = 1$ and a general form of the noise polynomial $c_\omega(p)$, and (2) arbitrary delay $k \geq 1$ and $c_\omega(p) = 1$. Later this approach was extended to cover the arbitrary delay and general noise polynomial case [125]. It was pointed out in [56,69], however, that the performance of the self tuner [55] was inferior to that of [22] provided the latter scheme converges. The inferior performance was due to the use of the SG estimator in [55], the convergence rate of which was slower (e.g., 100 times) than that of the RLS algorithm employed in [22]. Therefore, in order to *improve the convergence rate* of the (globally convergent) self-tuning controller, a least-squares type estimator (i.e., a matrix gain estimator) could be employed.

However, certain difficulties were encountered in the extension of convergence results derived for the self tuner employing the SG estimator to the case of the least-squares type algorithm. Of primary interest is the problem of self-tuning closed-loop system instability resulting from unsatisfactory behaviour of the estimator (divergence of estimates) in the absence of persistent excitation of the controlled system.

The problem of the parameter estimates divergence and self-tuning closed-loop system instability is sometimes addressed by the following heuristic argument [54]. If parameter estimates are accurate and the control law ensures stability of the closed-loop system while being not sensitive to parameter variations, then a stable closed-loop system results. On the other hand, if the estimates are poor, then an unstable closed-loop system may result, and the input and output signals increase without limit. The stochastic disturbance becomes insignificant and the estimates of deterministic part of the system (i.e., of polynomials $A(p)$ and $B(p)$ in the ARMAX model 2.26, p. 29) converge to their true values. Then the closed-loop system becomes stable again, and a period of satisfactory performance follows.

As pointed out in [54], a potential problem associated with such an argument is revealed in the situation when not all of the system modes become unstable. Then only information about the unstable modes increases. As a result, only a certain combination of parameter estimates converges to its true value. This may not be sufficient to stabilize the closed-loop system or to change the unstable modes enough to reveal other modes of the system. Therefore, it was suggested in [193] that

> ... the (self-tuning) algorithm itself must be modified to ensure stability and convergence.

Such modifications proposed for self-tuning controllers can be classified as follows.

Estimator modification (method E). The self-tuning controllers based on method E rely on modification of the RELS estimator.

In the scheme of [56,69] a reduced weighting of the RELS performance index is introduced if insufficient persistence of excitation is detected over a finite period of time. The weightings are selected according to a scalar measure of system stability. The resulting estimator is called the weighted least squares (WLS) algorithm.

In the strategy of [57] a full RELS step is taken only if this is consistent with the convergence criterion; otherwise the estimator has the maximum possible gain for which boundedness of the condition number of the covariance matrix is guaranteed. The estimation algorithm introduced in [57] is called here the RELS with condition number monitoring (RELS-CNM). Its usual name is, however, the modified LS algorithm [57,56,69,180].

Dither injection (method D). Method D provides a solution to overcome the lack of persistent excitation in the closed-loop system by injecting a dither signal

[15]. In order to enhance the convergence rate of the SG based scheme of [55], a two-estimator strategy was proposed in [189]. In this strategy white noise dither is added to the control signal produced by the SG based algorithm of [55], and the approximate maximum likelihood (AML) estimator [2] is used to obtain consistent parameter estimates for the plant. The performance of this scheme is suboptimal due to dither added to the control signal to produce the so-called *continually disturbed control* (CDC). This means that the convergence result similar to that for the ST-MV-TIMO scheme (see eqn. 6.62, p. 231) can be established with asymptotic tracking error variance $\sigma^2 + \sigma_d^2$, where σ_d^2 is the dither variance (unit delay system was considered in [189], therefore $F = 1$).

Another example of the application of method D is the strategy of [180] involving the AML estimator. The bounded white noise "probing inputs" are introduced into the closed-loop system whenever inadequacy of information content for estimation is detected by a criterion based on the covariance estimate error matrix. Although it is claimed by the authors that in terms of convergence rate the latter scheme is superior to that of [57] involving the RELS-CNM, the performance of the closed-loop system is suboptimal due to dither signal.

Estimator modification and dither injection (method E-D). The above two methods E and D are sometimes combined into method E-D. The self-tuning strategies based on the CDC approach were introduced in [172,134]. These schemes involve the RELS-CNM estimator and dither added to the reference sequence to ensure parameter consistency. Various two-estimator, self-tuning strategies have been considered in [58,188]. For example, the self-tuning controller of [188] involves a mechanism switching between the minimum variance control law combined with the weighted RELS estimator of [69], and a "fall-back" controller combined with the weighted RLS algorithm. Dither is added to the control signal after switching to the "fall-back" controller to ensure pa-

[2] The AML algorithm is a version of the RELS algorithm with *a posteriori* prediction errors used in the regression vector [133] instead of *a priori* prediction errors [174].

rameter convergence.

Methods E, D, and E-D, represent the state of the art in the development of convergent self-tuning controllers involving the matrix gain estimators (see also Comment 6.11, p. 237).

Methods of overcoming the strictly positive real condition involving the noise polynomial $c_\omega(z^{-1})$.

Analysis of the SG and RELS estimators for open-loop system identification, presented in [181], revealed difficulties in convergence of parameter estimates for certain systems. This led to formulation of sufficient conditions for the convergence of parameter estimates (with probability one) to true parameter values of systems described by the ARMAX model in [53]. In particular, it was shown that the RELS estimator yields consistent parameter estimates if the transfer function $c_\omega^{-1}(z^{-1}) - 1/2$ is *strictly positive real* [179].

The *strictly positive real* (SPR) condition is assumed in the convergence analysis [3] of recursive stochastic algorithms for open-loop parameter estimation [53,133,68,191] [172,134,9,190,180] as well as for self-tuning controllers developed for TISO systems [53,56,189,57,172,188,134,9,190,18,180]. Furthermore, it appears in the convergence analysis using the ODE or martingale methods, or for the combination of both. The SPR condition is assumed for both convergence of parameter estimates and global convergence of self tuners. Therefore [126]

> The need for this assumption [4] is unfortunate, but represents the current state of art in the analysis of recursive estimation algorithms.

In fact it was suggested in [56] that the SPR condition is close to being a necessary one for the tracking error convergence.

The question arises whether the SPR condition is restrictive in practical situations. The best answer is provided by the example in [53]. For a second order noise

[3] Heuristic explanation of the SPR condition is given in [53,194] [9, Remark 8.4.1, p. 319].

[4] i.e., the SPR condition

polynomial $c_\omega(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2}$ the region which is admissible for coefficients c_1 and c_2 in view of the SFT is the triangle depicted in fig. G.1. The shaded area is the region for which the SPR condition for the RELS estimator is satisfied. This means that for 50% of the combinations of values of coefficients c_1 and c_2 the convergence of the self-tuning controllers employing the RELS estimator (or its variants) is not guaranteed.

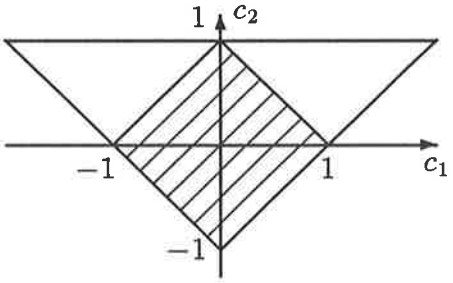

Figure G.1: The triangle represents the admissible region for coefficients c_1 and c_2 of a second order noise polynomial $c_\omega(z^{-1})$; the shaded area is the region where the SPR condition is satisfied.

Therefore, it is desirable to develop methods of overcoming the SPR condition.

The first method is based on the modification of the RELS estimator proposed in [181,53] which leads to the RML algorithm. The RML estimator avoids the SPR condition at the expense of filtering the input-output data by a time-varying filter $D(t,p) = \widehat{c}_\omega(t,p)$, where $\widehat{c}_\omega(t,p)$ is the estimate of the noise polynomial $c_\omega(p)$ (see eqn. 6.63, p. 232) [53,177]. A potential problem associated with the RML estimator is that the noise filter estimate $\widehat{c}_\omega(t,p)$ may become unstable. Therefore $\widehat{c}_\omega(t,p)$ must be kept within the stability region. This requires a stability test to be performed at every sample instant t. If $\widehat{c}_\omega(t,p)$ is unstable than some projection facility must be invoked to map the estimate of the noise polynomial into the stability region. Various methods for projection of the estimate $\widehat{c}_\omega(t,p)$ into the stability region were proposed for open-loop estimation [195,196].

The RML estimator was suggested for self-tuning controllers in many survey papers devoted to adaptive control [170,10,5] [160, p. 414]. The performance of the

RML-based self tuners was assessed by simulation studies [197] (see also subsection 6.2.3). However, to the author's knowledge, no convergence result is available for such strategies. One of the reasons is that there is no general method of estimate projection into the stability domain which would guarantee the closed-loop system stability [5].

Let us restrict our attention to those strategies which overcome the SPR condition and are shown to be convergent. Then the following methods of overcoming the SPR condition can be identified [6].

Dither injection (method D). The first approach, proposed in [48], is based on the direct on-line minimization of the performance criterion using the stochastic gradient algorithm. To avoid the SPR condition, the coloured noise is ignored and only the deterministic part of the system is estimated using the recursive instrumental variables method [177]. The convergence of the scheme to a local minimum of the cost function with probability one is subject to convergence of the estimates of the deterministic part of the system to true values. In order to ensure the latter condition, white noise dither is added to the control signal leading to suboptimal asymptotic performance.

The second approach, proposed in [58], relies on adding zero mean white noise dither having variance σ_d^2 to the output of the controlled system. Let us define the noise polynomial $\overline{c}_\omega(z^{-1})$ as a spectral factor of $c_\omega(z^{-1})c_\omega(z) + \sigma_d^2$. Then $\overline{c}_\omega(z^{-1})$ describes the noise polynomial of the original system augmented with dither. Furthermore, it is shown in [58] that $\overline{c}_\omega(z^{-1})$ satisfies the SPR condition provided σ_d^2 is sufficiently large. The self-tuning controller employing the WLS estimator [56] is globally convergent when applied to the system with added dither (the reference sequence $r(t)$ is required to be persistently exciting). The

[5] The first step towards development of a globally convergent self tuner based on the RML estimator is, perhaps, a sophisticated strategy of [58] involving the RML algorithm as one of its two estimators and obviating necessity of the projection of the noise polynomial estimate into the stability region (see method D).

[6] All references known to the author are quoted in this classification.

closed-loop system performance is suboptimal due to dither.

Subsequently, the above strategy was extended to recover asymptotic optimality. Namely, the estimate of the original noise polynomial $c_\omega(z^{-1})$ (assumed to have roots strictly inside the unit circle) can be calculated from the estimate of $\overline{c}_\omega(z^{-1})$ and estimate of the variance σ^2 of the stochastic disturbance $\omega(t)$. With the converging estimates of $\overline{c}_\omega(z^{-1})$, converging estimates of $c_\omega(z^{-1})$ can be obtained and used in the calculation of the control signal; then dither injection at the system output can be discontinued. The resulting technique involves two estimators, spectral factorization and switching between two controllers. The asymptotic optimality of the scheme is subject to the convergence of estimates of $\overline{c}_\omega(z^{-1})$, i.e., it requires persistency of excitation of $r(t)$; otherwise it may be necessary to continue to apply some dither of arbitrarily small variance.

Further improvement in the convergence rate of the overall system is possible if the prewhiting filter $D(t,p)$ for the input-output data is introduced for the estimator. The filter is chosen to be the estimate of the original noise polynomial $c_\omega(p)$. This implies that the RML estimator is asymptotically employed. [7]

Model overparameterization (method O). A method of overcoming the SPR condition by overparameterization was proposed in [194]. It has been shown that for a noise polynomial $c_\omega(z^{-1})$ of order n and not necessarily satisfying the SPR condition, there exists a polynomial $\tilde{c}_\omega(z^{-1})$ of order $\tilde{n}$ such that the polynomial $c_\omega(z^{-1})\tilde{c}_\omega(z^{-1})$ satisfies the SPR condition. The upper bound on order $\tilde{n}$ is determined both by the degree of $c_\omega(z^{-1})$ and the maximum modulus of roots of $c_\omega(z^{-1})$ ($\tilde{n}$ may be large if roots of $c_\omega(z^{-1})$ lie close to the unit circle). Therefore, global convergence of a self tuner employing the SG or RELS estimator can be established for the overparameterized system model, i.e., the SPR condition is overcome at the cost of ill-conditioning of the

[7] The strategy involving switching between the WLS and RML estimators obviates the necessity of the stability test and projection of the noise polynomial estimate into the stability region required for the RML algorithm [58]. As already mentioned, such a projection method has not been developed yet for closed-loop applications.

estimator and slower convergence rate due to the increase in the number of estimated parameters. Some prior knowledge of the noise polynomial $c_\omega(z^{-1})$ is required to determine $\tilde{n}$.

Model overparameterization and dither injection (method O-D). This approach involves a special case of the multistep multivariable adaptive regulator (MUSMAR) [130] based on minimization of a *multi-stage* performance criterion rather that a single-stage cost function. The multi-step-ahead predictor employed in the development of the MUSMAR strategy represents a nonminimal parameterization of the plant [135]. It was shown using the ODE method that for a simplified version of the MUSMAR regulator combined with the RLS estimator, the MV control law is a locally stable stationary point of convergence of the self tuner, even if the noise polynomial does not satisfy the SPR condition [135]. Appropriate choice of the prediction horizon is determined both by the degree of the noise polynomial $c_\omega(z^{-1})$ and the maximum modulus of roots of $c_\omega(z^{-1})$ (which are restricted to lie strictly inside the unit circle). Zero mean white noise dither is added to the control signal in order to ensure identifiability of parameters. The drawback of this approach is that it is restricted to the regulator case ($r(t) = 0$) and requires some knowledge of the noise polynomial in order to determine parameters of the control law.

Methods D, O, and O-D, represent the state of the art in the development of convergent self-tuning controllers which overcome the SPR condition for TISO systems (see also Comment 6.12, p. 238).

Bibliography

[1] F. Jay, editor. *IEEE Standard Dictionary of Electrical and Electronics Terms.* IEEE, Inc., New York, USA, third edition, 1984.

[2] B. Wittenmark. Stochastic adaptive control methods: A survey. *Int. J. Control*, 21(5):705–730, May 1975.

[3] O. L. R. Jacobs. Introduction to adaptive control. In C. J. Harris and S. A. Billings, editors, *Self-tuning and Adaptive Control: Theory and Applications*, pages 1–35, Peter Peregrinus Ltd on behalf of the IEE, England, 1985.

[4] M. M. Gupta, editor. *Adaptive Methods for Control System Design.* IEEE Press, New York, 1986.

[5] K. J. Åström. Adaptive feedback control. *Proc. IEEE*, 75(2):185–217, February 1987.

[6] K. J. Åström. Theory and applications of adaptive control - A survey. *Automatica*, 19(5):471–486, September 1983.

[7] J. W. Patchell and O. L. R. Jacobs. Separability, neutrality and certainty equivalence. *Int. J. Control*, 13(2):337–342, February 1971.

[8] Y. Bar-Shalom and E. Tse. Dual effect, certainty equivalence, and separation in stochastic control. *IEEE Trans. Automat. Control*, AC-19(5):494–500, October 1974.

[9] G. C. Goodwin and K. S. Sin. *Adaptive Filtering Prediction and Control.* Prentice-Hall, Englewood Cliffs, N.J. 07632, 1984.

[10] R. Isermann. Parameter adaptive control algorithms - A tutorial. *Automatica*, 18(5):513–528, September 1982.

[11] T. R. Fortescue, L. S. Kershenbaum, and B. E. Ydstie. Implementation of self-tuning regulators with variable forgetting factors. *Automatica*, 17(6):831–835, November 1981.

[12] G. C. Goodwin, H. Elliott, and E. K. Teoh. Deterministic convergence of a self-tuning regulator with covariance resetting. *IEE Proc.*, 130, Pt. D(1):6–8, January 1983.

[13] R. Kulhavý and M. Kárný. Tracking of slowly varying parameters by directional forgetting. In *IFAC 9-th Triennial World Congress*, pages 687–692, Budapest, 1984.

[14] B. D. O. Anderson and C. R. Johnson. Exponential convergence of adaptive identification and control algorithms. *Automatica*, 18(1):1–13, January 1982.

[15] P. E. Caines. Stochastic adaptive control: Randomly varying parameters and continually disturbed controls. In *IFAC 8-th Triennial World Congress*, pages 925–930, Kyoto, Japan, 1981.

[16] J. Holst and N. K. Poulsen. Self tuning control of plants with abrupt changes. In *IFAC 9-th Triennial World Congress*, pages 923–928, Budapest, 1984.

[17] H. F. Chen and P. E. Caines. On the adaptive control of stochastic systems with random parameters. In *23-rd IEEE Confr. Decision and Control*, pages 33–38, Las Vegas, NV, December 1984.

[18] G. C. Goodwin, D. J. Hill, and X. Xianya. Stochastic adaptive control for exponentially convergent time-varying systems. *SIAM J. Control and Optimization*, 24(4):589–603, July 1986.

[19] R. H. Middleton, G. C. Goodwin, D. J. Hill, and D. Q. Mayne. *Design issues in adaptive control.* Technical Report EE8544, University of Newcastle, Dept. of

Electrical and Computer Engineering, NSW 2308, Australia, December 1985. Revised April 1987.

[20] C. J. Harris and S. A. Billings, editors. *Self-tuning and Adaptive Control: Theory and Applications.* Peter Peregrinus Ltd on behalf of the IEE, England, second edition, 1985.

[21] R. E. Kalman. Design of a self-optimizing control system. *Trans. ASME*, 80:468–478, February 1958.

[22] K. J. Åström and B. Wittenmark. On self-tuning regulators. *Automatica*, 9(2):185–199, March 1973.

[23] D. W. Clarke and P. J. Gawthrop. Self-tuning controller. *Proc. IEE*, 122(9):929–934, September 1975.

[24] D. W. Clarke and P. J. Gawthrop. Self-tuning control. *Proc. IEE*, 126(6):633–640, June 1979.

[25] P. J. Gawthrop. Some interpretations of the self-tuning controller. *Proc. IEE*, 124(10):889–894, October 1977.

[26] D. W. Clarke, P. P. Kanjilal, and C. Mohtadi. A generalised LQG approach to self-tuning control. *Int. J. Control*, 41(6):1509–1544, June 1985.

[27] V. Peterka. Predictor-based self-tuning control. *Automatica*, 20(1):39–50, January 1984.

[28] B. E. Ydstie. Extended horizon adaptive control. In *IFAC 9-th Triennial World Congress*, pages 911–915, Budapest, 1984.

[29] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. *Generalized predictive control.* OUEL Report 1555/84 and 1557/84, Oxford University, Dept. of Engineering Science, Park Road, Oxford OX1 3PJ, UK, 1984.

[30] C. R. Cutler and B. L. Ramaker. Dynamic matrix control - A computer control algorithm. In *Proc. Joint Automatic Control Conference*, San Francisco, Ca, August 1980. Paper WP5-B.

[31] U. Borisson. Self-tuning regulators for a class of multivariable systems. *Automatica*, 15(2):209–215, March 1979.

[32] G. C. Goodwin, P. J. Ramadge, and P. E. Caines. Discrete-time multivariable adaptive control. *IEEE Trans. Automat. Control*, AC-25(3):449–456, June 1980.

[33] W. A. Wolovich and P. L. Falb. Invariants and canonical forms under dynamic compensation. *SIAM J. Control Optim.*, 14(6):996–1008, November 1976.

[34] G. C. Goodwin and R. S. Long. Generalization of results on multivariable adaptive control. *IEEE Trans. Automat. Control*, AC-25(6):1241–1245, December 1980.

[35] C. A. Tsiligiannis and S. A. Sovronos. Multivariable self-tuning control via the right interactor matrix. *IEEE Trans. Automat. Control*, AC-31(10):987–989, October 1986.

[36] L. Dugard, G. C. Goodwin, and X. Xiana. The role of the interactor matrix in multivariable stochastic adaptive control. *Automatica*, 20(5):701–709, September 1984.

[37] L. Dugard, G. C. Goodwin, and C. de Souza. Prior knowledge in model reference adaptive control of multiinput multioutput systems. *IEEE Trans. Automat. Control*, AC-29(8):761–765, August 1984.

[38] B. E. Ydstie and L. K. Liu. Single- and multi-variable control with extended prediction horizons. In *Proc. of 1984 American Control Confr.*, pages 1303–1308, San Diego, Ca, June 1984.

[39] H. Elliott and W. A. Wolovich. Parameterization issues in multivariable adaptive control. *Automatica*, 20(5):533–545, September 1984.

[40] P. E. Wellstead, J. M. Edumunds, D. Prager, and P. Zanker. Self-tuning pole/zero assignment regulators. *Int. J. Control*, 30(1):1–26, July 1979.

[41] P. E. Wellstead and P. Zanker. Servo self-tuners. *Int. J. Control*, 30(1):27–36, July 1979.

[42] A. Y. Allidina and F. M. Hughes. Generalised self-tuning controller with pole assignment. *IEE Proc.*, 127, Pt. D(1):13–18, January 1980.

[43] H. Elliott, W. A. Wolovich, and M. Das. Arbitrary adaptive pole placement for linear multivariable systems. *IEEE Trans. Automat. Control*, AC-29(3):221–228, March 1984.

[44] K. J. Åström and B. Wittenmark. Self-tuning controllers based on pole-zero placement. *IEE Proc.*, 127, Pt. D(3):120–130, May 1980.

[45] C. R. Johnson, D. A. Lawrence, and J. P. Lyons. A flaw in the reduced-order behaviour of a direct adaptive pole placer. *IEEE Trans. Automat. Control*, AC-28(9):922–924, September 1983.

[46] J. B. Moore. A globally convergent recursive adaptive LQG regulator. In *IFAC 9-th Triennial World Congress*, pages 955–959, Budapest, 1984.

[47] J-C. Savelli, K. Warwick, and J. H. Westcott. Implementation of an adaptive PID self-tuning controller. In *Proc. Melecon 83*, Athens, Greece, 1983. Paper C10.01.

[48] L. Ljung and E. Trulsson. Adaptive control based on the explicit criterion minimization. In *IFAC 8-th Triennial World Congress*, pages 881–886, Kyoto, Japan, 1981.

[49] M. A. Lelić and M. B. Zarrop. Generalized pole-placement self-tuning controller. Part 1. Basic algorithm. *Int. J. Control*, 46(2):547–568, August 1987.

[50] H. Elliott, T. Depkovich, J. Kelly, and B. Draper. Nonlinear adaptive control of mechanical linkage systems with applications to robotics. In *Proc. of 1983 American Control Confr.*, pages 1050–1055, San Francisco, Ca, June 1983.

[51] L. Ljung and B. Wittenmark. On a stabilizing property of adaptive regulators. In *Proc. of the 4th IFAC Symposium: Identification and System Parameter Estimation*, Tbilisi, USSR, September 1976.

[52] L. Ljung. Analysis of recursive stochastic algorithms. *IEEE Trans. Automat. Control*, AC-22(4):551–575, August 1977.

[53] L. Ljung. On positive real transfer functions and the convergence of some recursive schemes. *IEEE Trans. Automat. Control*, AC-22(4):539–551, August 1977.

[54] K. J. Åström, U. Borisson, L. Ljung, and B. Wittenmark. Theory and applications of self-tuning regulators. *Automatica*, 13(5):457–476, September 1977.

[55] G. C. Goodwin, P. R. Ramadge, and P. E. Caines. Discrete time stochastic adaptive control. *SIAM J. Control and Optimization*, 19(6):829–853, November 1981.

[56] R. Kumar and J. B. Moore. Convergence of adaptive minimum variance algorithms via weighting coefficient selection. *IEEE Trans. Automat. Control*, AC-27(1):146–153, February 1982.

[57] K. S. Sin and G. C. Goodwin. Stochastic adaptive control using a modified least squares algorithm. *Automatica*, 18(3):315–321, May 1982.

[58] J. B. Moore. Sidestepping the positive real restriction for stochastic adaptive schemes. *Ricerche di Automatica*, 13(1):56–84, October 1982.

[59] K. J. Åström and B. Wittenmark. *Computer Controlled Systems Theory and Design.* Prentice-Hall, Englewood Cliffs, N.J. 07632, 1984.

[60] W. M. Wonham. *Linear Multivariable Control: a Geometric Approach.* Springer-Verlag, 1979.

[61] L. Pernebo. An algebraic theory for the design of controllers for linear multivariable systems-part 1: Structure matrices and feedforward design. *IEEE Trans. Automat. Control*, AC-26(1):171–182, February 1981.

[62] J. C. Willems and C. Commault. Disturbance decoupling by measurement feedback with stability or pole placement. *SIAM J. Control Optim.*, 19(4):490–504, July 1981.

[63] A. C. Antoulas. A new approach to synthesis problems in linear system theory. *IEEE Trans. Automat. Control*, AC-30(5):465–473, May 1985.

[64] A. C. Antoulas. On the solvability of the general feedback control problem. *Int. J. Control*, 44(5):1273–1284, November 1986.

[65] V. Kučera. *Discrete Linear Control.* John Wiley & Sons, New York, 1979.

[66] H. N. Koivo. A multivariable self-tuning controller. *Automatica*, 16(4):351–366, July 1980.

[67] L. Dugard and J. M. Dion. Direct adaptive control for linear multivariable systems. *Int. J. Control*, 42(6):1251–1281, December 1985.

[68] R. Kumar. Almost sure convergence of adaptive identification prediction and control algorithms. In *20-th IEEE Confr. Decision and Control*, pages 1241–1246, San Diego, Ca, December 1981.

[69] R. Kumar. Simultaneous adaptive control and identification via the weighted least-squares algorithm. *IEEE Trans. Automat. Control*, AC-29(3):259–263, March 1984.

[70] B. D. O. Anderson and J. B. Moore. *Optimal Filtering.* Prentice-Hall, Englewood Cliffs, N.J. 07632, 1979.

[71] Y. T. Tsay and L. S. Shieh. State-space approach for self-tuning control with pole assignment. *Proc. IEE*, 128, Pt.D(3):93–101, May 1981.

[72] K. Warwick. Self-tuning regulators – A state-space approach. *Int. J. Control*, 33(5):839–858, May 1981.

[73] K. Warwick. Self-tuning property of state-space self-tuners. *Proc. IEE*, 129, Pt.D(3):96–100, May 1982.

[74] K. P. Lam. Design of stochastic discrete time linear optimal regulators. *Int. J. Systems Sci.*, 13(9):979–1011, September 1982.

[75] W. A. Wolovich. *Linear Multivariable Systems.* Springer-Verlag, New York, N.Y., 1974.

[76] F. M. Callier and C. A. Desoer. *Multivariable Feedback Systems.* Springer-Verlag, New York, 1982.

[77] W. A. Wolovich and H. Elliott. Discrete models for linear multivariable systems. *Int. J. Control*, 38(2):337–357, August 1983.

[78] A. Nehorai and M. Morf. Recursive identification algorithms for right matrix fraction description models. *IEEE Trans. Automat. Control*, AC-29(12):1103–1106, December 1984.

[79] M. J. Gibbard and A. P. Papliński. *A New Class of Computational Methods and Associated Methodology for the Design of Digital Controllers for Multi-input, Multi-output Linear Systems.* Technical Report CONT 86/01, University of Adelaide, Dept. of Electrical and Electronic Eng., G.P.O. Box 498, Adelaide, S.A. 5001, Australia, January 1986.

[80] T. Kailath. *Linear Systems.* Prentice-Hall, Englewood Cliffs, N.J. 07632, 1980.

[81] A. P. Papliński. *Transformation of a polynomial matrix to a monic or comonic form.* Technical Report CONT 87/01, University of Adelaide, Dept. of Electri-

cal and Electronic Eng., G.P.O. Box 498, Adelaide, S.A. 5001, Australia, April 1987.

[82] Yu. A. Rozanov. *Stationary Random Processes.* Holden-Day, San Francisco, Ca., 1967.

[83] K. J. Åström. *Introduction to Stochastic Control Theory.* Academic Press, London, 1970.

[84] D. W. Clarke, A. J. F. Hodgson, and P. S. Tuffs. Offset problem and k-incremental predictors in self-tuning control. *IEE Proceedings*, 130, Pt. D(5):217–225, September 1983.

[85] C. S. Berger. Self-tuning control of offset using a moving average filter. *IEE Proc.*, 133, Pt. D(4):184–188, July 1986.

[86] A. M. Yaglom. *Stationary Random Functions.* Prentice-Hall, Englewood Cliffs, N.J. 07632, 1962.

[87] J. L. Doob. *Stochastic Processes.* John Wiley and Sons, New York, 1967.

[88] H. Kawashima. Parameter estimation of autoregressive integrated processes by least squares. *Ann. Statist.*, 8(2):423–432, 1980.

[89] T. J. Harris, J. F. MacGregor, and J. D. Wright. Self-tuning and adaptive controllers: An application to catalytic reactor control. *Technometrics*, 22(2):153–164, May 1980.

[90] P. R. Belanger. On type 1 systems and the Clarke-Gawthrop regulator. *Automatica*, 19(1):91–94, January 1983.

[91] P. S. Tuffs and D. W. Clarke. *Self-tuning control of offset: A unified approach.* OUEL Report 1539/84, Oxford University, Dept. of Engineering Science, Park Road, Oxford OX1 3PJ, UK, 1984.

[92] R. Scattolini and D. W. Clarke. Multivariable model-following self-tuning control with offset elimination. *Int. J. Control*, 42(6):1309–1322, December 1985.

[93] S. C. Shah and G. F. Franklin. Internal model adaptive control. In *20-th IEEE Confr. Decision and Control*, pages 1257–1265, San Diego, Ca, December 1981.

[94] M. A. Lelić and P. E. Wellstead. Generalized pole-placement self-tuning controller. Part 2. Application to robot manipulator control. *Int. J. Control*, 46(2):569–601, August 1987.

[95] H. Elliott and W. A. Wolovich. A parameter adaptive control structure for linear multivariable systems. *IEEE Trans. Automat. Control*, AC-27(2):340–352, April 1982.

[96] A. S. Morse. Parameterizations for multivariable adaptive control. In *20-th IEEE Confr. Decision and Control*, pages 970–972, San Diego, Ca, December 1981.

[97] J. M. Dion and L. Dugard. Direct adaptive control of discrete time multivariable systems. In *21-st IEEE Confr. Decision and Control*, pages 780–781, Orlando, Florida, December 1982.

[98] S. W. Chan and G. C. Goodwin. On the role of the interactor matrix in multiinput-multioutput adaptive control. *IEEE Trans. Automat. Control*, AC-27(3):713–714, June 1982.

[99] K. Y. Wong and M. M. Bayoumi. Multivariable self-tuning regulators. In *20-th IEEE Confr. Decision and Control*, pages 978–983, San Diego, Ca, December 1981.

[100] K. S. Lee and Won-Kyoo Lee. Extended discrete-time multivariable adaptive control using long-term predictor. *Int. J. Control*, 38(3):495–514, September 1983.

[101] C. A. Tsiligiannis and S. A. Sovronos. Multivariable self-tuning controllers for systems with measurements of differing time delays. In *Proc. of 1984 American Control Confr.*, pages 899–904, San Diego, Ca, June 1984.

[102] G. C. Goodwin, B. C. McInnis, and J. C. Wang. Model reference adaptive control for systems having non-square transfer functions. In *21-st IEEE Confr. Decision and Control*, pages 744–749, Orlando, Florida, December 1982.

[103] R. Johansson. Parametric models of linear multivariable systems for adaptive control. In *21-st IEEE Confr. Decision and Control*, pages 989–993, Orlando, Florida, December 1982.

[104] M. Das. Multivariable adaptive model matching using less a priori information. *J. of Dynamic Systems, Measurement, and Control, Trans. ASME*, 108(2):151–153, June 1986.

[105] G. Favier and M. Hassani. Multivariable self-tuning controllers based on generalized minimum variance strategy. In *21-st IEEE Confr. Decision and Control*, pages 770–777, Orlando, Florida, December 1982.

[106] R. P. Singh and K. S. Narendra. Prior information in the design of multivariable adaptive controllers. *IEEE Trans. Automat. Control*, AC-29(12):1108–1111, December 1984.

[107] J. Feinstein and Y. Bar-Ness. The solution of the matrix polynomial equation. *IEEE Trans. Automat. Control*, AC-29(1):75–77, January 1984.

[108] A. P. Papliński. Matrix multiplication and division of polynomials. *IEE Proc.*, 132, Pt.D(3):95–99, May 1985.

[109] M. W. Rogoziński, A. P. Papliński, and M. J. Gibbard. An algorithm for the calculation of a nilpotent interactor matrix for linear multivariable systems. *IEEE Trans. Automat. Control*, AC-32(3):234–237, March 1987.

[110] C. Moler. *MATLAB Users' Guide*. May 1981.

[111] G. W. Stewart. *Introduction to Matrix Computations*. Academic Press, New York, N.Y., 1973.

[112] Chi-Tsong Chen. *Linear System Theory and Design.* Holt, Rinehart and Winston, New York, 1984.

[113] *MATRIX$_X$ User's Guide.* Palo Alto, Ca, May 1986.

[114] V. C. Klema and A. J. Laub. The singular value decomposition: Its computation and some applications. *IEEE Trans. Automat. Control*, AC-25(2):164–176, April 1980.

[115] R. G. Brown. *Smoothing, Forecasting and Prediction of Discrete Time Series.* Prentice-Hall, Englewood Cliffs, N.J. 07632, 1963.

[116] G. E. P. Box and G. M. Jenkins. *Time Series Analysis Forecasting and Control.* Holden-Day, San Francisco, 1976.

[117] R. L. Kashyap and A. R. Rao. Real time recursive prediction of river flows. *Automatica*, 9(2):175–183, March 1973.

[118] R. M. C. de Keyser and A. R. van Cauwenberghe. A self-tuning multistep predictor application. *Automatica*, 17(1):167–174, January 1981.

[119] H. Akaike. Block Toeplitz matrix inversion. *SIAM J. Appl. Math.*, 24(2):234–241, March 1973.

[120] I. S. Iohvidov. *Hankel and Toeplitz Matrices and Forms. Algebraic Theory.* Birkhauser Verlag, 1980.

[121] D. Commenges and M. Monsion. Fast inversion of triangular Toeplitz matrices. *IEEE Trans. Automat. Control*, AC-29(3):250–251, March 1984.

[122] A. Ben-Israel and T. N. E. Greville. *Generalized Inverses: Theory and Applications.* John Wiley and Sons, New York, 1974.

[123] B. Wittenmark. A self-tuning predictor. *IEEE Trans. Automat. Control*, AC-19(6):848–851, December 1974.

[124] J. T. Tanttu. A self-tuning predictor for a class of multivariable stochastic processes. *Int. J. Control*, 32(2):359–370, August 1980.

[125] G. C. Goodwin, K. S. Sin, and K. K. Saluja. Stochastic adaptive control and prediction – The general delay-colored noise case. *IEEE Trans. Automat. Control*, AC-25(5):946–950, October 1980.

[126] K. S. Sin, G. C. Goodwin, and R. R. Bitmead. An adaptive d-step ahead predictor based on least squares. *IEEE Trans. Automat. Control*, AC-25(6):1161–1165, December 1980.

[127] P. P. Kanjilal and E. Rose. Self-tuning multi-step prediction applied to speed control of a sinter strand. In *American Control Conference*, pages 863–868, Arlington, USA, June 1982.

[128] R. Isermann. Practical aspects of process identification. *Automatica*, 16(5):575–587, September 1980.

[129] R. Lamanna de R., R. A. Padilla, and M. Uria de C. On-line order selection and parameter estimation-an experimental application. *IEEE Control Systems Magazine*, 4(2):6–13, May 1984.

[130] C. Greco, G. Menga, E. Mosca, and G. Zappa. Performance improvements of self-tuning controllers by multistep horizons: The MUSMAR approach. *Automatica*, 20(5):681–699, September 1984.

[131] W. D. T. Davies. Generation and properties of maximum-length sequences. *Control*, 302–304, June 1966.

[132] J. Sternby. On consistency for the method of least-squares using martingale theory. *IEEE Trans. Automat. Control*, AC-22(3):346–352, June 1977.

[133] V. Solo. The convergence of AML. *IEEE Trans. Automat. Control*, AC-24(6):958–962, December 1979.

[134] H. F. Chen. Recursive system identification and adaptive control by use of the modified least squares algorithm. *SIAM J. Control and Optimization*, 22(5):758–776, September 1984.

[135] E. Mosca and G. Zappa. Removal of a positive realness condition in minimum variance adaptive regulators by multistep horizons. *IEEE Trans. Automat. Control*, AC-29(9):844–846, September 1984.

[136] E. J. Hannan. The identification problem for multiple equation systems with moving average errors. *Econometrica*, 39(5):751–765, September 1971.

[137] T. J. Moir and M. J. Grimble. Optimal self-tuning filtering, prediction, and smoothing for discrete multivariable processes. *IEEE Trans. Automat. Control*, AC-29(2):128–137, February 1984.

[138] U. Shaked and P. R. Kumar. Minimum variance control of discrete time multivariable ARMAX systems. *SIAM J. Control Optim.*, 24(3):396–411, May 1986.

[139] E. J. Hannan. *Multiple Time Series.* John Wiley and Sons, New York, 1970.

[140] J. L. Willems. *Stability Theory of Dynamical Systems.* Nelson, London, 1970.

[141] R. Paul. *Robot Manipulators: Mathematics, Programming and Control.* MIT Press, Cambridge, MA, 1981.

[142] A. J. Koivo and T. H. Guo. Control of robotic manipulator with adaptive control. In *20-th IEEE Confr. Decision and Control*, pages 271–276, San Diego, Ca, December 1981.

[143] C. S. G. Lee and M. J. Chung. An adaptive control strategy for computer-based manipulators. In *21-st IEEE Confr. Decision and Control*, pages 95–100, Orlando, Florida, December 1982.

[144] H. Seraji, M. Jamshidi, Y. T. Kim, and M. Shahinpoor. Linear multivariable control of two-link robots. *Journal of Robotic Systems*, 3(4):349–365, 1986.

[145] C. P. Neuman and V. D. Tourassis. Discrete dynamic robot models. *IEEE Trans. Systems, Man, and Cybernetics*, SMC-15(2):193–204, March/April 1985.

[146] A. O. Cordero and D. Q. Mayne. Deterministic convergence of a self-tuning regulator with variable forgetting factor. *IEE Proc.*, 128, Pt. D(1):19–23, January 1981.

[147] M. B. Zarrop. Variable forgetting factors in parameter estimation. *Automatica*, 19(3):295–298, May 1983.

[148] S. Saelid and B. Foss. Adaptive controllers with a vector variable forgetting factor. In *22-nd IEEE Confr. Decision and Control*, pages 1488–1494, San Antonio, Tx, December 1983.

[149] D. W. Clarke and L. Zhang. Long-range predictive control using weighting-sequence models. *IEE Proc.*, 134, Pt. D(3):187–195, May 1987.

[150] M. W. Rogoziński and M. J. Gibbard. *Long-range predictive control; self-tuning performance and offset elimination.* Report No CONT 85/04, University of Adelaide, Dept. of Electrical and Electronic Engineering, GPO Box 498, Adelaide, SA 5001, Australia, 1985.

[151] D. M. Prett and R. D. Gillette. Optimization and constrained multivariable control of a catalytic cracking unit. In *Proc. Joint Automatic Control Conference*, San Francisco, Ca, August 1980. Paper WP5-C.

[152] G. D. Martin. Long-range predictive control. *AIChE Journal*, 27(5):748–753, September 1981.

[153] V. Peterka. On steady state minimum variance control strategy. *Kybernetica*, 8(3):219–232, 1972.

[154] A. Graham. *Kronecker Products and Matrix Calculus with Applications.* Ellis Horwood, England, 1981.

[155] R. Rouhani and R. K. Mehra. Model algorithmic control (MAC): Basic theoretical properties. *Automatica*, 18(4):401–414, July 1982.

[156] J. Richalet, A. Rault, J. L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5):413–428, September 1978.

[157] J. M. Martin-Sanchez and S. L. Shah. Multivariable adaptive predictive control of a binary distillation column. *Automatica*, 20(5):607–620, September 1984.

[158] F. M. D'Hulster, R. M. C. de Keyser, and A. R. van Cauwenberghe. Simulations of adaptive controllers for a paper machine headbox. *Automatica*, 19(4):407–414, July 1983.

[159] R. M. C. de Keyser and A. R. van Cauwenberghe. Typical application possibilities for self-tuning predictive control. In *IFAC Symposium on Identification and System Parameter Estimation*, pages 975–979, Washington D.C., USA, 1982.

[160] R. Isermann. *Digital Control Systems.* Springer-Verlag, Berlin, 1981.

[161] P. G. Kaminski, A. E. Bryson, and S. F. Schmidt. Discrete square root filtering: A survey of current techniques. *IEEE Trans. Automat. Control*, AC-16(6):727–735, December 1971.

[162] G. J. Bierman. Sequential square root filtering and smoothing of discrete linear systems. *Automatica*, 10(2):147–158, March 1974.

[163] G. J. Bierman. Measurement updating using the U-D factorization. In *Proc. 1975 IEEE Confr. Decision and Control*, pages 337–346, Houston, Tx, December 1975.

[164] G. J. Bierman. *Factorization Methods for Discrete Sequential Estimation.* Academic Press, New York, 1977.

[165] V. Sima. Factorization techniques in discrete time adaptive control. In *IFAC 9-th Triennial World Congress*, pages 1029–1034, Budapest, 1984.

[166] F. P. Demello and C. Concordia. Concepts of synchronous machine stability as affected by excitation control. *IEEE Trans. Power Apparatus and Systems*, PAS-88(4):316–327, April 1969.

[167] S. Cheng and G. S. Hope. Self-tuning stabiliser for a multimachine power system. *IEE Proc.*, 133, Pt. C(4):176–185, May 1986.

[168] A. Chandra, O. P. Malik, and G. S. Hope. A self-tuning controller for the control of multi-machine power systems. In *Proc. IEEE/PES Summer Meeting*, San Francisco, Ca, July 1987. Paper 452-6.

[169] D. W. Clarke and R. Hastings-James. Design of digital controllers for randomly disturbed systems. *Proc. IEE*, 118(10):1503–1506, October 1971.

[170] R. Schumann, K.-H. Lachmann, and R. Isermann. Towards applicability of parameter-adaptive control algorithms. In *IFAC 8-th Triennial World Congress*, pages 903–908, Kyoto, Japan, 1981.

[171] J. Neveu. *Discrete-Parameter Martingales.* North-Holland, Amsterdam, 1975.

[172] H. F. Chen. Recursive system identification and adaptive control by use of the modified least squares algorithm. In *22-nd IEEE Confr. Decision and Control*, pages 441–444, San Antonio, Tx, December 1983.

[173] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification.* The MIT Press, Cambridge, Massachusetts, 1983.

[174] P. C. Young. Recursive approaches to time series analysis. *Bull. Inst. Math. Appl.*, 10:209–224, 1974.

[175] T. Söderström. *An on-line algorithm for approximate maximum likelihood identification of linear dynamic systems.* Report 7308, Lund Inst. of Technology, Dept. of Automatic Control, Lund, Sweden, 1973.

[176] K. J. Åström and T. Bohlin. Numerical identification of linear dynamic systems from normal operating records. In P. H. Hammond, editor, *Theory of Self-Adaptive Systems*, pages 96–111, Plenum Press, New York, 1965.

[177] T. Söderström, L. Ljung, and I. Gustavsson. A theoretical analysis of recursive identification methods. *Automatica*, 14(3):231–244, May 1978.

[178] S. W. Chan, G. C. Goodwin, and K. S. Sin. Convergence properties of the Riccati difference equation in optimal filtering on nonstabilizable systems. *IEEE Trans. Automat. Control*, AC-29(2):110–118, February 1984.

[179] L. Hitz and B. D. O. Anderson. Discrete positive-real functions and their application to system stability. *IEE Proc.*, 116(1):153–155, January 1969.

[180] T. L. Lai and C. Z. Wei. Extended least squares and their applications to adaptive control and prediction in linear systems. *IEEE Trans. Automat. Control*, AC-31(10):898–906, October 1986.

[181] L. Ljung, T. Söderström, and I. Gustavsson. Counterexamples to general convergence of a commonly used recursive identification method. *IEEE Trans. Automat. Control*, AC-20(5):643–652, October 1975.

[182] D. L. Prager and P. E. Wellstead. Multivariable pole-assignment self-tuning regulators. *IEE Proc.*, 128, Pt. D(1):9–18, January 1981.

[183] E. I. Jury. *Sampled-Data Control Systems.* McGraw-Hill, New York, 1958.

[184] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials.* Academic Press, London, 1982.

[185] S. Zohar. Toeplitz matrix inversion: The algorithm of W. F. Trench. *J. ACM*, 16(4):592–601, October 1969.

[186] B. Friedlander. A modified prefilter for some recursive parameter estimation algorithms. *IEEE Trans. Automat. Control*, AC-27(1):232–235, February 1982.

[187] G. C. Goodwin and R. L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis.* Academic Press, New York, N.Y., 1977.

[188] J. B. Moore. *Stochastic adaptive control via consistent parameter estimation.* Technical Report, Australian National University, Department of Systems Engineering, Canberra ACT 2601 Australia, August 1984.

[189] P. E. Caines and S. Lafortune. Adaptive control with recursive identification for stochastic linear systems: Multivariable case. In *21-st IEEE Confr. Decision and Control*, pages 978–982, Orlando, Florida, December 1982.

[190] H. F. Chen and L. Guo. Convergence rate of least-squares identification and adaptive control for stochastic systems. *Int. J. Control*, 44(5):1459–1476, November 1986.

[191] B. D. O. Anderson and C. R. Johnson. Exponential convergence of adaptive identification and control algorithms. In *IFAC 8-th Triennial World Congress*, pages 911–916, Kyoto, Japan, 1981.

[192] B. D. O. Anderson. Exponential convergence and persistent excitation. In *21-st IEEE Confr. Decision and Control*, pages 12–17, Orlando, Florida, December 1982.

[193] P. J. Gawthrop. On the stability and convergence of a self-tuning controller. *Int. J. Control*, 31(5):973–998, May 1980.

[194] S. C. Shah and G. F. Franklin. On satisfying strict positive real condition for convergence by overparameterization. In *20-th IEEE Confr. Decision and Control*, pages 528–530, San Diego, Ca, December 1981.

[195] L. Ljung. Asymptotic behaviour of the extended Kalman filter as a parameter estimator for linear systems. *IEEE Trans. Automat. Control*, AC-24(1):36–50, February 1979.

[196] H. Weiss and J. B. Moore. Recursive prediction error algorithms without a stability test. *Automatica*, 16(6):683–687, November 1980.

[197] H. Kurz, R. Isermann, and R. Schumann. Experimental comparison and application of various parameter adaptive control algorithms. *Automatica*, 16(2):117–133, March 1980.