Data-Driven Methods for Tracking Improvement

THÈSE N^{O} 4408 (2009)

PRÉSENTÉE LE 5 JUIN 2009

À LA FACULTÉ SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

LABORATOIRE D'AUTOMATIQUE

PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Mark Edward John BUTCHER

acceptée sur proposition du jury:

Prof. D. Bonvin, président du jury
Prof. R. Longchamp, Dr A. Karimi, directeurs de thèse
Prof. A. Billard, rapporteur
Prof. S. Gunnarsson, rapporteur
Prof. M. Verhaegen, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2009

To my parents and my brother

Acknowledgements

The realisation of this thesis has only been possible due to the support of numerous people. I would therefore like to express my gratitude to all those that have played a role.

More specifically I would like to thank those that gave me the opportunity to join the Laboratoire d'Automatique, namely Professor Roland Longchamp, Professor Dominique Bonvin, Dr Denis Gillet and Dr Alireza Karimi. I especially want to thank Professor Roland Longchamp for his approachability in my first few uncertain weeks at the EPFL. I also would like to thank him, in his role as a co-director of this thesis, for his support and encouragement. I would like to express my gratitude to my other co-director, Dr Alireza Karimi, for his constant availability, reassurance and advice. The numerous discussions we had during the thesis, and the very astute insights he provided, were invaluable to the success of this work.

I gratefully acknowledge the financial support received from the Swiss Confederation.

I am honoured to have had as members of my thesis jury Prof. Aude Billard, Prof. Dominique Bonvin, Prof. Svante Gunnarsson, Dr Alireza Karimi, Prof. Roland Longchamp and Prof. Michel Verhaegen. Thank you for accepting to evaluate this thesis.

Thank you to Max Boegli, Ralph Coleman, Michel Mathia and Vincent Very of ETEL S.A. with whom I have had a very interesting and fruitful industrial collaboration throughout the duration of this thesis.

Thank you to all my colleagues at the Laboratoire d'Automatique for such an enjoyable working atmosphere. I would like to express my appreciation to the lab's 'support staff' for their assistance both in administrative matters, Francine, Homeira, Ruth and Sol, and for all things IT, Philippe and Christophe. Thank you to my housemates from the lab, Alejandro, Davide and Yvan, for their good company. Thank you also to Benoit, Gorka, Marc and Seb with whom life has continued beyond the physical boundaries of the lab, normally up a mountain! Many a good moment cycling, hiking and skiing were spent together and even more joking about them afterwards in the lab!

Thank you to all those friends that have made my Swiss life so memorable and enjoyable: Alejandro, Alexandre D., Alexandre D.F., Andrea, Annick, Benoit, Bruno, Clare, Davide, Davnah, Fanny, Frances, Gorka, Henri, Intan, Isis, Katatau, Kristijan, Lambert, Luis, Marc, Marcelo, Montse, Natalia, Nicholas, Paul, Paulo, Pedro, Pietro, Rosy, Sabrina, Saioa, Sami, Sandy, Seb, Simona, Steph, Victor, Wellerson, Yvan.

I am also indebted to my friends back home who have always remained close despite the physical distance between us.

I extend my thanks to Professor Arthur Dexter of the University of Oxford who gave me my first control course and thereby started the process that lead to this thesis.

I thank profoundly my parents and brother for their unfaltering love and support in all that I've done throughout my life.

Last, but by no means least, I would like to deeply thank Joana for her love and companionship, and the happiness she has given me. Moltes gracis!

Abstract

The tracking precision required by modern industrial applications is continuously increasing. Feedback control alone is often no longer capable of giving the necessary tracking accuracy and so the use of two-degree-of-freedom controllers, which include a feedforward term, has become commonplace. Traditionally the feedforward term is a filter based on the inverse of an identified model of the system. It is, however, not possible to obtain very high precision tracking with this approach because the identified model will always suffer from model uncertainty.

In this thesis, data-driven methods are investigated. These methods derive the feedforward control directly from measured data and thus avoid the system identification step, which is where the model uncertainty is introduced. They are, therefore, capable of producing higher precision tracking than the traditional methods.

For the general tracking problem, a precompensator controller is considered as the feedforward term. This controller filters the desired output signal before it is applied as an input to the system. The precompensator's parameters are tuned directly using measured data. These data are affected by stochastic disturbances, such as measurement noise. The effect of these disturbances on the calcu-

lated parameters is studied and the correlation approach is used to reduce it.

For the specific problem where the tracking task is repetitive, a situation frequently encountered in industrial applications, Iterative Learning Control is proposed. Iterative Learning Control uses measurements from previous repetitions to adjust the system's input for the current repetition in a manner that improves the tracking. As measurements are used, the calculated input is sensitive to the stochastic disturbances. The effect of these disturbances on the learning procedure is examined and algorithms, which are less sensitive to their presence, are developed.

Extensions of the methods are also made for linear parameter varying systems in which the system's dynamics change as a function of a scheduling parameter.

The developed methods are successfully applied to an industrial linear motor positioning system.

Keywords: tracking, data-driven controller tuning, feedforward, Iterative Learning Control, linear parameter varying systems

Résumé

La précision en poursuite requise par les applications industrielles modernes augmente sans cesse. La commande par rétroaction n'est souvent plus capable de produire le niveau de précision nécessaire. Ainsi les régulateurs à deux degrés de liberté, qui utilisent un terme de commande a priori, sont devenus répandus. Traditionnellement la commande a priori est un filtre basé sur l'inverse du modèle identifié du système à commander. Cependant, il n'est pas possible d'obtenir un asservissement de haute précision avec cette démarche à cause de l'incertitude associée au modèle identifié.

Dans cette thèse, des méthodes basées sur les données sont développées. Ces méthodes fournissent la commande a priori directement à partir des données mesurées et, ainsi, évitent l'étape d'identification du système, où l'incertitude est présente. Elles peuvent, par conséquent, conduire à une précision plus élevée qu'avec les méthodes traditionnelles.

Pour le problème général de la poursuite, un précompensateur est considéré pour le terme de commande a priori. Ce précompensateur filtre la sortie désirée du système à commander avant qu'elle soit appliquée comme entrée au système. Les paramètres du précompensateur sont dimensionnés directement en utilisant les données mesurées.

Ces données sont contaminées par des perturbations stochastiques, par exemples des bruits de mesure. L'effet de ces perturbations sur les paramètres estimés est étudié et l'approche de corrélation est utilisée pour le réduire.

Pour le problème spécifique d'une poursuite répétitive, une situation souvent rencontrée en industrie, la commande itérative basée sur l'apprentissage (ILC de 'Iterative Learning Control' en anglais) est suggérée. L'approche ILC utilise des mesures des itérations précédentes pour modifier l'entrée du système à commander lors de l'itération actuelle pour que la poursuite soit améliorée. Comme des mesures sont utilisées, l'entrée calculée est sensible aux perturbations stochastiques présentes dans ces mesures. L'effet de ces perturbations sur le processus d'apprentissage est examiné et des algorithmes moins sensibles à leur présence sont développés.

Les méthodes sont étendues aux systèmes linéaires à paramètres variants.

Les méthodes développées sont finalement implémentées avec succès sur un système de positionnement industriel.

Mots-clés : poursuite, commande a priori, synthèse d'un régulateur basée sur les données, commande itérative basée sur l'apprentissage, systèmes linéaires à paramètres variants

Contents

Frequently used nomenclature

Abbreviations and acronyms

a.s.	almost sure(ly)
ILC	iterative learning control
IV	instrumental variables
LPV	linear parameter varying
LPMSM	linear, permanent magnet, synchronous motor
LTI	linear time invariant
LTV	linear time varying
RMS	root mean square
PID	proportional-integral-derivative
PRBS	pseudo random binary signal
SA	stochastic approximation
SISO	single input single output
SPR	strictly positive real
SLLN	strong law of large numbers
w.r.t.	with respect to
w.p.	with probability

Operators

$\arg\min_x f(x)$	minimising argument of $f(x)$
$\mathbf{A}^{-1}$	inverse of the matrix $\mathbf{A}$
$\mathbf{A}^T$	transpose of the matrix $\mathbf{A}$
$\mathrm{cov}(\mathbf{x})$	covariance matrix of the vector $\mathbf{x}$
$\exp(\cdot)$	exponential function
$E\{\cdot\}$	mathematical expectation
$\hat{E}\{\cdot\}$	estimate of $E\{\cdot\}$
$\lambda(\mathbf{A})$	eigenvalue of the matrix $\mathbf{A}$
$\overline{\lambda}(\mathbf{A})$	maximum eigenvalue of the matrix $\mathbf{A}$
$\underline{\lambda}(\mathbf{A})$	minimum eigenvalue of the matrix $\mathbf{A}$
$\mathcal{F}\{\cdot\}$	Fourier transform
inf	infimum
$\mathbf{x}_N \in \mathrm{As}\ \mathcal{N}(\mathbf{m}, \mathbf{V})$	sequence of random vectors $\mathbf{x}_N$ converges in distribution to the normal distribution with mean $\mathbf{m}$ and covariance matrix $\mathbf{V}$
$P(A)$	the probability of event A
q	forward-shift time-domain operator
q^{-1}	backward-shift time-domain operator
sup	supremum
$\mathrm{sol}[c(\mathbf{x}) = 0]$	$\mathbf{x}$ is the solution to the equation $c(\mathbf{x}) = 0$
$\sigma(\mathbf{A})$	singular value of the matrix $\mathbf{A}$
$\overline{\sigma}(\mathbf{A})$	maximum singular value of the matrix $\mathbf{A}$
$\underline{\sigma}(\mathbf{A})$	minimum singular value of the matrix $\mathbf{A}$
$\mathrm{tr}(\mathbf{A})$	trace of the matrix $\mathbf{A}$
w	forward-shift iteration-domain operator
$\chi^2_\alpha(l)$	α-level of the $\chi^2(l)$ distribution with l degrees of freedom
$\mathcal{Z}\{\cdot\}$	z-transform
$\langle \cdot, \cdot \rangle$	inner product of two vectors
$\| \cdot \|$	norm of a vector, matrix or system
$\| \cdot \|_F$	Frobenius norm

Symbols

Latin symbols

$d(t)$	load disturbance
$\mathbf{d}$	supervector of $d(t)$
$e(t)$	measured tracking error
$\mathbf{e}$	supervector of $e(t)$
$\hat{e}(t)$	estimate of $e(t)$
$F(q^{-1})$	LTI precompensator
$F(\boldsymbol{\sigma}(t), q^{-1})$	LPV precompensator
$G(q^{-1})$	LTI discrete-time system
$G(\boldsymbol{\sigma}(t), q^{-1})$	LPV discrete-time system
$G_c(s)$	LTI continuous-time system
$\hat{G}(q^{-1})$	model of the system $G(q^{-1})$
$\mathbf{G}$	matrix representation of $G(q^{-1})$
$\hat{\mathbf{G}}$	matrix representation of $\hat{G}(q^{-1})$
$H_u(q^{-1})$	LTI input noise filter
$H_u(\boldsymbol{\sigma}(t), q^{-1})$	LPV input noise filter
k	iteration/repetition number
$L(q^{-1})$	ILC learning filter
$\mathbf{L}$	matrix representation of $L(q^{-1})$
$J(\cdot)$	cost function
$J^N(\cdot)$	cost function based on N data points
m	system's relative degree
$n(t)$	measurement disturbance
n_x	dimension of the vector $\mathbf{x}$
$\mathbf{n}$	supervector of $n(t)$
N	number of data points
$r_{ij}(\tau)$	cross-correlation function at lag τ between signals $i(t)$ and $j(t)$
$\mathbf{R_i}$	covariance matrix of vector $\mathbf{i}$
$\mathbf{R_{ij}}$	cross-covariance matrix of vectors $\mathbf{i}$ and $\mathbf{j}$
$\mathbb{R}$	space of real numbers
$\mathbb{R}^i$	space of real i-dimensional vectors
$\mathbb{R}^{i\times j}$	space of real $i \times j$ matrices

t	time
$u(t)$	system input
$\mathbf{u}$	supervector of $u(t)$
$\mathbf{u}_0$	ideal input vector giving zero mean controlled error
$y(t)$	measured output
$\mathbf{y}$	supervector of $y(t)$
$y_d(t)$	desired output
$\mathbf{y_d}$	supervector of $y_d(t)$
$z(t)$	controlled output
$\mathbf{z}$	supervector of $z(t)$

Greek symbols

$\boldsymbol{\beta}(q^{-1})$	vector of transfer operators
δ	number of time steps of preview
$\Delta(q^{-1})$	multiplicative uncertainty
$\boldsymbol{\Delta}$	matrix representation of $\Delta(q^{-1})$
$\epsilon(t)$	controlled tracking error
$\boldsymbol{\epsilon}$	supervector of $\epsilon(t)$
$\boldsymbol{\zeta}(t)$	instrumental variable vector
$\eta(t)$	sequence of independent, zero-mean random variables
$\boldsymbol{\eta}(t)$	sequence of independent, zero-mean random vectors
$\boldsymbol{\rho}$	vector of precompensator parameters
$\boldsymbol{\rho}_0$	parameters of the ideal precompensator
$\sigma_x^2(t)$	variance of the random process $x(t)$
$\sigma(t)$	single scheduling parameter
$\boldsymbol{\sigma}(t)$	scheduling parameter vector
$\boldsymbol{\phi}(t)$	regressor vector
$\Phi_i(\omega)$	spectral density of signal i
$\Phi_{ij}(\omega)$	cross-spectral density between signals i and j
ω	angular frequency

1

Introduction

1.1 Motivation

Trajectory tracking is essential in a multitude of applications. Telescopes should track the trajectory of the stars through the night's sky in order to obtain a clear image, robots spray painting car bodies should follow the correct path to produce an even coat and milling machines need to stick closely to the programmed position track in order to manufacture usable components. These examples are just a fraction of the diverse applications where trajectory tracking is necessary.

Typically, feedback loops are used to obtain the required precision. These loops can produce reasonable tracking performance, in addition to their ability to stabilise a system, reject external disturbances and be robust to model uncertainty.

Modern assemblies and mechanisms are, however, being continuously miniaturised and the required tracking precision is, thus, also continuously increasing. For example, in order to store more data on hard disk drives, the data tracks on the disk are being placed closer together and so the accuracy with which the reading head must follow each track is increasing. Similarly, to make smaller portable

electronic devices, circuits must be smaller and thus the components placed closer together. The manufacturing process of fitting these components, in turn, becomes trickier. In addition, in order to save time (to retrieve data more quickly, reduce manufacturing costs etc.), the speed with which these trajectories must be tracked precisely is also increasing. Feedback control alone is not capable of achieving these two conflicting exigencies simultaneously since a compromise of tracking performance versus stability and high-frequency disturbance attenuation should always be made. For modern, high precision tracking tasks feedforward control must thus be used in addition to feedback control in a two-degree-of-freedom controller structure. Feedforward control typically uses a model of the system to back calculate the input needed by the system to make it follow the trajectory correctly. Unlike feedback control, stability is ensured, if the feedforward controller is stable, and noisy measurements are not used so disturbance amplification is not an issue.

Model-based feedforward controllers are normally thus taken as the stable inverse of an identified model of the system performing the tracking task. The feedforward input signal to the system is generated by filtering the desired output trajectory with the feedforward controller. The tracking achieved using this traditional approach is, however, very dependent on the quality of the model, i.e. how well it represents the true system [13, 16].

Data-driven methods for feedforward control, on the other hand, generate the feedforward control directly using data measured from the system. They, therefore, avoid the use of an uncertain system model and, consequently, can produce improved tracking performance over model-based techniques.

A reasonable approach for the general tracking problem is to use data-driven methods to directly identify the system inverse from measured data and use it as a precompensator/prefilter that filters the desired output before it is applied to the system's input.

For the specific class of tracking problems where the tracking task is repetitive it makes sense to use information available from previous repetitions to improve the performance in the current one. This

principle is employed by Iterative Learning Control (ILC), which is a data-driven method using measurements from previous repetitions to adapt the feedforward signal so as to improve the current repetition's tracking.

A major problem encountered by data-driven methods, however, is the presence of stochastic disturbances affecting the system and measurements. These disturbances mean the measurements contain false information about the system and so the control signals produced by the data-driven methods do not lead to the desired tracking performance. For this reason, it is important to understand the effect of these stochastic disturbances on data-driven methods and to develop new methods that are less sensitive to them.

Furthermore, it cannot always be assumed that real systems are well represented by the linear time invariant (LTI) class of systems, as is often done due to this class' mathematical tractability. If this assumption is not valid and an LTI feedforward controller is used, then the good tracking achieved at one operating point may not be replicated at other operating points. Good tracking across the entire operating range can only be achieved, in this case, by using feedforward control for broader system classes. One such class is that of linear parameter varying (LPV) systems. LPV systems can represent a large number of practically encountered systems whose dynamics change as a function of the operating point, but still retain the useful properties of linear systems.

1.2 State of the art

This thesis makes contributions in data-driven feedforward controller/precompensator tuning and ILC. Methods are proposed for both LTI and LPV systems. This section will provide a non-exhaustive overview of related work in these areas.

1.2.1 Data-driven feedforward controller/precompensator tuning

Data-driven controller tuning can be seen as a subcategory of direct adaptive control. The main difference with standard direct adaptive control is the adaptation period. Whereas in the standard case the controller parameters are adjusted at every sampling instant, in data-driven controller tuning the controller parameters are adjusted based on batches of data i.e. a large number of samples, typically making this approach less prone to stability problems. Nonetheless, like the standard method, the controller parameters should be directly calculated using real measured data and so similar parameter estimation algorithms can be used. Research into data-driven controller tuning has been active in recent years.

Iterative Feedback Tuning is a model-free approach for tuning the parameters of two-degree-of-freedom controllers such that the tracking error variance is minimised. It uses specific closed-loop experiments to compute an unbiased estimate of the gradient of the control criterion [23, 24]. Separate tuning of the feedforward and the feedback controllers is proposed to improve the tracking performance using the Iterative Feedback Tuning approach in [19]. Another iterative method based on the correlation approach has been proposed and successfully applied to a magnetic suspension system in [27] as well as a benchmark problem in [36]. An overview of this approach can be found in [28]. The main idea is that instead of minimising a norm of the tracking error, it is made uncorrelated with the reference signal. It can be shown that the controller parameter estimates are asymptotically unaffected by stochastic disturbances.

A disadvantage of these iterative approaches is that they require many experiments to be done on the real system for criterion evaluation. The main interest of the controller tuning method Virtual Reference Feedback Tuning is that only one set of data is required to tune a controller for the model reference problem [10]. In this approach the controller tuning problem is transformed into an identification problem by defining a virtual reference signal. An extension

of this method to two-degree-of-freedom controllers is given in [29]. Virtual Reference Feedback Tuning, however, requires the inverse of the reference model to be used as a filter, which if the reference model contains unstable zeros can be problematic. Furthermore it proposes using either a second set of measured data or a model of the system to obtain unbiased controller parameter estimates in the presence of stochastic disturbances.

In [58] a method for designing precompensators directly using data in the frequency domain is proposed. Whilst this approach is well suited to the model reference problem, time-domain tracking specifications must be approximated in the frequency domain.

1.2.2 Iterative learning control

ILC can be regarded as drawing on the basic principles of human learning, with the adage 'practice makes perfect' being taken as its key concept. Typically for a human to learn a task he or she repeats the task a number of times and after each repetition analyses where errors were made. He or she then makes adjustments in the next repetition in order to perform the task better. ILC uses this same principle to ameliorate a repetitive system's tracking performance.

It is generally accepted that the first publications on ILC were [59], which was written in Japanese in 1978, and [4], written in English in 1984. Since then the subject has been intensely researched producing a number of surveys [1, 8, 38], books, theses and several hundred papers.

The technique has been demonstrated to be capable of considerably improving the tracking performance of systems that are predominately affected by deterministic disturbances. Since these are repeated from one repetition to the next, ILC is capable of learning them and, in turn, compensating for them. However, when the system is affected by stochastic, non-repeating disturbances, such as measurement noise, the achievable performance is greatly degraded.

Some research has already been done into the influence of disturbances on ILC. In [47] a disturbance analysis is done and both

recursive and explicit expressions for the measured error in terms of desired output and disturbances are found. They are used to discuss generally how the presence of disturbances affects the measured error evolution. In [35] a similar analysis is done and simulation and experimental results are used to illustrate the general effect of iteration-dependent disturbances. In [18] analytical expressions for the covariance matrix of the controlled error are developed for high order ILC algorithms with load and measurement disturbances separately.

Recognising the degrading effect of stochastic iteration-dependent disturbances, certain researchers have proposed algorithms which are less sensitive to their presence.

The use of a forgetting factor in ILC was first proposed in [22] for a derivative-type ILC law. It was then proposed in [3] for proportional-type ILC. It is shown that by introducing the forgetting factor the system's output converges to a neighbourhood of the desired one, despite the presence of norm-bounded initialisation errors, fluctuations of the dynamics and random disturbances. However, in [50], it is shown that the trace of the input error covariance matrix is minimised when no forgetting factor is used.

The filtering of the ILC command has been proposed in certain papers as a way of reducing the influence of noise on the error [44]. However, whilst it reduces the error's variance, it causes a nonzero converged mean error.

Kalman filtering-type techniques have also been applied to ILC to estimate the controlled output, in the presence of stochastic disturbances [2,14,43,51,52,56]. In the case of perfect knowledge of the disturbance covariance matrices and system parameters, convergence to the optimal input can be shown. However, this perfect knowledge is unrealistic.

In [56] an ILC algorithm is proposed using a learning gain that decreases inversely proportionally to the iteration number and has the form of a stochastic approximation (SA) algorithm. No detailed analysis is, however, carried out. An algorithm with a similar iteration decreasing learning gain is also developed in [46] for repetitive

disturbance rejection in the presence of measurement noise. This algorithm is derived in a similar way to recursive least squares identification algorithms. The direct application of SA theory to ILC is considered in [11] and [12] for the linear and nonlinear cases respectively. It is shown that the proposed ILC law converges almost surely to the optimal input and the output error is minimised in the mean square sense as the number of iterations tends to infinity. The algorithm requires only that the optimal input is realisable. Knowledge of neither the disturbance covariance matrix nor the system matrices is required because a simultaneous perturbation type algorithm is employed, which uses random perturbations to estimate the gradient. The disadvantage of this approach is slow convergence.

1.2.3 LPV systems

In many real world control applications an LTI assumption is made on the system's dynamics in order to use the well developed identification and control techniques available for this class of systems. This assumption is valid when the system remains within a certain operating zone. However, when the operating point changes considerably the identified model is no longer valid and the controlled system's performance is degraded, and in the worst case it becomes unstable. In order to achieve good performance throughout the entire operating region, but still use the linear system techniques, the class of linear parameter varying systems has, therefore, been defined [53].

LPV systems can be thought of as a weighted combination of linear models, each valid at a specific operating point. The weightings are a function of the operating point, which is, in turn, a function of certain scheduling parameters. These scheduling parameters can either be endogenous signals, such as the system's states or outputs, or exogenous signals, which cause the dynamics to change as a function of time according to the trajectories of these signals. The former case is sometimes referred to as Local Linear Models [61]. The latter means that the LPV system represents a family of linear time vary-

ing (LTV) systems, each system corresponding to a particular set of trajectories.

For LPV systems methods have been proposed [17, 48] to tune precompensators and feedforward controllers whose parameters also vary as a function of the operating point. These methods, however, are based on uncertain identified LPV models and thus, unlike direct data-driven methods, suffer from model uncertainty.

No data-driven precompensator, or feedforward controller, tuning methods for LPV systems have been proposed to the author's knowledge. But, as is the case for LTI systems, parameter estimation and system identification techniques are adaptable to the tuning of these controllers. A brief review of the rapidly expanding LPV system identification literature is therefore now given.

In [41] a method is proposed for the identification of LPV state-space models. It, however, requires full-state measurement, which is often not possible, and is only capable of handling one scheduling parameter. Multiple scheduling parameters can be dealt with using the method proposed in [34], but full-state measurement is still required. In [62] full-state knowledge is no longer necessary as a scheme for estimating the states is proposed. In order to obtain consistent estimates of the states, when the scheduling parameter is noise-free, the use of instrumental variables (IV) is proposed. When the LPV system to be identified has a large number of inputs and outputs the dimensions of the matrices involved in this approach can become too large for standard computers to handle. For this reason, a modification using a kernel method has been proposed [63] to reduce the size of the matrices, at the cost of only identifying an approximation of the system.

The identification of a linear fractional transformation representation of LPV state-space models is examined in [30]. Non-linear programming is used to minimise a quadratic cost function, making the parameter estimates sensitive to the choice of initial estimates.

In [5] a method is proposed for the identification of the parameters of single input single output (SISO) LPV systems in input-output form. Each parameter of the system transfer operator is a

linear combination of predefined, operating point dependent functions. The identification procedure is then one of estimating the coefficients multiplying these functions, which is a linear regression problem, and so can be computed using the standard linear least squares technique. However, as occurs in the LTI case, the linear least squares technique generally gives biased parameter estimates in the presence of stochastic disturbances. In [9] (see appendix) it is shown that, when certain conditions on the scheduling parameter dependence are satisfied, consistent estimates can be obtained using instrumental variables for both the noise-free and noisy scheduling parameter measurement cases.

In [57] an extension of the well-established orthonormal basis function identification framework for LTI systems to LPV systems has been made. Orthonormal basis function models have the nice properties of being linear in their parameters, having a noise model that is independently parameterised from the system and being parsimonious when a good choice is made of the basis functions. A good choice of basis functions is, however, critical.

Regarding ILC for LPV systems, the only publication in this area, to the author's knowledge, is [31]. There the variation of the system's dynamics due to the changing scheduling parameter is assumed to take place during the iteration and is the same for all iterations. A gain scheduling type approach is used whereby a stabilising gain is found at a fixed time in the iteration to ensure convergence in the iterations at that instant. The gains for a number of fixed times are then interpolated to give the global ILC controller.

1.3 Organisation and contributions of the thesis

In Chapter 2 some background information is given. This information takes the form of basic notation, principles, theorems and details of the linear motor positioning system that constitutes the application in this thesis.

The main contributions of this thesis are presented in Chapters 3-7. In Chapters 3 and 4, data-driven methods for precompensator tuning for LTI and LPV systems, respectively, will be studied. Then in Chapters 5-7 ILC for LTI and LPV systems will be investigated. The individual chapters are described in more detail below.

In Chapter 3 a data-driven precompensator tuning method for LTI systems is proposed. The method uses data from only one experiment on the real system and employs the correlation approach to estimate parameters that are asymptotically unaffected by stochastic disturbances. By a frequency-domain analysis of the criterion, it is shown that the model-following criterion is minimised. The method is successfully applied to the tuning of a precompensator for the linear motor system.

Methods for direct data-driven tuning of LPV precompensators are developed in Chapter 4. Since LPV operators do not generally commute, the previously proposed method for LTI systems cannot be directly adapted. When the ideal precompensator giving perfect mean tracking exists in the proposed precompensator parameterisation, the LPV transfer operators do commute, nonetheless, and an algorithm using only two experiments on the real system is proposed. It is shown that this algorithm gives consistent estimates of the ideal parameters despite the presence of stochastic disturbances. For the more general case, when the ideal precompensator does not belong to the set of parameterised precompensators, another technique is developed. This technique requires a number of experiments equal to twice the number of precompensator parameters and it is shown that the calculated parameters minimise the mean squared tracking error. The theoretical results are demonstrated in simulation.

Chapters 5 and 6 study ILC for LTI systems. In Chapter 5 new expressions for the mean value and variance of the controlled error signal, obtained using a general LTI ILC algorithm, are developed. They are then used to analyse the statistical properties of certain, previously proposed ILC algorithms. The analysis is illustrated via simulation and experimental results. In Chapter 6 it is shown how ILC fits into the well-established stochastic approximation theory

framework. Using this framework, conditions are given that ensure almost sure convergence of the ILC input to the optimal input in the presence of stochastic disturbances. Specific choices for the algorithm's learning gain are proposed and tested in simulation and experimentally.

In Chapter 7 an ILC algorithm is proposed for a certain class of LPV systems whose dynamics change between iterations. Consistency of the algorithm in the presence of stochastic disturbances is shown. The proposed algorithm is applied to the linear motor positioning system, which is shown to be an LPV system for a specific class of movements. The tracking performance obtained using the proposed algorithm is compared with that obtained using an LTI ILC algorithm and better results are achieved.

Finally in Chapter 8 the results of the thesis are summarised and recommendations for possible future work are made.

2

Preliminaries

In this chapter some background information is given that will be used in the thesis.

2.1 Basic notation

The controlled output of an LTI, discrete-time, causal SISO system $G(q^{-1})$, shown in Figure 2.1, at time t is given by:

$$z(t) = G(q^{-1})u(t) + d(t), \tag{2.1}$$

where $u(t)$ is the system input, $d(t)$ is the load disturbance and q^{-1} is the backward-shift time-domain operator. The system's measured output is:

$$y(t) = z(t) + n(t), \tag{2.2}$$

where $n(t)$ is the measurement disturbance.

$G(q^{-1})$ may represent an open or closed-loop system. When the latter case applies, $d(t)$ and $n(t)$ will be signals resulting from the filtering of external disturbances by closed-loop transfer functions i.e. for the closed-loop system in Figure 2.2 we have that:

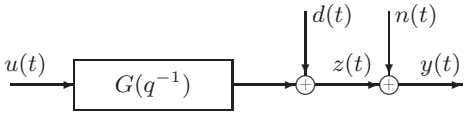

Fig. 2.1. System affected by disturbances

$$G(q^{-1}) = \frac{P(q^{-1})K(q^{-1})}{1 + P(q^{-1})K(q^{-1})}, \tag{2.3}$$

$$d(t) = \frac{1}{1 + P(q^{-1})K(q^{-1})} \left(d'(t) - P(q^{-1})K(q^{-1})n'(t)\right) \tag{2.4}$$

and $$n(t) = n'(t). \tag{2.5}$$

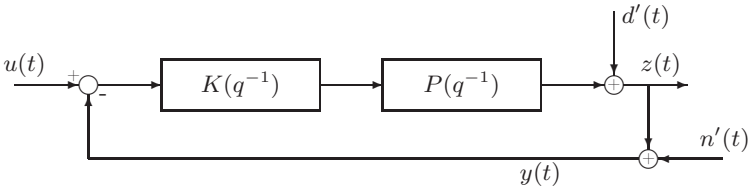

Fig. 2.2. Closed-loop system affected by disturbances

In general, each of the signals $d(t)$ and $n(t)$ may be either deterministic or stochastic, or contain both deterministic and stochastic components.

The controlled tracking error signal is defined as:

$$\epsilon(t) = y_d(t) - z(t), \tag{2.6}$$

where $y_d(t)$ is the desired system output and the measured tracking error signal is given by:

$$e(t) = y_d(t) - y(t). \tag{2.7}$$

When considering repetitive processes, the signals are indexed by the repetition number k as well as the time e.g. $u(t,k)$, $z(t,k)$ etc.

The controlled output of an LPV, discrete-time, causal SISO system $G(\boldsymbol{\sigma}(t), q^{-1})$ is given by:

$$z(t, \boldsymbol{\sigma}(t)) = G(\boldsymbol{\sigma}(t), q^{-1})u(t) + d(t, \boldsymbol{\sigma}(t)) \tag{2.8}$$

where $\boldsymbol{\sigma}(t) \in \mathbb{R}^{n_\sigma}$ is a scheduling parameter vector at time t, which contains the measurable signals that correspond to the system's current operating point. Likewise the measured output, $y(t, \boldsymbol{\sigma}(t))$, is:

$$y(t, \boldsymbol{\sigma}(t)) = z(t, \boldsymbol{\sigma}(t)) + n(t, \boldsymbol{\sigma}(t)). \tag{2.9}$$

The disturbances are now possibly functions of $\boldsymbol{\sigma}(t)$.

2.2 Correlation approach and instrumental variables in system identification

In certain chapters of this thesis use will be made of the correlation approach and instrumental variables. They are used extensively in the field of system identification in order to obtain consistent parameter estimates in the presence of coloured disturbances. They are briefly presented below.

We consider that the measured output of an LTI system is given by:

$$y(t) = \boldsymbol{\phi}^T(t)\boldsymbol{\theta}_0 + v(t), \tag{2.10}$$

where $\boldsymbol{\phi}(t)$ is a regressor vector containing time-shifted versions of the input and output, $\boldsymbol{\theta}_0$ is the vector of true system parameters and $v(t)$ is a zero-mean disturbance that is uncorrelated with the system input. A predictor, in the form of a linear regression model, is:

$$y_p(t|\boldsymbol{\theta}) = \boldsymbol{\phi}^T(t)\boldsymbol{\theta}. \tag{2.11}$$

The philosophy behind the correlation approach is that if the model represents the true system perfectly then the prediction error $e_p(t, \boldsymbol{\theta}) = y(t) - y_p(t|\boldsymbol{\theta})$ will only contain a contribution from the disturbance. Since the disturbance is uncorrelated with the input, the prediction error will also be uncorrelated with it in this case. The correlation approach therefore proposes estimating the system's parameters by decorrelating the prediction error from the system input. This idea can be formulated mathematically as:

$$\hat{\boldsymbol{\theta}}^N = \text{sol}\left[\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\zeta}(t,\boldsymbol{\theta})e_p(t,\boldsymbol{\theta}) = \mathbf{0}\right], \tag{2.12}$$

where N is the number of data points and $\boldsymbol{\zeta}(t, \boldsymbol{\theta})$ is a vector of instrumental variables that is correlated with the input $u(t)$ and uncorrelated with $v(t)$. In the linear case reviewed here, an explicit expression for the parameter estimate is given by:

$$\hat{\boldsymbol{\theta}}^N = \left[\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\zeta}(t)\boldsymbol{\phi}^T(t)\right]^{-1}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\zeta}(t)y(t). \tag{2.13}$$

The fact that this estimate can be consistent is seen by first replacing $y(t)$ with (2.10), giving:

$$\hat{\boldsymbol{\theta}}^N = \boldsymbol{\theta}_0 + \left[\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\zeta}(t)\boldsymbol{\phi}^T(t)\right]^{-1}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\zeta}(t)v(t). \tag{2.14}$$

The estimate is then consistent when the following conditions are satisfied:

i) $\lim_{N\to\infty}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\zeta}(t)\boldsymbol{\phi}^T(t)$ is nonsingular.

ii) $\lim_{N\to\infty}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\zeta}(t)v(t) = \mathbf{0}$.

When the input is 'sufficiently exciting', the first condition is satisifed since $\boldsymbol{\zeta}(t)$ is correlated with the non-noisy component of $\boldsymbol{\phi}(t)$. The second condition is also fulfilled because $\boldsymbol{\zeta}(t)$ is uncorrelated with $v(t)$.

2.3 Stochastic approximation

Robbins and Monro founded the subject of stochastic approximation with their seminal paper [49]. There they addressed the problem of using successive approximations $\theta(i)$ to find the unique scalar root θ_0 of a function $b(\theta)$, when only noisy measurements of the function are available. In their paper mean-square convergence of $\theta(i)$ to the root was proved. Later Blum [7] showed almost sure (a.s.) convergence for the multi-dimensional case. Blum's result will be used later in this thesis and so is presented below.

Theorem 2.1 *Suppose that for the vector* $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ *there exists a random vector* $\mathbf{a}(\boldsymbol{\theta})$ *such that* $\mathbf{b}(\boldsymbol{\theta}) = E\{\mathbf{a}(\boldsymbol{\theta})\} \in \mathbb{R}^{n_\theta}$. *Now let* $c(\boldsymbol{\theta})$ *be a real-valued function possessing continuous partial derivatives of first and second order. The vector of first partial derivatives is denoted by* $\mathbf{q}_c(\boldsymbol{\theta})$ *and the matrix of second partial derivatives by* $\mathbf{M}_c(\boldsymbol{\theta})$. *Then, for any real number* γ, *by Taylor's theorem one has:*

$$c\left(\boldsymbol{\theta}+\gamma\mathbf{a}(\boldsymbol{\theta})\right) = c(\boldsymbol{\theta}) + \gamma\langle\mathbf{q}_c(\boldsymbol{\theta}), \mathbf{a}(\boldsymbol{\theta})\rangle + \frac{1}{2}\gamma^2\langle\mathbf{a}(\boldsymbol{\theta}), \mathbf{M}_c\left(\boldsymbol{\theta}+\kappa\gamma\mathbf{a}(\boldsymbol{\theta})\right)\mathbf{a}(\boldsymbol{\theta})\rangle, \quad (2.15)$$

where κ *is a real number with* $0 \leq \kappa \leq 1$. *By taking expectations on both sides we get:*

$$E\{c\left(\boldsymbol{\theta}+\gamma\mathbf{a}(\boldsymbol{\theta})\right)\} = c(\boldsymbol{\theta}) + \gamma\langle\mathbf{q}_c(\boldsymbol{\theta}), \mathbf{b}(\boldsymbol{\theta})\rangle + \frac{1}{2}\gamma^2 E\{\langle\mathbf{a}(\boldsymbol{\theta}), \mathbf{M}_c\left(\boldsymbol{\theta}+\kappa\gamma\mathbf{a}(\boldsymbol{\theta})\right)\mathbf{a}(\boldsymbol{\theta})\rangle\}. \quad (2.16)$$

Let $\gamma(k)$ be a sequence of positive numbers and consider the following sequence of recursively defined random vectors:

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \gamma(k)\mathbf{a}(\boldsymbol{\theta}(k)). \tag{2.17}$$

Moreover, assume that $\mathbf{b}(\mathbf{0}) = \mathbf{0}$, without loss of generality, and consider the following set of conditions:

$$i)\ \sum_{k=0}^{\infty}\gamma(k) = \infty \quad and \quad \sum_{k=0}^{\infty}\gamma^2(k) < \infty. \tag{2.18}$$

$$ii)\ c(\boldsymbol{\theta}) \geq 0. \tag{2.19}$$

$$iii)\ \sup_{\Xi \leq \|\boldsymbol{\theta}\|} \langle \mathbf{q}_c(\boldsymbol{\theta}), \mathbf{b}(\boldsymbol{\theta})\rangle < 0 \text{ for every } \Xi > 0. \tag{2.20}$$

$$iv)\ \inf_{\Xi \leq \|\boldsymbol{\theta}\|} \|c(\boldsymbol{\theta}) - c(\mathbf{0})\| > 0 \text{ for every } \Xi > 0. \tag{2.21}$$

$$v)\ E\{\langle \mathbf{a}(\boldsymbol{\theta}), \mathbf{M}_c\left(\boldsymbol{\theta} + \kappa\gamma\mathbf{a}(\boldsymbol{\theta})\right)\mathbf{a}(\boldsymbol{\theta})\rangle\} < V_\gamma < \infty \text{ for every number } \gamma. \tag{2.22}$$

Then the sequence $\boldsymbol{\theta}(k)$, defined by (2.17), converges to zero a.s..

In Theorem 2.1 the $\langle\cdot,\cdot\rangle$ operator denotes the inner product of two vectors and $\|\cdot\|$ denotes the norm of a vector.

Remark: The conditions on the gains or step sizes $\gamma(k)$ in (2.18) ensure a balance between damping out the oscillations about the true root due to the noise ($\sum_{k=0}^{\infty}\gamma^2(k) < \infty$) but not stopping prematurely before reaching the true solution ($\sum_{k=0}^{\infty}\gamma(k) = \infty$).

Also of interest, other than conditions for almost sure convergence, is the speed of convergence of the estimate to the true root. A theorem, which will also be used later in the thesis, is now presented which provides sufficient conditions for the sequence $\sqrt{k}(\boldsymbol{\theta}(k) - \boldsymbol{\theta}_0)$ to converge to a normal distribution [42].

Theorem 2.2 *Consider the process defined in (2.17) with initial condition $\boldsymbol{\theta}(0) = \boldsymbol{\theta}$. Let the sequence $\gamma(k)$ be given by $\gamma(k) = \frac{\alpha}{k+1}$,*

where α is a positive constant. Suppose that the following conditions are satisfied:

i) The process defined in (2.17) converges a.s. to $\boldsymbol{\theta}_0$ as $k \to \infty$ and is a Markov process.

ii) The function $\mathbf{b}(\boldsymbol{\theta})$ may be represented by $\mathbf{b}(\boldsymbol{\theta}) = \mathbf{Q}(\boldsymbol{\theta}_0)(\boldsymbol{\theta} - \boldsymbol{\theta}_0) + o(|\boldsymbol{\theta} - \boldsymbol{\theta}_0|)$, where the matrix $\mathbf{D} = \frac{1}{2}\mathbf{I} + \alpha\mathbf{Q}(\boldsymbol{\theta}_0)$ is stable in the sense that its eigenvalues have negative real parts.

iii) All the elements of the matrix

$$E\left\{[\mathbf{a}(\boldsymbol{\theta}(k)) - \mathbf{b}(\boldsymbol{\theta}(k))]\,[\mathbf{a}(\boldsymbol{\theta}(k)) - \mathbf{b}(\boldsymbol{\theta}(k))]^T\right\}$$

are finite for $k \geq 0$ and

$$\lim_{k\to\infty} \lim_{\boldsymbol{\theta}(k)\to\boldsymbol{\theta}_0} E\left\{[\mathbf{a}(\boldsymbol{\theta}(k)) - \mathbf{b}(\boldsymbol{\theta}(k))]\,[\mathbf{a}(\boldsymbol{\theta}(k)) - \mathbf{b}(\boldsymbol{\theta}(k))]^T\right\} = \lim_{k\to\infty} E\left\{\mathbf{a}(\boldsymbol{\theta}_0)\mathbf{a}^T(\boldsymbol{\theta}_0)\right\} = \mathbf{P}.$$

iv) For some $\kappa > 0$,

$$\lim_{\Xi\to\infty} \sup_{|\boldsymbol{\theta}(k)-\boldsymbol{\theta}_0|<\kappa} \sup_{k\geq 1} \int_{|\mathbf{a}(\boldsymbol{\theta}(k))-\mathbf{b}(\boldsymbol{\theta}(k))|>\Xi} |\mathbf{a}(\boldsymbol{\theta}(k)) - \mathbf{b}(\boldsymbol{\theta}(k))|^2 \mathcal{P} d\omega = 0$$

with Ξ being a real variable and $\mathcal{P}$ a probability density function corresponding to $\mathbf{a}(\boldsymbol{\theta}(k))$.

Then the sequence $\sqrt{k}(\boldsymbol{\theta}(k) - \boldsymbol{\theta}_0) \in As\,\mathcal{N}(\mathbf{0}, \mathbf{V})$ i.e it converges asymptotically in distribution to a zero-mean normal distribution with covariance:

$$\mathbf{V} = \alpha^2 \int_0^\infty \exp(\mathbf{D}\mathbf{x})\mathbf{P}\exp(\mathbf{D}^T\mathbf{x})d\mathbf{x}. \tag{2.23}$$

2.4 Stability

Certain definitions of stability that will be used in the thesis are now given.

Definition 2.1 *An LTI transfer operator*

$$P(q^{-1}) = \sum_{i=0}^{\infty} p_i q^{-i}, \tag{2.24}$$

where p_i is the i^{th} impulse response coefficient, is said to be stable if

$$\sum_{i=0}^{\infty} |p_i| < \infty. \tag{2.25}$$

Definition 2.2 *A family of transfer operators*

$$P(\varsigma, q^{-1}) = \sum_{i=0}^{\infty} p_i(\varsigma) q^{-i}, \quad \varsigma \in \mathcal{A}, \tag{2.26}$$

is said to be uniformly stable if

$$|p_i(\varsigma)| \leq p_i, \quad \forall \varsigma \in \mathcal{A}, \quad \sum_{i=0}^{\infty} p_i < \infty. \tag{2.27}$$

Definition 2.3 *An LPV transfer operator*

$$P(\boldsymbol{\sigma}(t), q^{-1}) = \sum_{i=0}^{\infty} p_i(\boldsymbol{\sigma}(t)) q^{-i}, \quad \boldsymbol{\sigma}(t) \in \mathcal{A} \subset \mathbb{R}^{n_\sigma}, \tag{2.28}$$

is said to be LPV stable if

$$|p_i(\boldsymbol{\sigma}(t))| \leq p_i, \quad \textit{for } t = 0, 1, \ldots, \quad \sum_{i=0}^{\infty} p_i < \infty. \tag{2.29}$$

Remark: The form of stability in the above definitions is sometimes referred to as bounded-input, bounded-output stability.

2.5 Ergodicity

It is often useful to be able to equate (probabilistically) the infinite time average of a random process over a single realisation with its expected value i.e.

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} (x(t) - E\{x(t)\}) = 0 \quad \text{w.p. } 1 \tag{2.30}$$

or, in the case that $x(t)$ is stationary,

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} x(t) = E\{x(t)\} \quad \text{w.p. } 1. \tag{2.31}$$

When the above relationships hold the random process in question is said to be ergodic. Essentially, for a process to be ergodic it should satisfy the strong law of large numbers (SLLN).

The theorem below provides sufficient conditions for the SLLN to apply to a random sequence.

Theorem 2.3 *[p. 253 in [55]] Let $x(t)$ be an independent random sequence with constant mean μ_x and variance $\sigma_x^2(t)$ defined for $t \geq 0$. Define another random sequence as:*

$$\hat{\mu}_x(N) = \frac{1}{N} \sum_{t=0}^{N-1} x(t) \quad \textit{for } N \geq 1. \tag{2.32}$$

Then if

$$\lim_{N\to\infty} \sum_{t=0}^{N-1} \frac{\sigma_x^2(t)}{(t+1)^2} < \infty, \tag{2.33}$$

$$\hat{\mu}_x(N) \to \mu_x \textit{ as } N \to \infty \quad \textit{w.p. 1.} \tag{2.34}$$

In the more specific case where the random process results from the filtering of a sequence of zero-mean independent random variables by a stable linear time varying filter, the theorem below indicates when ergodicity in the correlation exists.

Theorem 2.4 *[p. 55 in [32]] Let* $\{P_\varsigma(q^{-1}), \varsigma \in \mathcal{A}\}$ *and* $\{M_\varsigma(q^{-1}), \varsigma \in \mathcal{A}\}$ *be uniformly stable families of filters, and assume that the deterministic signal* $\mathbf{w}(t)$, $t = 1, 2, \ldots$, *is subject to*

$$|\mathbf{w}(t)| \leq C_w, \quad \forall t. \tag{2.35}$$

Let the signal vector $\mathbf{s}_\varsigma(t)$ *be defined, for each* $\varsigma \in \mathcal{A}$, *by*

$$\mathbf{s}_\varsigma(t) = P_\varsigma(q^{-1})\mathbf{v}(t) + M_\varsigma(q^{-1})\mathbf{w}(t) \tag{2.36}$$

where

$$\mathbf{v}(t) = \sum_{i=0}^{\infty} h_i(t)\boldsymbol{\eta}(t-i) = H(t, q^{-1})\boldsymbol{\eta}(t) \tag{2.37}$$

and $\boldsymbol{\eta}(t)$ *is a sequence of independent random vectors with zero-mean values,* $E\{\boldsymbol{\eta}(t)\boldsymbol{\eta}^T(t)\} = \mathbf{R}_\eta(t)$ *and bounded fourth moments, and* $\{H(t, q^{-1}), t = 1, 2, \ldots\}$ *is a uniformly stable family of filters. Then:*

$$\sup_{\varsigma \in \mathcal{A}} \left\| \frac{1}{N} \sum_{t=0}^{N-1} \left[\mathbf{s}_\varsigma(t)\mathbf{s}_\varsigma^T(t) - E\{\mathbf{s}_\varsigma(t)\mathbf{s}_\varsigma^T(t)\} \right] \right\|_F \to 0$$
$$w.p.\ 1,\ as\ N \to \infty, \tag{2.38}$$

where $\|\cdot\|_F$ *is the Frobenius norm.*

The corollary below states when certain types of nonstationary signals, those generated by filtering a sequence of zero-mean independent random variables by a stable LPV system, can be ergodic in the correlation.

Corollary 2.1 *Assume that the deterministic signal* $\mathbf{w}(t)$, $t = 1, 2, \ldots$, *is subject to*

$$|\mathbf{w}(t)| \leq C_w, \quad \forall t. \tag{2.39}$$

Let $M_v(q^{-1})$ *be a stable filter, and the signal vector* $\mathbf{s}(t)$ *be defined by*

$$\mathbf{s}(t) = M_v(q^{-1})\mathbf{v}(t) + \mathbf{w}(t) \tag{2.40}$$

where

$$\mathbf{v}(t) = \sum_{i=0}^{\infty} h_i(\boldsymbol{\sigma}(t))\boldsymbol{\eta}(t-i) = H(\boldsymbol{\sigma}(t), q^{-1})\boldsymbol{\eta}(t) \tag{2.41}$$

and $\boldsymbol{\eta}(t)$ *is a sequence of independent random vectors with zero-mean values,* $E\{\boldsymbol{\eta}(t)\boldsymbol{\eta}^T(t)\} = \mathbf{R}_\eta(t)$ *and bounded fourth moments, and* $H(\boldsymbol{\sigma}(t), q^{-1}), t = 1, 2, \ldots$ *is an LPV stable filter. Then:*

$$\left\| \frac{1}{N} \sum_{t=0}^{N-1} \left[\mathbf{s}(t)\mathbf{s}^T(t) - E\{\mathbf{s}(t)\mathbf{s}^T(t)\} \right] \right\|_F \to 0 \quad \textit{w.p. 1, as } N \to \infty. \tag{2.42}$$

Proof: The proof follows immediately from Theorem 2.4. ■

2.6 Application

The majority of the methods developed in this thesis are tested in application on a linear, permanent magnet, synchronous motor (LPMSM), which forms the upper axis of an x-y positioning table, see Figure 2.3.

LPMSMs are now used abundantly in industry whenever precise, linear motion is required. They offer a number of advantages over the traditional solution for linear motion of a rotary motor and leadscrew, principally:

- No backlash
- Reduced translator inertia
- Reduced friction
- Increased mechanical stiffness

These properties mean that LPMSMs are capable of fast, high precision movements. Typical uses for LPMSMs include wafer stages,

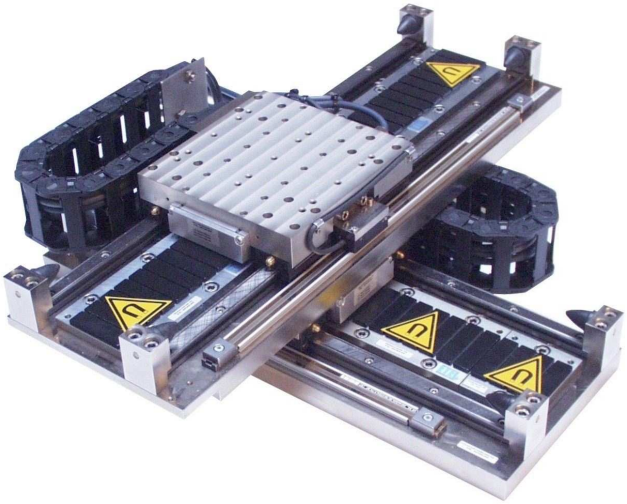

Fig. 2.3. The linear motor positioning system (courtesy of ETEL)

microscale robotic decomposition, electronic assembly and manufacturing, and machine tools. The precision required by all of these applications is, however, continuously increasing. These continuously increasing demands on the tracking performance can be met in two ways, either via improved hardware or via improved software. The hardware option includes techniques such as using better bearings to reduce friction and skewing the permanent magnets to reduce the cogging forces caused by the attraction between the magnets and iron core of the translator. These options, however, are expensive because they increase either the component or manufacturing cost, or both. The software option implies the use of more advanced control algorithms. This latter option can be expensive to develop but, once an algorithm is designed, it is much cheaper to implement in production than a hardware change, leading to long-term savings. LPMSMs are

therefore an obvious application for the new algorithms developed in this thesis.

The motor used in the experiments is manufactured by the Swiss company ETEL SA. It can use either a standard proportional-integral-derivative (PID) controller or a, more general, two-degree-of-freedom controller to control the motor's position. The chosen controller can operate at sampling frequencies $18/i$ kHz for $i = 1, 2, \ldots$ An analog position encoder using sinusoidal signals with periods of 2μm, which are then interpolated with 8192 intervals/period to obtain a resolution of 0.24nm, is used to measure the motor's position. However, the accuracy of this encoder is limited to 20nm.

Two movements of different amplitudes have been defined as benchmarks for typical industrial movements by the manufacturer. Both movements have the form of low-pass filtered steps, which, being less abrupt, is desirable in industry as does not excite unwanted vibrations and avoids input saturation. They are defined in terms of their amplitude, maximum velocity and maximum acceleration. The smaller movement, which will be referred to as a Micromotion, has an amplitude of 5μm, see Figure 2.4. The larger movement, the Macromotion, has an amplitude of 25 mm. Both movements have a maximum velocity of 0.5m/s and a maximum acceleration of 3 m/s^2.

As can be seen from the above movements, the LPMSM is expected to operate accurately over a large range of amplitudes. Often a 'dual-stage' approach using two actuators, for course and fine positioning, is adopted to deal with this wide operating range [15]. If a single actuator can be used to obtain the desired positioning then costs will normally be reduced. This further promotes the LPMSM as an interesting application for the methods developed in the thesis.

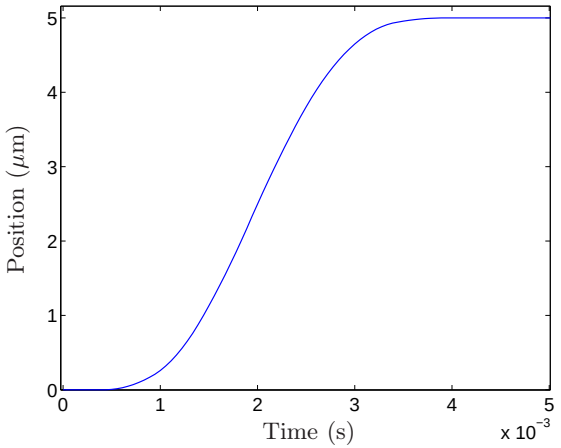

Fig. 2.4. The Micromotion

3

Data-driven precompensator tuning for LTI systems

3.1 Introduction

In this chapter, we consider the data-driven tuning of the parameters of precompensators for LTI systems. A precompensator filters the desired output signal before it is applied to a system's input. The correlation approach is used to tune the precompensator's parameters such that the correlation between the tracking error and the desired output is minimised. A new tuning scheme is proposed in which the positions of the precompensator and the system are swapped. This scheme means that the evaluation of the control criterion does not require multiple experiments on the system and only one set of data is sufficient for the tuning of the precompensator's parameters. This implies that, similarly to the Virtual Reference Feedback Tuning approach, parameter estimation algorithms can be used to directly 'identify' the controller parameters. However, in this tuning scheme the stochastic disturbances affect the input of the precompensator, which makes the problem more difficult than classical identification problems. This problem is known as errors-in-variables in the literature [54]. Here, it is shown that the use of extended instrumental variables with a specific choice of instruments leads to consistent parameter estimates. Moreover, a frequency-domain analysis of the

criterion shows that, even when the ideal controller does not belong to the set of parameterised controllers, the two-norm of the difference between the reference model and the system is minimised.

The chapter is organised as follows. In Section 3.2 the precompensator tuning problem is formulated. The precompensator tuning method is presented in Section 3.3. Simulation and experimental results are presented in Sections 3.4 and 3.5, respectively. Finally, the chapter conclusions are given in Section 3.6.

3.2 Problem formulation

The controlled output $z(t)$ of the system when a precompensator $F(q^{-1})$ is used is given by, see Figure 3.1:

$$z(t) = G(q^{-1})F(q^{-1})y_d(t) + d(t). \tag{3.1}$$

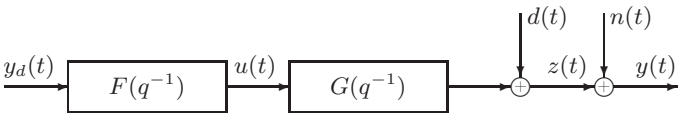

Fig. 3.1. System with a precompensator

3.2.1 Assumptions

A3.1: The system $G(q^{-1})$ is stable.

A3.2: The disturbances $d(t)$ and $n(t)$ are zero-mean, weakly stationary random processes with bounded, unknown variances, fourth moments and cross-correlation terms.

A3.3: The desired output $y_d(t)$ is bounded and uncorrelated with the disturbances.

3.2.2 Ideal precompensator

The objective is to calculate the precompensator parameters so as to reduce the controlled tracking error $\epsilon(t)$. We define the ideal precompensator as that which gives a zero-mean controlled tracking error i.e.

$$E\{\epsilon(t)\} = 0 \quad \forall t. \tag{3.2}$$

Under the assumptions of Section 3.2.1, it is clear that the ideal precompensator $F(q^{-1})$ is equal to the inverse of the system's transfer function.

3.2.3 Precompensator parameterisation

Let $F(q^{-1})$ be parameterised as:

$$F(\boldsymbol{\rho}, q^{-1}) = \boldsymbol{\beta}^T(q^{-1})\boldsymbol{\rho}, \tag{3.3}$$

where $\boldsymbol{\rho}^T = [\rho_0, \rho_1, \ldots, \rho_{n_\rho}]$ is the vector of controller parameters and $\boldsymbol{\beta}(q^{-1})$ is the vector of linear discrete-time transfer operators:

$$\boldsymbol{\beta}^T(q^{-1}) = [\beta_0(q^{-1}), \beta_1(q^{-1}), \ldots, \beta_{n_\rho}(q^{-1})]. \tag{3.4}$$

The elements of the vector $\boldsymbol{\beta}(q^{-1})$ can be any orthogonal basis functions, such as Laguerre or Kautz. In the sequel, for simplicity of presentation, we suppose that $\boldsymbol{\beta}^T(q^{-1}) = [q^{\delta}, q^{\delta-1}, \ldots, q^{\delta-n_\rho}]$ which leads to the following FIR model for $F(\boldsymbol{\rho}, q^{-1})$:

$$F(\boldsymbol{\rho}, q^{-1}) = \rho_0 q^{\delta} + \rho_1 q^{\delta-1} + \cdots + \rho_{n_\rho} q^{\delta-n_\rho}, \tag{3.5}$$

where δ is a positive scalar. In fact, the desired output is applied δ sampling periods in advance to the real system to improve the tracking error [25].

3.3 Tuning method

3.3.1 Correlation approach

It is evident that if the exact inverse of the system exists the measured tracking error $e(t)$ will contain only contributions from the stochastic disturbances. Hence, it is reasonable to adjust the controller $F(\boldsymbol{\rho}, q^{-1})$ in such a way that the measured error $e(t)$ be uncorrelated with the desired output. For many systems, the exact inverse does not exist because the system is non-minimum phase or of infinite order. As a result, $e(t)$ is always correlated with the desired output. Nevertheless, it can be considered that a good precompensator $F(\boldsymbol{\rho}, q^{-1})$ minimises the correlation between the measured error $e(t)$ and the desired output $y_d(t)$. In order to formulate this idea as an optimisation problem, let the correlation function vector $\mathbf{f}(\boldsymbol{\rho})$ be defined as:

$$\mathbf{f}(\boldsymbol{\rho}) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} E\{\boldsymbol{\zeta}(t)e(t)\}, \tag{3.6}$$

where

$$\boldsymbol{\zeta}^T(t) = [y_d(t+l), \ldots, y_d(t+1), y_d(t), y_d(t-1), \ldots, y_d(t-l)] \tag{3.7}$$

with $\boldsymbol{\zeta}(t) \in \mathbb{R}^{n_\zeta}$ where $n_\zeta = 2l+1$. In general, $\boldsymbol{\zeta}(t)$ is a vector of instrumental variables correlated with $y_d(t)$ and uncorrelated with $d(t)$ and $n(t)$. Now, a new control criterion based on the correlation approach is defined as:

$$J_{CA}(\boldsymbol{\rho}) = \|\mathbf{f}(\boldsymbol{\rho})\|_2^2 = \mathbf{f}^T(\boldsymbol{\rho})\mathbf{f}(\boldsymbol{\rho}) \tag{3.8}$$

and the optimal controller parameters are:

$$\boldsymbol{\rho}^* = \arg\min_{\boldsymbol{\rho}} J_{CA}(\boldsymbol{\rho}). \tag{3.9}$$

Since the control criterion involves the mathematical expectation, an exact solution, when only one finite set of data is available, is

not attainable. However, under the assumptions of Subsection 3.2.1, Theorem 2.4 applies meaning the signals are ergodic in the correlation so a good estimate of the correlation function can be given by:

$$\hat{\mathbf{f}}(\boldsymbol{\rho}) = \frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\zeta}(t)e(t), \tag{3.10}$$

where N should be large enough with respect to l. The estimate of the correlation function leads to the following criterion:

$$J_{CA}^{N}(\boldsymbol{\rho}) = \|\hat{\mathbf{f}}(\boldsymbol{\rho})\|_2^2 = \hat{\mathbf{f}}^T(\boldsymbol{\rho})\hat{\mathbf{f}}(\boldsymbol{\rho}). \tag{3.11}$$

The criterion $J_{CA}^{N}(\boldsymbol{\rho})$ converges almost surely to $J_{CA}(\boldsymbol{\rho})$ when N tends to infinity.

Next a tuning scheme to estimate the parameters of the precompensator $F(\boldsymbol{\rho}, q^{-1})$ is proposed.

3.3.2 Tuning scheme

The measured tracking error can be computed as (see Figure 3.1):

$$\begin{aligned} e(t) &= y_d(t) - y(t) \\ &= y_d(t) - G(q^{-1})F(\boldsymbol{\rho}, q^{-1})y_d(t) - d(t) - n(t). \end{aligned} \tag{3.12}$$

Computing $e(t)$ for different values of $\boldsymbol{\rho}$ requires many experiments on the system that can be avoided by a new tuning scheme in which the positions of the system and precompensator are interchanged so that $F(\boldsymbol{\rho}, q^{-1})$ acts as a post-compensator (see Figure 3.2). In this scheme $y_m(t)$ is the measured output of the system from an experiment where the desired output $y_d(t)$ is applied directly as the input signal $u(t)$. An estimate of the measured tracking error can now be computed with only one set of data as follows:

$$\begin{aligned} \hat{e}(t) &= y_d(t) - \hat{y}(t) = y_d(t) - F(\boldsymbol{\rho}, q^{-1})y_m(t) \\ &= y_d(t) - F(\boldsymbol{\rho}, q^{-1})G(q^{-1})y_d(t) - F(\boldsymbol{\rho}, q^{-1})(d(t) + n(t)). \end{aligned} \tag{3.13}$$

It is clear that in the absence of disturbances ($d(t) = n(t) = 0$) $e(t)$ and $\hat{e}(t)$ are equal. However, even in their presence, we have:

$$\mathbf{f}(\boldsymbol{\rho}) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} E\{\boldsymbol{\zeta}(t)e(t)\} = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} E\{\boldsymbol{\zeta}(t)\hat{e}(t)\}. \tag{3.14}$$

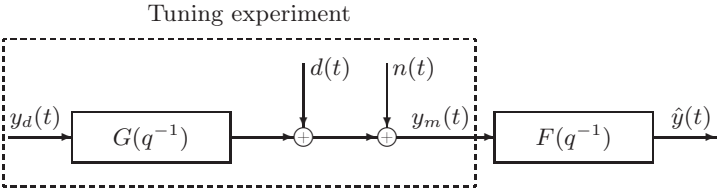

Fig. 3.2. Precompensator tuning scheme

Remarks:

1. The equality (3.14) holds true because $d(t)+n(t)$ is uncorrelated with $y_d(t)$ and zero mean. In the case that $G(q^{-1})$ represents a closed-loop system and the external disturbances affecting the system are not zero mean, then $d(t)+n(t)$ can still be zero mean as it is the product of filtering the sum of the external disturbances by a closed-loop transfer function, see (2.3)–(2.5). This closed-loop transfer function can be designed to reject non-zero mean disturbances, such as steps and ramps, using the internal model principle.
2. It should be mentioned that the variance of the measured tracking error $E\{e^2(t)\}$ is not equal to the variance of the measured tracking error estimate $E\{\hat{e}^2(t)\}$. Therefore, the minimisation of the variance of the measured tracking error cannot be carried out with only one experiment and should be done iteratively with several experiments on the real system.

The estimate of the measured tracking error $\hat{e}(t)$ can be presented in linear regression form as:

$$\hat{e}(t) = y_d(t) - \boldsymbol{\phi}^T(t)\boldsymbol{\rho}, \tag{3.15}$$

where

$$\boldsymbol{\phi}^T(t) = [y_m(t+\delta), y_m(t+\delta-1), \ldots, y_m(t+\delta-n_\rho)]. \tag{3.16}$$

This representation leads to the following expression for the correlation function estimate:

$$\hat{\mathbf{f}}(\boldsymbol{\rho}) = \frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\zeta}(t)[y_d(t) - \boldsymbol{\phi}^T(t)\boldsymbol{\rho}] = \mathbf{r} - \mathbf{Q}_y\boldsymbol{\rho} \tag{3.17}$$

where

$$\mathbf{r} = \frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\zeta}(t)y_d(t) \quad \text{and} \quad \mathbf{Q}_y = \frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\zeta}(t)\boldsymbol{\phi}^T(t). \tag{3.18}$$

Finally, if $\mathbf{Q}_y^T\mathbf{Q}_y$ is nonsingular (i.e. $y_d(t)$ is sufficiently rich), straightforward calculation gives:

$$\boldsymbol{\rho}^N = (\mathbf{Q}_y^T\mathbf{Q}_y)^{-1}\mathbf{Q}_y^T\mathbf{r}, \tag{3.19}$$

where $\boldsymbol{\rho}^N$ is the global minimiser of the criterion in (3.11).

3.3.3 Frequency-domain analysis

The correlation criterion in (3.8) can be reformulated as:

$$J_{CA}(\boldsymbol{\rho}) = \mathbf{f}^T(\boldsymbol{\rho})\mathbf{f}(\boldsymbol{\rho}) = \sum_{\tau=-l}^{l} r_{ey_d}^2(\tau) \tag{3.20}$$

where $r_{ey_d}(\tau)$ is the cross-correlation function between the measured tracking error $e(t)$ and the desired output $y_d(t)$, defined by:

$$r_{ey_d}(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} E\{e(t)y_d(t-\tau)\}$$
$$= \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} E\left\{[y_d(t) - F(\boldsymbol{\rho}, q^{-1})G(q^{-1})y_d(t)]y_d(t-\tau)\right\}. \tag{3.21}$$

The correlation criterion can be represented in the frequency domain by applying Parseval's theorem when l tends to infinity:

$$\lim_{l\to\infty} J_{CA}(\boldsymbol{\rho}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} |\Phi_{ey_d}(\omega)|^2 d\omega$$
$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} |1 - F(\boldsymbol{\rho}, e^{-j\omega})G(e^{-j\omega})|^2 \Phi_{y_d}^2(\omega) d\omega, \tag{3.22}$$

where $\Phi_{ey_d}(\omega)$ is the cross-spectral density between $e(t)$ and $y_d(t)$ and $\Phi_{y_d}(\omega)$ is the spectral density of $y_d(t)$. This expression shows that:

- The criterion is asymptotically unaffected by the disturbances.
- In the ideal case (i.e. $G(q^{-1})$ is minimum phase and $F(\boldsymbol{\rho}, q^{-1})$ is properly parameterised), the minimum of the correlation criterion in (3.8) is such that its minimiser $\boldsymbol{\rho}^* = \boldsymbol{\rho}_0$, where $\boldsymbol{\rho}_0$ are the parameters of the ideal precompensator i.e. $F(\boldsymbol{\rho}_0, q^{-1}) = G^{-1}(q^{-1})$.
- If $y_d(t)$ is white noise with unit variance, the correlation criterion becomes:

  $$J_{CA}(\boldsymbol{\rho}) = \|1 - F(\boldsymbol{\rho}, e^{-j\omega})G(e^{-j\omega})\|_2^2$$
  $$= \|E\{\epsilon(t)\}\|_2^2.$$

  So, using the correlation approach, the difference between $F(\boldsymbol{\rho}, q^{-1})G(q^{-1})$ and 1, or alternatively the mean controlled error, is minimised in the two-norm sense.
- If $y_d(t)$ is a deterministic signal, $|1 - F(\boldsymbol{\rho}, e^{-j\omega})G(e^{-j\omega})|$ is minimised at the frequencies where $y_d(t)$'s spectrum is large.

Remarks:

1. The above frequency-domain analysis rationalises the choice of criterion (3.8) and instrumental variable vector (3.7) used in this chapter. With these choices it is possible to obtain the above frequency-domain interpretation of (3.8). This interpretation shows that even when the exact system inverse does not exist in the parameterised set of precompensators, which will nearly always be the case in practice, a logical criterion is still minimised. It should be mentioned that other choices of the instrumental variable vector may be more efficient when the ideal precompensator exists in the parameterised set of precompensators, however they will give undesirable frequency weightings when the ideal precompensator is not achievable.
2. The general model following problem in the two-norm can also be treated with this data-driven approach. Consider that we aim to compute the precompensator $F(\boldsymbol{\rho}, q^{-1})$ such that

$$\|M(e^{-j\omega}) - F(\boldsymbol{\rho}, e^{-j\omega})G(e^{-j\omega})\|_2^2$$

   is minimised. To proceed, let us define

$$\hat{e}_M(t) = M(q^{-1})y_d(t) - \boldsymbol{\phi}^T(t)\boldsymbol{\rho}$$

   and compute $\boldsymbol{\rho}$ such that $\hat{e}_M(t)$ is not correlated with $y_d(t)$, which is chosen to be a white noise signal that is uncorrelated with $d(t)$ and $n(t)$. If, for practical reasons, $y_d(t)$ cannot be chosen as a white noise but can be expressed as $y_d(t) = D(q^{-1})\eta(t)$ where $\eta(t)$ is a white noise, the use of the filtered error $D^{-1}(q^{-1})\hat{e}_M(t)$ and filtered instrumental variable $D^{-1}(q^{-1})\boldsymbol{\zeta}(t)$ leads to the minimisation of $\|M(e^{-j\omega}) - F(\boldsymbol{\rho}, e^{-j\omega})G(e^{-j\omega})\|_2^2$.
3. If instead of the correlation criterion in (3.11) the variance of $\hat{e}(t)$ is minimised, unacceptable results may be obtained even if the noise to signal ratio is not very high. The reason for these poor results can be seen from the frequency expression of the variance of $\hat{e}(t)$:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} E\{\hat{e}^2(t)\}$$
$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} \big[|1 - F(\boldsymbol{\rho}, e^{-j\omega})G(e^{-j\omega})|^2 \Phi_{y_d}(\omega)$$
$$+ |F(\boldsymbol{\rho}, e^{-j\omega})|^2 \left(\Phi_d(\omega) + \Phi_n(\omega) + 2\Phi_{nd}(\omega)\right)\big] d\omega, \quad (3.23)$$

where $\Phi_d(\omega)$ and $\Phi_n(\omega)$ are the auto-spectral density of $d(t)$ and $n(t)$, respectively, and $\Phi_{nd}(\omega)$ is the cross-spectral density of $d(t)$ and $n(t)$. It is clear that, when $\Phi_{nd}(\omega) = 0$, the variance of the measured tracking error estimate is minimised when the three positive terms in the integral are minimised. However, minimising the first term requires that $F(\boldsymbol{\rho}, q^{-1})$ be close to a high-pass filter (as $G(q^{-1})$ is usually a low-pass filter) which consequently increases the effect of high-frequency disturbances in the second and third terms of the integral.

4. Although, theoretically l in (3.22) should go to infinity in order to obtain the frequency interpretation of the criterion, in practice a large value for l is sufficient. The reason is that $r_{ey_d}(\tau)$ is close to zero for τ greater than the settling time of the impulse response of the transfer function between $y_d(t)$ and $e(t)$ when $y_d(t)$ is white noise. This gives a guideline to choose the value of l. Additionally the number of data N should be chosen much greater than l (e.g. $N > 10l$).

3.3.4 Controller structure selection

The controller structure given in (3.5) has only two parameters δ and n_ρ to be chosen. Here a simple algorithm to select these parameters is presented. It is clear that for a given value of δ, increasing n_ρ will reduce the correlation criterion in (3.8). However, it is not reasonable to continue increasing the controller order when the design objective (decorrelation of the measured tracking error and the desired output) is already achieved. It can be shown that if $J_{CA}^N(\boldsymbol{\rho})$ is within a

confidence interval the measured tracking error and the desired output can be considered uncorrelated. This confidence interval can be computed using the fact that, when the measured tracking error and the desired output are uncorrelated, the random variable $\sqrt{N}\hat{r}_{\hat{e}y_d}(\tau)$ converges in distribution to a normal distribution when N goes to infinity [32]:

$$\sqrt{N}\hat{r}_{\hat{e}y_d}(\tau) = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} \hat{e}(t) y_d(t-\tau) \rightarrow \mathcal{N}(0, C) \tag{3.24}$$

where

$$C = \sum_{\tau=-\infty}^{\infty} r_{\hat{e}}(\tau) r_{y_d}(\tau) \tag{3.25}$$

with $r_{\hat{e}}(\tau)$ and $r_{y_d}(\tau)$ being the autocorrelation functions of $\hat{e}(t)$ and $y_d(t)$, respectively. Thus, from the criterion (3.11), it follows that:

$$\lim_{N\rightarrow\infty} \frac{N}{C} J_{CA}^N(\boldsymbol{\rho}_0) \rightarrow \chi^2(n_\zeta) \tag{3.26}$$

where $\boldsymbol{\rho}_0$ is the parameter vector that corresponds to the ideal precompensator and therefore that achieves decorrelation. We denote the α-level of the $\chi^2(n_\zeta)$ distribution as $\chi^2_\alpha(n_\zeta)$, i.e. $\alpha = P(x > \chi^2_\alpha(n_\zeta))$ where $P(A)$ is the probability of event A and x is a random variable with a $\chi^2(n_\zeta)$ distribution and n_ζ degrees of freedom. The condition to be satisfied in selecting the controller order is then:

$$J_{CA}^N(\boldsymbol{\rho}) \leq \frac{\hat{C}}{N} \chi^2_\alpha(n_\zeta) \tag{3.27}$$

where $\hat{C}$ is an estimate of C based on the calculated parameter vector $\boldsymbol{\rho}$. This condition allows an algorithm to be proposed for the selection of the values of the parameters δ and n_ρ:

Algorithm: $n_\rho = 1$

I : $\delta^* = \arg\min_\delta J_{CA}^N(\boldsymbol{\rho}, n_\rho, \delta)$ for $\delta = 0 : \delta_{max}$

if $J_{CA}^N(\boldsymbol{\rho}, n_\rho, \delta^*) \leq \frac{\hat{C}}{N}\chi^2_\alpha(n_\zeta)$

stop; $n^*_\rho = n_\rho$ and $\boldsymbol{\rho} = \boldsymbol{\rho}_0$

else $n_\rho = n_\rho + 1$ and go to I

3.4 Simulation results

The proposed method is tested in simulation.

Example 1

The method is applied to a system represented by the following transfer function:

$$G(q^{-1}) = \frac{0.2q^{-4}}{1 - 1.5q^{-1} + 0.7q^{-2}}. \tag{3.28}$$

This simple model is used in this example as its exact inverse exists in the set of parameterised precompensators. The ideal precompensator $F(\boldsymbol{\rho}_0, q^{-1}) = G^{-1}(q^{-1})$, where

$$\boldsymbol{\rho}_0 = [5, -7.5, 3.5]^T, \tag{3.29}$$

should therefore be found by the proposed method under ideal conditions.

The desired output is the response of the discrete-time second-order system:

$$\frac{0.0941q^{-1} + 0.0708q^{-2}}{1 - 1.262q^{-1} + 0.4274q^{-2}} \tag{3.30}$$

to a square-wave signal (between -1 and 1) of ten periods ($N = 2000$). 200 simulations are carried out in order to assess the mean performance of the method. In each simulation the desired output $y_d(t)$ is applied to the system without a precompensator to obtain the simulated measured output as:

$$y_m(t) = G(q^{-1})y_d(t) + d(t) + n(t) \tag{3.31}$$

where the disturbances $d(t)$ and $n(t)$ are signals resulting from filtering independent realisations of a normally distributed zero-mean random sequence with a variance of 0.05^2 by the transfer function $1 - G(q^{-1})$. They both have average variances of 0.0582^2, where the average variances are calculated as

$$\frac{1}{N}\sum_{t=0}^{N-1}\hat{E}\{v^2(t)\}$$

and the estimated expected value $\hat{E}$ is evaluated using the ensemble average over the 200 simulations i.e.

$$\hat{E}\{v^2(t)\} = \frac{1}{N_s}\sum_{j=1}^{N_s} v_j^2(t) \tag{3.32}$$

where $v_j(t)$ represents a random variable at simulation j, and N_s is the number of simulations.

Figure 3.3 shows one period of the desired output $y_d(t)$ and a realisation of the measured output $y_m(t)$. The precompensator tuning

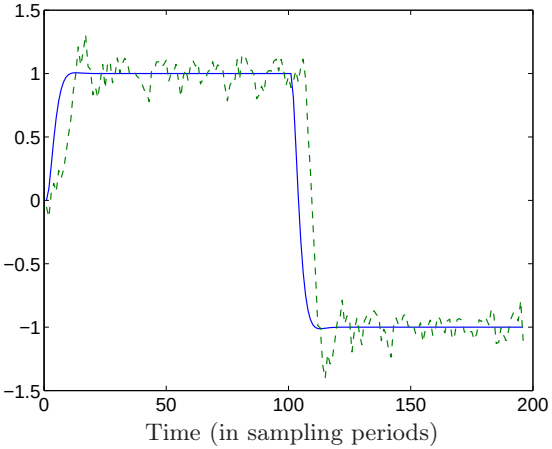

Fig. 3.3. Desired output (solid) and a realisation of the measured output $y_m(t)$ (dashed) for Example 1

algorithm in Eqs.(3.17)-(3.19), together with the controller structure

selection algorithm, are used to calculate the precompensator. A value of $l = 25$ is chosen based on the estimated settling time of $y_m(t)$. Figure 3.4 shows the value of $J_{CA}^N(\boldsymbol{\rho}, n_\rho, \delta^*)$ for different values of n_ρ for one of the simulations. Additionally the corresponding values of $\frac{\hat{C}}{N}\chi_\alpha^2(n_\zeta)$ for $\alpha = 0.05$ and $n_\zeta = 2l + 1 = 51$ are shown. It is clearly seen that the condition (3.27) is satisfied for $n_\rho \geq 2$, thus $n_\rho = 2$ was chosen. Additionally 4 sampling periods of preview ($\delta = 4$) gives the minimum value of the correlation criterion for this order. These values are found for each of the 200 simulations, despite the different noise realisations. Figure 3.5 compares the disturbance-

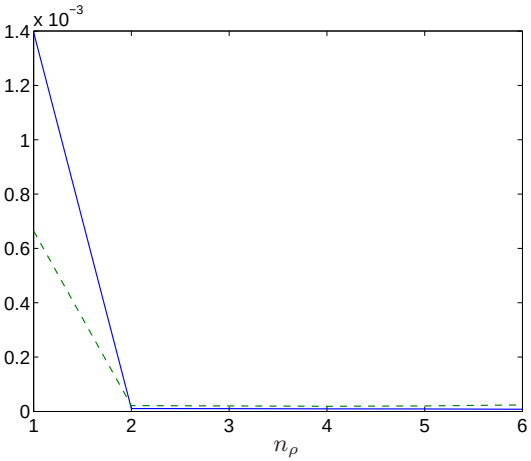

Fig. 3.4. Correlation criterion (solid) and controller order selection criterion (dashed) versus n_ρ for Example 1

free output of the original system without a precompensator with the disturbance-free output of the system using the precompensator computed in one of the simulations. The disturbances are not present in this validation-type simulation so that the true tracking obtained

using a precompensator, tuned in the presence of disturbances, is clearly visible. A precompensator whose parameters are tuned to minimise the variance of $\hat{e}(t)$ is also computed each simulation. The disturbance-free output of the system with a precompensator calculated this way is also shown in Figure 3.5. Table 3.1 compares the

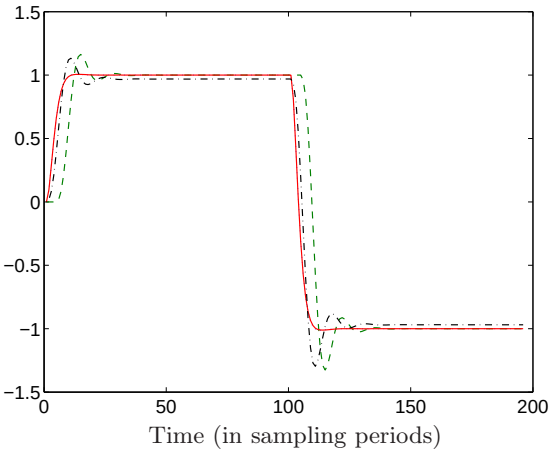

Fig. 3.5. Desired output (dotted), tracking obtained in one of the validation simulations for Example 1 for the system: without precompensator (dashed), with precompensator tuned using the correlation approach (solid/superimposed on desired output) and with precompensator tuned to minimise the variance of $\hat{e}(t)$ (dash-dot).

mean square values of the parametric error $\boldsymbol{\rho}_0 - \boldsymbol{\rho}^N$ obtained by the two parameter tuning methods. Additionally the mean square tracking error obtained in the validation simulation with the different precompensators is presented. It can be observed that the proposed method estimates the ideal precompensator parameters very well and produces almost perfect tracking in the validation experiment. The presence of the stochastic disturbances in $y_m(t)$ causes

biased precompensator parameters to be estimated by the minimisation of the variance of $\hat{e}(t)$, however, leading to greatly reduced tracking improvement. This reduced performance is further emphasised in Figure 3.6, which compares the mean square tracking error values obtained using the different precompensators calculated at each simulation. Considerably smaller mean square tracking error values are achieved for all of the simulations using the precompensators tuned by the correlation approach.

System	$\hat{E}\{(\boldsymbol{\rho}_0 - \boldsymbol{\rho}^N)^T(\boldsymbol{\rho}_0 - \boldsymbol{\rho}^N)\}$	$\hat{E}\left\{\frac{1}{N}\sum_{t=0}^{N-1} e^2(t)\right\}$
i)	–	0.1201
ii)	0.000012	0.0000089
iii)	0.4351	0.0121

Table 3.1. Results for Example 1 for the system: i) without precompensator, ii) with precompensator tuned using the correlation approach and iii) with precompensator tuned to minimise the variance of $\hat{e}(t)$.

Example 2

A second simulation example is carried out with a different system:

$$G(q^{-1}) = \frac{-0.2q^{-3} + 0.4q^{-4}}{1 - 1.5q^{-1} + 0.7q^{-2}}. \tag{3.33}$$

This $G(q^{-1})$ is chosen to represent a more complicated system whose exact inverse is unstable and does not exist in the set of parameterised precompensators. 200 simulations are carried out again using the same $y_d(t)$ as in Example 1. The disturbance signals were generated similarly to Example 1 i.e. normally distributed zero-mean random sequences being filtered by $1 - G(q^{-1})$. This time they have

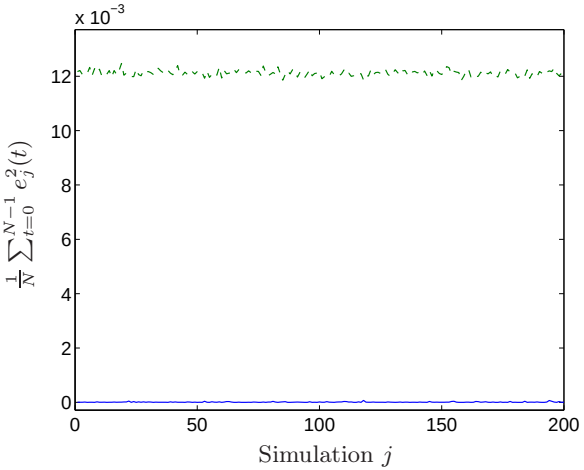

Fig. 3.6. Mean square values obtained in the validation simulations for Example 1 for the system: with precompensator tuned using the correlation approach (solid) and with precompensator tuned to minimise the variance of $\hat{e}(t)$ (dashed).

average variances of 0.0616^2. $l = 25$ is also used. In each simulation the controller structure selection algorithm is utilised. This time, however, since the exact system inverse does not exist in the precompensator parameterisation and a finite number of data are used, the controller structure selection algorithm does not find the same structure in each iteration. Nonetheless, disturbance-free validation simulations are carried out to test the precompensators tuned in the presence of disturbances and, as can be seen from Figure 3.7 and Table 3.2, the tracking is still greatly improved. Moreover, Figure 3.8 shows that smaller mean square error values are obtained than those achieved using precompensators tuned to minimise the variance of $\hat{e}(t)$ in all simulations.

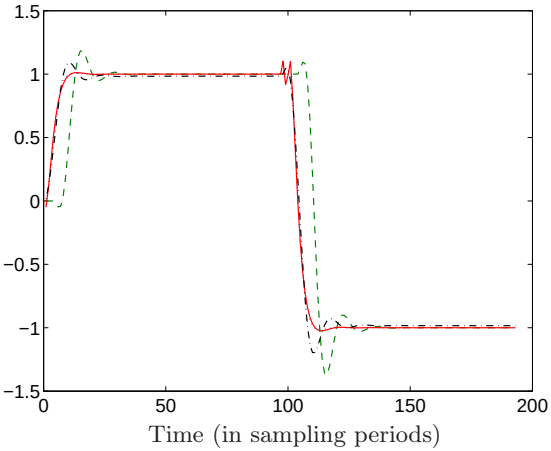

Fig. 3.7. Desired output (dotted), tracking obtained in one of the validation simulations for Example 2 for the system: without precompensator (dashed), with precompensator tuned using the correlation approach (solid/partially superimposed on desired output) and with precompensator tuned to minimise the variance of $\hat{e}(t)$ (dash-dot).

System	$\hat{E}\left\{\frac{1}{N}\sum_{t=0}^{N-1} e^2(t)\right\}$
Without precompensator	0.1689
With precompensator tuned with the correlation approach	0.00080
With precompensator tuned to minimise the variance of $\hat{e}(t)$	0.0064

Table 3.2. Results for Example 2

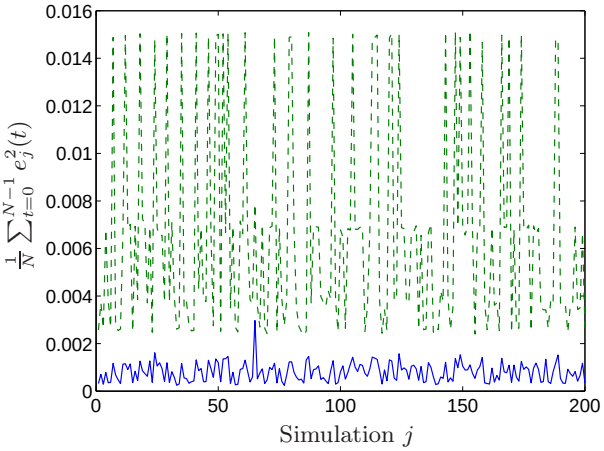

Fig. 3.8. Mean square values obtained in the validation simulations for Example 2 for the system: with precompensator tuned using the correlation approach (solid) and with precompensator tuned to minimise the variance of $\hat{e}(t)$ (dashed).

3.5 Experimental results

The proposed precompensator tuning method is applied to the LPMSM described in Chapter 2. As mentioned there, LPMSM are used extensively in industry for high precision positioning applications. The traditional model-based route for precompensator tuning is very labour intensive as it requires an extremely precise model to be identified in order to obtain the precision needed by these applications, if it is obtainable at all. Additionally, it requires that this identification process be repeated each time the system's parameters change, which happens regularly in an industrial setting due to wear. The method proposed in this chapter is thus highly suited to industrial applications as it provides a fast and efficient way to tune the precompensator parameters.

In the experiment the LPMSM uses the two-degree-of-freedom position controller, which contains an integrator. It operates at a sampling frequency of 6kHz, chosen to ensure good conditioning of the matrix $\mathbf{Q}_y^T\mathbf{Q}_y$. The controller was tuned previously in order to achieve robust stability using the method in [26]. It should be noted that in this experiment $G(q^{-1})$ represents the closed-loop plant and $u(t)$ its reference signal.

The desired output is taken as the Micromotion. At the chosen sampling frequency, the duration of Micromotion is such that $N = 1200$.

$y_d(t)$ is applied as the reference signal to the closed-loop system, without a precompensator, to obtain $y_m(t)$. The precompensator order and preview value are selected using the controller structure selection algorithm. A value of $l = 110$ is used, this value being estimated by measuring the settling time of the error signal of the system, without a precompensator, when an impulse is applied as a reference signal. The controller structure selection algorithm selects a precompensator with an order of $n_\rho = 3$ and a preview of $\delta = 4$. It can be seen from Figure 3.9 that this order is the first to satisfy condition (3.27). The calculated precompensator is applied to the system and the resulting system's output can be seen in Figure 3.10. In a similar way to the simulations a precompensator, with the same structure, is also calculated in order to minimise the variance of $\hat{e}(t)$. The system's response using this precompensator is also shown in Figure 3.10. As measures of performance the root mean square (RMS) of the measured error signal, the maximum overshoot and the settling time to within 2% of the final value are used. Table 3.3 shows the results obtained without and with the precompensator. It is clearly seen that the proposed technique greatly improves the system's tracking performance compared to the original performance. Compared to the performance obtained using the precompensator tuned to minimise the variance of $\hat{e}(t)$, the proposed technique can be seen, from the RMS values, to give better general tracking. The benefit of the proposed technique, however, is not as obvious as in

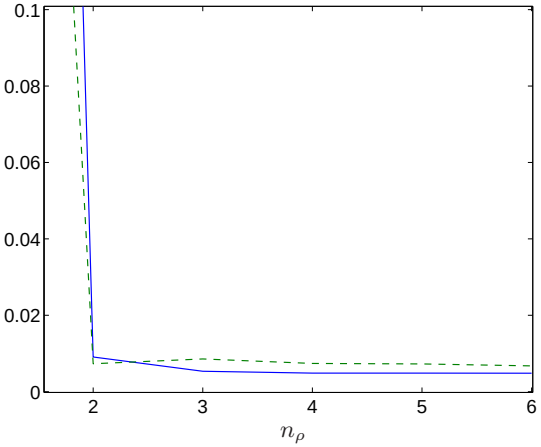

Fig. 3.9. Correlation criterion (solid) and controller order selection criterion (dashed) versus n_ρ for the experimental results

the simulations because the signal-to-noise ratio in this application is much higher.

System	$\sqrt{\frac{1}{N}\sum_{t=0}^{N-1} e^2(t)}$ (μm)	Overshoot (μm)	Settling time (s)
Without precomp.	0.5128	1.1367	0.0188
With precomp. tuned with the correlation approach	0.0490	0.0212	0.0162
With precomp. tuned to minimise the variance of $\hat{e}(t)$	0.0763	0.0171	0.0163

Table 3.3. System tracking performance

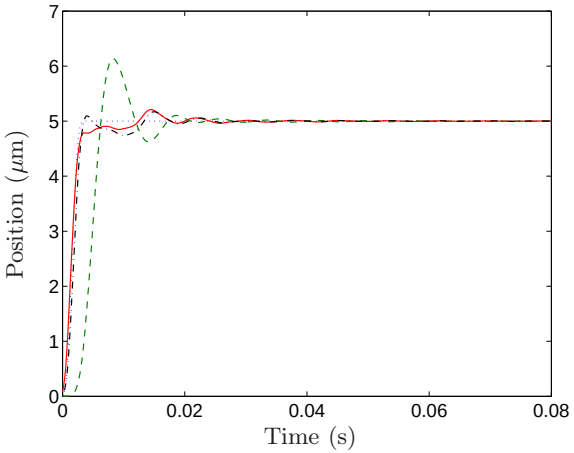

Fig. 3.10. Desired output (dotted), output without precompensator (dashed), output with precompensator tuned using the correlation approach (solid) and output with precompensator tuned to minimise the variance of $\hat{e}(t)$ (dash-dot) for the experimental results

3.6 Conclusions

A model-free approach to precompensator tuning based on the correlation approach has been proposed. It is shown that, using only one set of data and a specific tuning scheme, the controller parameters can be tuned for desired output tracking or the general model following problem. The approach is based on a correlation criterion which is asymptotically insensitive to noise and can be minimised using the linear least squares algorithm. The effectiveness of the method has been illustrated via simulation and experimental results.

The traditional approach to precompensator design involves two steps. Firstly a model of the system is identified. Secondly the precompensator is taken as the stable inverse of the identified model.

The approach proposed in this chapter has several advantages over the standard one:

- Only one-step is required.
- The inverse of an identified model may be unstable even if the system is minimum phase. This problem is avoided in the proposed method as the precompensator is parameterised as a finite impulse response, which is always stable.
- The structure of the precompensator (order, preview and basis functions $\beta_i(q^{-1})$) can be directly specified by the user, rather than being dictated by the identified model.
- In the popular prediction error identification methods the difference between $G(q^{-1})$ and the identified model is weighted by the inverse of the noise model. This weighting is not necessarily appropriate for control. If a method without a noise model, such as the output error method, is used,

$$\begin{aligned} &\|G(e^{-j\omega}) - F^{-1}(e^{-j\omega})\|_2^2 \\ &\qquad = \||F^{-1}(e^{-j\omega})||F(e^{-j\omega})G(e^{-j\omega}) - 1|\|_2^2 \end{aligned} \tag{3.34}$$

  is minimised so the control objective, which is the minimisation of the two norm of $F(q^{-1})G(q^{-1}) - 1$, is weighted by the inverse of the precompensator. This weighting is not necessarily desirable and cannot be compensated for by data pre-filtering because $F(q^{-1})$ is not known a priori and the use of iterative (bootstrap) methods does not guarantee convergence to the optimal solution.

Appendix

3.A Input weighting

When a precompensator is used to improve the tracking performance, it is possible that an input signal, most notably the plant input, becomes too large and saturates for certain desired outputs. Therefore, it is reasonable to take this signal into account in the design of the precompensator. In general, the input signal that is at risk of saturating, and therefore in need of reducing, can be described as:

$$u_w(t) = U(q^{-1})u(t) + H_u(q^{-1})(d(t) + n(t)). \tag{3.35}$$

It should be noted that in the case where $G(q^{-1})$ describes an open-loop, stable system, it is normally the signal $u(t)$ that we wish to influence and so $U(q^{-1}) = 1$ and $H_u(q^{-1}) = 0$. In the case that $G(q^{-1})$ represents a closed-loop system $u_w(t)$ may be an internal input, such as the plant input, so $U(q^{-1})$ will have another transfer function and $H_u(q^{-1})$ will be non-zero.

We denote as $u_{w,m}(t)$ the input $u_w(t)$ measured when $u(t) = y_d(t)$ is applied in the tuning experiment. An estimate of the input $u_w(t)$ that will occur when the precompensator is used can be obtained by:

$$\hat{u}_w(t) = F(\boldsymbol{\rho}, q^{-1})u_{w,m}(t) = \boldsymbol{\varphi}^T(t)\boldsymbol{\rho} \tag{3.36}$$

where

$$\boldsymbol{\varphi}^T(t) = [u_{w,m}(t+\delta), u_{w,m}(t+\delta-1), \ldots, u_{w,m}(t+\delta-n_\rho)]. \tag{3.37}$$

It is clear that in the absence of noise, $\hat{u}_w(t)$ is equal to $u_w(t)$ and in the presence of noise we have:

$$\mathbf{g}(\boldsymbol{\rho}) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} E\{\boldsymbol{\zeta}(t)u_w(t)\} = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} E\{\boldsymbol{\zeta}(t)\hat{u}_w(t)\}. \tag{3.38}$$

Now in order to consider the spectrum of $u_w(t)$ in the control design, let the following correlation criterion be defined:

$$\begin{aligned} J_{CA}(\boldsymbol{\rho}) &= \mathbf{f}^T(\boldsymbol{\rho})\mathbf{f}(\boldsymbol{\rho}) + w_u \mathbf{g}^T(\boldsymbol{\rho})\mathbf{g}(\boldsymbol{\rho}) \\ &= \sum_{\tau=-l}^{l} \left[r^2_{ey_d}(\tau) + w_u r^2_{u_w y_d}(\tau) \right] \end{aligned} \tag{3.39}$$

where $r_{u_w y_d}(\tau)$ is the cross-correlation function between the input $u_w(t)$ and the desired output $y_d(t)$, and w_u is a positive scalar weighting factor. This new criterion can be interpreted in the frequency domain as:

$$\begin{aligned} \lim_{l\to\infty} J_{CA}(\boldsymbol{\rho}) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \left[|\Phi_{ey_d}(\omega)|^2 + w_u |\Phi_{u_w y_d}(\omega)|^2 \right] d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \left[|1 - F(\boldsymbol{\rho}, e^{-j\omega})G(e^{-j\omega})|^2 \right. \\ &\qquad \left. + \, w_u |F(\boldsymbol{\rho}, e^{-j\omega})U(e^{-j\omega})|^2 \right] \Phi^2_{y_d}(\omega) d\omega. \end{aligned} \tag{3.40}$$

Therefore, using the criterion in (3.39) and an appropriate choice of w_u the magnitude of the frequency response of $u_w(t)$ can be reduced in the frequency range where the spectrum of the desired output is large.

For a finite number of data, an approximation of the criterion can be obtained as:

$$J^N_{CA}(\boldsymbol{\rho}) = \hat{\mathbf{f}}^T(\boldsymbol{\rho})\hat{\mathbf{f}}(\boldsymbol{\rho}) + w_u \hat{\mathbf{g}}^T(\boldsymbol{\rho})\hat{\mathbf{g}}(\boldsymbol{\rho}) \tag{3.41}$$

where

$$\hat{\mathbf{g}}(\boldsymbol{\rho}) = \frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\zeta}(t)\boldsymbol{\varphi}^T(t)\boldsymbol{\rho} = \mathbf{Q}_u \boldsymbol{\rho} \quad \text{and} \quad \mathbf{Q}_u = \frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\zeta}(t)\boldsymbol{\varphi}^T(t).$$

The global minimum of this criterion is given by:

$$\boldsymbol{\rho}^N = (\mathbf{Q}_y^T \mathbf{Q}_y + w_u \mathbf{Q}_u^T \mathbf{Q}_u)^{-1} \mathbf{Q}_y^T \mathbf{r}. \tag{3.42}$$

4

Data-driven precompensator tuning for LPV systems

4.1 Introduction

Motivated by the interest to extend the methodology of the previous chapter to broader system classes, data-driven tuning of precompensators for LPV systems is investigated in this chapter. It is shown that if the ideal precompensator giving zero-mean tracking error exists in the precompensator parameterisation, the LPV transfer operators commute and a tuning technique is proposed which gives consistent estimates using measurements from just two experiments. For the more general case, where the LPV transfer operators do not commute, another algorithm is proposed requiring a number of experiments equal to twice the number of precompensator parameters. The algorithm leads to parameter estimates that converge to those that minimise the desirable mean squared error criterion.

The chapter is organised as follows. In Section 4.2 the ideal precompensator is defined and the LPV precompensator parameterisation is described. The tuning technique when the ideal precompensator exists in the precompensator parameterisation is presented in Section 4.3. In Section 4.4, the tuning method for the general case is developed. Simulation results are presented in Section 4.5. Finally, concluding remarks for the chapter are made in Section 4.6.

4.2 Problem formulation

In this chapter only measurement disturbances will be considered in order to simplify notation. The presence of zero-mean stochastic load disturbances would not affect the main results. Moreover the dependence of the signals on the scheduling parameter i.e. $y(\boldsymbol{\sigma}(t), t)$ will not be explicitly stated, again for notational clarity. With this notation the measured output of the SISO LPV system $G(\boldsymbol{\sigma}(t), q^{-1})$ is given by:

$$\begin{aligned} y(t) &= G(\boldsymbol{\sigma}(t), q^{-1})u(t) + n(t) \\ &= G(\boldsymbol{\sigma}(t), q^{-1})u(t) + H(\boldsymbol{\sigma}(t), q^{-1})\eta(t) \end{aligned} \tag{4.1}$$

where $H(\boldsymbol{\sigma}(t), q^{-1})$ is a, possibly LPV, transfer operator filtering the sequence of zero-mean, independent random variables $\eta(t)$ to give $n(t)$. The scheduling parameter vector contains the measurable signal(s) which correspond to the system's current operating point.

4.2.1 Assumptions

A4.1: $G(\boldsymbol{\sigma}(t), q^{-1})$ and $H(\boldsymbol{\sigma}(t), q^{-1})$ are LPV stable.
A4.2: The scheduling parameter $\boldsymbol{\sigma}(t)$ and the desired output $y_d(t)$ are bounded signals.
A4.3: $\eta(t)$ is a sequence of independent random variables with zero-mean values and unknown, bounded variances and fourth moments.

4.2.2 Ideal precompensator

The output of the system with an LPV precompensator $F(\boldsymbol{\sigma}(t), q^{-1})$, when the desired output $y_d(t)$ is applied at the precompensator's input, is given by (see Figure 4.1):

$$y(t) = G(\boldsymbol{\sigma}(t), q^{-1})F(\boldsymbol{\sigma}(t), q^{-1})y_d(t) + n(t). \tag{4.2}$$

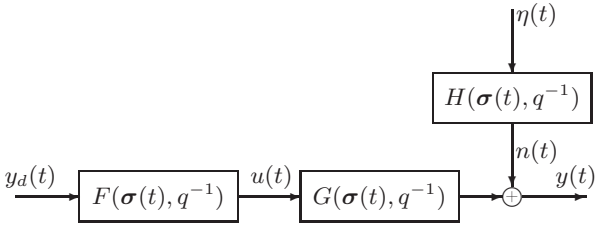

Fig. 4.1. System with precompensator

The ideal precompensator is defined as that which produces a zero-mean measured tracking error. It can be seen from (4.2) that it is the precompensator for which:

$$G(\boldsymbol{\sigma}(t), q^{-1})F(\boldsymbol{\sigma}(t), q^{-1}) = 1 \quad \forall \boldsymbol{\sigma}(t), \forall t. \tag{4.3}$$

A fact which should be noted is that due to the time-varying nature of the transfer operators, commutativity does not apply to them, in general. In fact, the backward-shift operator should obey a non-commutative multiplicative operation i.e.

$$q^{-i}\left(\boldsymbol{\sigma}(t)q^{-j}u(t)\right) = \boldsymbol{\sigma}(t-i)u(t-i-j).$$

4.2.3 Precompensator parameterisation

As in the LTI case, the precompensator is parameterised such that $F(\boldsymbol{\sigma}(t), q^{-1})$ is linear in its parameters and can be expressed as:

$$F(\boldsymbol{\rho}, \boldsymbol{\sigma}(t), q^{-1}) = \boldsymbol{\beta}^T(\boldsymbol{\sigma}(t), q^{-1})\boldsymbol{\rho} \tag{4.4}$$

where

$$\boldsymbol{\beta}(\boldsymbol{\sigma}(t), q^{-1}) = \Big[\beta_0^1(q^{-1})\big(\boldsymbol{\sigma}^T(t)\beta_0^2(q^{-1})\big), \beta_1^1(q^{-1})\big(\boldsymbol{\sigma}^T(t)\beta_1^2(q^{-1})\big), \\ \ldots, \beta_{n_\beta}^1(q^{-1})\big(\boldsymbol{\sigma}^T(t)\beta_{n_\beta}^2(q^{-1})\big)\Big]^T, \tag{4.5}$$

and $\boldsymbol{\rho} \in \mathbb{R}^{n_\rho}$, with $n_\rho = (n_\sigma + 1)(n_\beta + 1)$, is the vector of controller parameters:

$$\boldsymbol{\rho} = [\rho_0^0, \rho_0^1, \ldots, \rho_0^{n_\sigma}, \rho_1^0, \rho_1^1, \ldots, \rho_1^{n_\sigma}, \ldots, \rho_{n_\beta}^0, \rho_{n_\beta}^1, \ldots, \rho_{n_\beta}^{n_\sigma}]^T. \quad (4.6)$$

The $\beta_i^j(q^{-1})$ are LTI discrete-time transfer operators, which can be any orthogonal basis functions. In the sequel, however, for clarity of presentation, we suppose that $\beta_i^1(q^{-1}) = 1$ and $\beta_i^2(q^{-1}) = q^{-i}$. These choices mean $\boldsymbol{\beta}^T(\boldsymbol{\sigma}(t), q^{-1})$ is given by:

$$\begin{aligned}\boldsymbol{\beta}(\boldsymbol{\sigma}(t), q^{-1}) = [&\sigma_0(t), \sigma_1(t), \ldots, \sigma_{n_\sigma}(t), \sigma_0(t)q^{-1}, \sigma_1(t)q^{-1}, \ldots, \\ &\sigma_{n_\sigma}(t)q^{-1}, \ldots, \sigma_0(t)q^{-n_\beta}, \sigma_1(t)q^{-n_\beta}, \ldots, \sigma_{n_\sigma}(t)q^{-n_\beta}]^T \quad (4.7)\end{aligned}$$

where $\sigma_j(t)$ represents the jth element of $\boldsymbol{\sigma}(t)$. This parameterisation allows a wide range of dependence on the scheduling parameter to be described. For example each $\sigma_j(t)$ could represent a function of a different scheduling parameter. Alternatively the $\sigma_j(t)$ could be a set of orthogonal basis functions of a single scheduling parameter e.g. polynomials:

$$\sigma_j(t) = \sigma^j(t), \quad (4.8)$$

where $\sigma(t)$ is the single scheduling parameter considered.

With the above choices, $F(\boldsymbol{\rho}, \boldsymbol{\sigma}(t), q^{-1})$ is given by:

$$\begin{aligned}F(\boldsymbol{\rho}, \boldsymbol{\sigma}(t), q^{-1}) = &\left[\rho_0^0\sigma_0(t) + \rho_0^1\sigma_1(t) + \cdots + \rho_0^{n_\sigma}\sigma_{n_\sigma}(t)\right] \\ &+ \left[\rho_1^0\sigma_0(t) + \rho_1^1\sigma_1(t) + \cdots + \rho_1^{n_\sigma}\sigma_{n_\sigma}(t)\right] q^{-1} + \ldots \\ &+ \left[\rho_{n_\beta}^0\sigma_0(t) + \rho_{n_\beta}^1\sigma_1(t) + \cdots + \rho_{n_\beta}^{n_\sigma}\sigma_{n_\sigma}(t)\right] q^{-n_\beta}. \quad (4.9)\end{aligned}$$

Remark: In the case that the desired output $y_d(t)$ and scheduling parameter $\boldsymbol{\sigma}(t)$ are known a priori, they can be used to improve the tracking of systems with a time delay. This improvement is achieved by setting $\beta_i^1(q^{-1}) = q^\delta$, where δ equals the system's time delay. This fact can be illustrated via the following example. Consider the noise-free system with a time delay m:

$$y(t) = -a_1(\boldsymbol{\sigma}(t))y(t-1) + u(t-m). \tag{4.10}$$

We want $y(t) = y_d(t)$, so substituting this equality into the above equation gives:

$$u(t-m) = y_d(t) + a_1(\boldsymbol{\sigma}(t))y_d(t-1) \tag{4.11}$$

or

$$u(t) = y_d(t+m) + a_1(\boldsymbol{\sigma}(t+m))y_d(t+m-1),$$

which shows that the structure required for perfect tracking is achieved by choosing $\delta = m$. This result implies that $\boldsymbol{\sigma}(t+\delta)$ and $y_d(t+\delta)$ should be used at time t, which is possible if they are known in advance. Unfortunately, in applications where $\boldsymbol{\sigma}(t)$ is measured in real time advanced knowledge of $\boldsymbol{\sigma}(t)$ will not be available.

4.3 Tuning when LPV transfer operators commute

As mentioned previously, in general, time-varying operators do not commute. One case, however, where they do is when the two operators considered are reciprocal. Thus, in the case that the precompensator's parameterisation and parameters are such that (4.3) is satisfied then $F(\boldsymbol{\rho}_0, \boldsymbol{\sigma}(t))G(\boldsymbol{\sigma}(t)) = G(\boldsymbol{\sigma}(t))F(\boldsymbol{\rho}_0, \boldsymbol{\sigma}(t)) = 1$, where $\boldsymbol{\rho}_0$ are the parameters satisfying (4.3). This fact means that a similar tuning scheme to that employed for LTI precompensators is usable.

For clarity of presentation, the argument q^{-1} will be omitted where appropriate in the rest of this chapter.

4.3.1 Tuning scheme

We have that the measured tracking error of the system, with a precompensator, is given by:

$$e(t) = y_d(t) - G(\boldsymbol{\sigma}(t))F(\boldsymbol{\rho}, \boldsymbol{\sigma}(t))y_d(t) - n(t). \tag{4.12}$$

In the absence of noise and when $G(\boldsymbol{\sigma}(t))$ and $F(\boldsymbol{\rho}, \boldsymbol{\sigma}(t))$ commute, the same tracking error would be obtained if the positions of the system and the precompensator were swapped so that $F(\boldsymbol{\rho}, \boldsymbol{\sigma}(t))$ acts as a post-compensator. So, as for the LTI case, it is possible to estimate $e(t)$ from one set of data obtained from the system without a precompensator as:

$$\begin{aligned}\hat{e}(t) &= y_d(t) - \hat{y}(t) \\ &= y_d(t) - F(\boldsymbol{\rho}, \boldsymbol{\sigma}(t)) y_m(t) \\ &= y_d(t) - F(\boldsymbol{\rho}, \boldsymbol{\sigma}(t)) z_m(t) - F(\boldsymbol{\rho}, \boldsymbol{\sigma}(t)) n(t) \end{aligned} \tag{4.13}$$

where $z_m(t)$ and $y_m(t)$ are the noise-free and noisy system outputs, respectively, when $y_d(t)$ is applied as the input (see Figure 4.2).

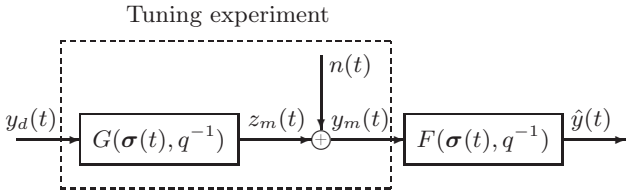

Fig. 4.2. Precompensator tuning scheme

This estimate can be expressed in linear regression form as:

$$\hat{e}(t) = y_d(t) - \boldsymbol{\phi}_m^T(t)\boldsymbol{\rho}$$

where:

$$\begin{aligned}\boldsymbol{\phi}_m(t) = [&\sigma_0(t)y_m(t), \sigma_1(t)y_m(t), \ldots, \sigma_{n_\sigma}(t)y_m(t), \\ &\sigma_0(t)y_m(t-1), \sigma_1(t)y_m(t-1), \ldots, \sigma_{n_\sigma}(t)y_m(t-1), \ldots, \\ &\sigma_0(t)y_m(t-n_\beta), \sigma_1(t)y_m(t-n_\beta), \ldots, \sigma_{n_\sigma}(t)y_m(t-n_\beta)]^T. \end{aligned} \tag{4.14}$$

It is possible to imagine that the precompensator parameters could be estimated by the minimisation of the quadratic cost function:

$$J^N_{mLS}(\boldsymbol{\rho}) = \frac{1}{2N}\sum_{t=0}^{N-1}\hat{e}^2(t), \tag{4.15}$$

whose minimiser is given by:

$$\boldsymbol{\rho}^N_{mLS} = \left[\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_m(t)\boldsymbol{\phi}^T_m(t)\right]^{-1}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_m(t)y_d(t). \tag{4.16}$$

However, as in the LTI case, when $F(\boldsymbol{\rho},\boldsymbol{\sigma}(t))$ is placed as a post-compensator it also filters the noise $n(t)$, as seen in Figure 4.2. The parameters that minimise the variance of the measured tracking error estimate will, therefore, not be the same as those that minimise the variance of the true measured tracking error.

The instrumental variables method can be used, nonetheless, to give consistent estimates of the ideal precompensator's parameter vector $\boldsymbol{\rho}_0$. For the IV estimate to converge to the ideal vector, the IV vector must be correlated with the non-noisy component of $\boldsymbol{\phi}_m(t)$, but not with the noise $n(t)$. These conditions are satisfied by the use of a vector similar to $\boldsymbol{\phi}_m(t)$, but with $y_m(t)$ obtained from a second experiment, performed in the same way as the first. The second experiment will, however, be affected by a different, independent noise realisation. This choice of IV vector is made rather than the choice of lagged inputs used in the last chapter because the output of a time-varying system is not necessarily correlated with the input. The IV estimate is thus given by:

$$\boldsymbol{\rho}^N_{mIV} = \left[\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}(t)\boldsymbol{\phi}^T_{m2}(t)\right]^{-1}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}(t)y_d(t), \tag{4.17}$$

where $\boldsymbol{\phi}_{m1}(t)$ and $\boldsymbol{\phi}_{m2}(t)$ are the $\boldsymbol{\phi}_m(t)$ from the two experiments using $y_{m1}(t)$ and $y_{m2}(t)$ respectively.

The consistency of the IV estimate is not directly obvious as, unlike the standard LTI case, the signals considered contain nonstationary stochastic components. The applicability of ergodicity type

results typically used in consistency analysis is not, therefore, immediately evident. An analysis is thus performed in the next subsection which demonstrates that the IV method does indeed lead to consistent estimates, despite the presence of these types of disturbances.

4.3.2 Consistency of the estimate

To see that the IV method gives a consistent estimate of $\boldsymbol{\rho}_0$ we begin by rewriting (4.17) as:

$$\begin{aligned}\boldsymbol{\rho}_{mIV}^N &= \left[\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}(t)\boldsymbol{\phi}_{m2}^T(t)\right]^{-1}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}(t)\boldsymbol{\phi}_z^T(t)\boldsymbol{\rho}_0 \\ &= \left[\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}(t)\boldsymbol{\phi}_{m2}^T(t)\right]^{-1}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}(t)\left(\boldsymbol{\phi}_{m2}^T(t)-\boldsymbol{\phi}_{n2}^T(t)\right)\boldsymbol{\rho}_0 \\ &= \boldsymbol{\rho}_0 - \left[\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}(t)\boldsymbol{\phi}_{m2}^T(t)\right]^{-1}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}(t)\boldsymbol{\phi}_{n2}^T(t)\boldsymbol{\rho}_0, \end{aligned} \tag{4.18}$$

where $\boldsymbol{\phi}_z(t)$ and $\boldsymbol{\phi}_{ni}(t)$ are similar to $\boldsymbol{\phi}_{mi}(t)$, but $y_{mi}(t)$ is replaced by $z_m(t)$ and $n_i(t)$ respectively. Additionally $\boldsymbol{\phi}_{mi}(t) = \boldsymbol{\phi}_z(t)+\boldsymbol{\phi}_{ni}(t)$.

In order for the parameter estimates to be consistent i.e. that $\boldsymbol{\rho}_{mIV}^N$ converges almost surely to $\boldsymbol{\rho}_0$ as $N \to \infty$, it is necessary that:

$$\text{i) } \lim_{N\to\infty}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}(t)\boldsymbol{\phi}_{m2}^T(t) \text{ be nonsingular.} \tag{4.19}$$

$$\text{ii) } \lim_{N\to\infty}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}(t)\boldsymbol{\phi}_{n2}^T(t) = \mathbf{0} \text{ w.p. 1.} \tag{4.20}$$

Condition (4.19) is a persistency of excitation condition. It is similar to the persistency of excitation condition found in methods for the identification of input-output form LPV models, as similar signals are involved i.e. noisy output signals multiplied by functions of the scheduling parameter. Persistency of excitation for LPV

identification was first considered in [5] and sufficient conditions for polynomial type coefficient dependence on the scheduling parameter are given. It is shown that if the system input signal is 'sufficiently rich' then persistency of excitation is ensured if the scheduling parameter 'visits' $n_\sigma+1$ distinct points infinitely many times, where n_σ is the order of the polynomial dependence. More recently [64] produced more general sufficient conditions for other types of coefficient dependence.

To show Condition (4.20) further analysis is required. We have that:

$$\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}(t)\boldsymbol{\phi}_{n2}^T(t)=\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{z}(t)\boldsymbol{\phi}_{n2}^T(t)+\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{n1}(t)\boldsymbol{\phi}_{n2}^T(t). \tag{4.21}$$

Considering the first matrix on the right hand side of (4.21), each of its elements is the time average of products of terms such as $\sigma_j(t)z_m(t-r)$ and $\sigma_i(t)n_2(t-p)$. Then, referring to Corollary 2.1, we can define:

$$\begin{aligned}\mathbf{s}(t)&=\begin{bmatrix}\sigma_j(t)z_m(t-r)\\ \sigma_i(t)n_2(t-p)\end{bmatrix}\\ &=\begin{bmatrix}0\\ \sigma_i(t)H(\boldsymbol{\sigma}(t-p))\eta_2(t-p)\end{bmatrix}\\ &\qquad\qquad+\begin{bmatrix}\sigma_j(t)G(\boldsymbol{\sigma}(t-r))y_d(t-r)\\ 0\end{bmatrix}\\ &=\begin{bmatrix}v_1^1(t)\\ v_2^1(t)\end{bmatrix}+\begin{bmatrix}w_1^1(t)\\ w_2^1(t)\end{bmatrix}\end{aligned} \tag{4.22}$$

where $w_1^1(t)$, $w_2^1(t)$, $v_1^1(t)$ and $v_2^1(t)$ have their obvious definitions. The signals $w_1^1(t)$ and $w_2^1(t)$ satisfy (2.39), due to the assumed LPV stability of $G(\boldsymbol{\sigma}(t))$ and the boundedness of $\sigma_j(t)$ and $y_d(t)$. Also $v_1^1(t)$ and $v_2^1(t)$ fit in with the desired form of (2.41) due to the assumed LPV stability of $H(\boldsymbol{\sigma}(t))$. The components of $\mathbf{s}(t)\mathbf{s}^T(t)$ give, amongst others, $\sigma_j(t)z_m(t-r)\sigma_i(t)n_2(t-p)$. So by applying Corollary 2.1 we can state that:

$$\left\| \frac{1}{N} \sum_{t=0}^{N-1} [\boldsymbol{\phi}_z(t)\boldsymbol{\phi}_{n2}^T(t) - E\{\boldsymbol{\phi}_z(t)\boldsymbol{\phi}_{n2}^T(t)\}] \right\|_F \to 0 \text{ w.p. } 1, \text{ as } N \to \infty,$$

and, as $z_m(t)$ and $n(t)$ are uncorrelated and $n(t)$ is zero mean, $E\{\boldsymbol{\phi}_z(t)\boldsymbol{\phi}_{n2}^T(t)\} = \mathbf{0}$ implying:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} \boldsymbol{\phi}_z(t)\boldsymbol{\phi}_{n2}^T(t) = \mathbf{0} \text{ w.p. } 1.$$

A similar result for the second matrix on the right hand side of (4.21) can be derived using Corollary 2.1 i.e. that:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} \boldsymbol{\phi}_{n1}(t)\boldsymbol{\phi}_{n2}^T(t) = \mathbf{0} \text{ w.p. } 1$$

and combining these two satisfies Condition (4.20), showing that, when Condition (4.19) is satisfied, the parameters estimates obtained with the IV method are consistent.

4.3.3 Noisy scheduling parameters

The values of the scheduling parameter $\sigma_j(t)$ used in the calculation of $\boldsymbol{\rho}_{mIV}^N$ will often be measured. They are, therefore, susceptible to measurement noise. In the case that the noise-to-signal ratio is very low, the effect of this noise can be neglected. However, if the ratio is not negligible the measurement noise may lead to biased parameter estimates.

The effect of noisy scheduling parameter measurements on the consistency of certain LPV identification methods has been considered in [9] (see appendix). It was shown that if the noise affecting the scheduling parameter is uncorrelated with that affecting the output signal and the dependency on the scheduling parameter is affine, then consistent parameter estimates are obtained using an IV vector

with scheduling parameter values measured during a second experiment. It will be shown in this section that a similar result holds for the consistency of the estimated precompensator parameters.

We express the noisy, measured scheduling parameter vector $\boldsymbol{\sigma}_v(t)$ as the sum of a noise-free component and a noisy component i.e.

$$\boldsymbol{\sigma}_v(t) = \boldsymbol{\sigma}(t) + \mathbf{v}_\sigma(t). \tag{4.23}$$

$\boldsymbol{\sigma}_v(t) \in \mathbb{R}^{n_\sigma}$ and $\mathbf{v}_\sigma(t) \in \mathbb{R}^{n_\sigma}$ are, therefore, stochastic vectors, and additionally $\mathbf{v}_\sigma(t)$ is assumed zero mean and to have independent realisations. The measured regressor vector is now given by:

$$\boldsymbol{\phi}_{mi}^{\boldsymbol{\sigma}_v}(t) = \boldsymbol{\phi}_{mi}(t) + \boldsymbol{\phi}_{mi}^{\mathbf{v}_\sigma}(t), \tag{4.24}$$

where $\boldsymbol{\phi}_{mi}(t)$ is as defined in (4.14) for experiment i and

$$\boldsymbol{\phi}_{mi}^{\mathbf{v}_\sigma}(t) = [\mathbf{v}_{\sigma i}^T(t)y_{mi}(t), \mathbf{v}_{\sigma i}^T(t)y_{mi}(t-1), \ldots, \mathbf{v}_{\sigma i}^T(t)y_{mi}(t-n_\beta)]^T. \tag{4.25}$$

The IV estimate is then given by:

$$\boldsymbol{\rho}_{mIV}^N = \left[\frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\phi}_{m1}^{\boldsymbol{\sigma}_v}(t)[\boldsymbol{\phi}_{m2}^{\boldsymbol{\sigma}_v}(t)]^T\right]^{-1} \frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\phi}_{m1}^{\boldsymbol{\sigma}_v}(t)y_d(t). \tag{4.26}$$

Consistency of the IV estimate with noisy scheduling parameters

To show that, using this estimate, consistent estimates can be obtained we start by rewriting (4.26) as:

$$\begin{aligned}\boldsymbol{\rho}_{mIV}^{N} &= \left[\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}^{\boldsymbol{\sigma}_v}(t)[\boldsymbol{\phi}_{m2}^{\boldsymbol{\sigma}_v}(t)]^T\right]^{-1}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}^{\boldsymbol{\sigma}_v}(t)\boldsymbol{\phi}_z^T(t)\boldsymbol{\rho}_0\\ &= \left[\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}^{\boldsymbol{\sigma}_v}(t)[\boldsymbol{\phi}_{m2}^{\boldsymbol{\sigma}_v}(t)]^T\right]^{-1}\\ &\qquad\qquad\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}^{\boldsymbol{\sigma}_v}(t)\left[\boldsymbol{\phi}_{m2}^{\boldsymbol{\sigma}_v}(t)-\boldsymbol{\phi}_{m2}^{\mathbf{v}_\sigma}(t)-\boldsymbol{\phi}_{n2}(t)\right]^T\boldsymbol{\rho}_0\\ &= \boldsymbol{\rho}_0-\left[\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}^{\boldsymbol{\sigma}_v}(t)[\boldsymbol{\phi}_{m2}^{\boldsymbol{\sigma}_v}(t)]^T\right]^{-1}\\ &\qquad\qquad\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}^{\boldsymbol{\sigma}_v}(t)\left[\boldsymbol{\phi}_{m2}^{\mathbf{v}_\sigma}(t)+\boldsymbol{\phi}_{n2}(t)\right]^T\boldsymbol{\rho}_0. \qquad (4.27)\end{aligned}$$

For the estimate to be consistent we require, in a similar fashion to before, that:

$$\text{i)}\ \lim_{N\to\infty}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}^{\boldsymbol{\sigma}_v}(t)[\boldsymbol{\phi}_{m2}^{\boldsymbol{\sigma}_v}(t)]^T \text{ be nonsingular.} \qquad (4.28)$$

$$\text{ii)}\ \lim_{N\to\infty}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}^{\boldsymbol{\sigma}_v}(t)\left[\boldsymbol{\phi}_{m2}^{\mathbf{v}_\sigma}(t)+\boldsymbol{\phi}_{n2}(t)\right]^T=\mathbf{0}\ \text{w.p. 1.} \qquad (4.29)$$

As before, Condition (4.28) is a persistency of excitation condition.

Condition (4.29) can, again, be analysed using Corollary 2.1. We first consider the matrix $\boldsymbol{\phi}_{m1}^{\boldsymbol{\sigma}_v}(t)[\boldsymbol{\phi}_{m2}^{\mathbf{v}_\sigma}(t)]^T$. This contains the following cross-product type terms:

$$\begin{aligned}&\sigma_{v1,j}(t)y_{m1}(t-p)v_{\sigma 2,i}(t)y_{m2}(t-r)\\ &\qquad=[\sigma_j(t)+v_{\sigma 1,j}(t)]y_{m1}(t-p)v_{\sigma 2,i}(t)y_{m2}(t-r). \qquad (4.30)\end{aligned}$$

Considering (4.30), it is equal to:

$$\begin{aligned}
&\sigma_j(t)\left[G(\boldsymbol{\sigma}(t-p))y_d(t-p)+H(\boldsymbol{\sigma}(t-p))\eta_1(t-p)\right]\\
&\quad v_{\sigma 2,i}(t)\left[G(\boldsymbol{\sigma}(t-r))y_d(t-r)+H(\boldsymbol{\sigma}(t-r))\eta_2(t-r)\right]\\
&\quad + v_{\sigma 1,j}(t)\left[G(\boldsymbol{\sigma}(t-p))y_d(t-p)+H(\boldsymbol{\sigma}(t-p))\eta_1(t-p)\right]\\
&\quad v_{\sigma 2,i}(t)\left[G(\boldsymbol{\sigma}(t-r))y_d(t-r)+H(\boldsymbol{\sigma}(t-r))\eta_2(t-r)\right]. \quad (4.31)
\end{aligned}$$

So, referring to Corollary 2.1, we can write:

$$\begin{aligned}
\mathbf{s}_2(t) &= \begin{bmatrix} 0 \\ \sigma_j(t)H(\boldsymbol{\sigma}(t-p))\eta_1(t-p) \\ v_{\sigma 2,i}(t)G(\boldsymbol{\sigma}(t-r))y_d(t-r) \\ v_{\sigma 2,i}(t)H(\boldsymbol{\sigma}(t-r))\eta_2(t-r) \\ v_{\sigma 1,j}(t)G(\boldsymbol{\sigma}(t-p))y_d(t-p) \\ v_{\sigma 1,j}(t)H(\boldsymbol{\sigma}(t-p))\eta_1(t-p) \end{bmatrix} \\
&\quad + \begin{bmatrix} \sigma_j(t)G(\boldsymbol{\sigma}(t-p))y_d(t-p) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} v_1^2(t) \\ v_2^2(t) \\ v_3^2(t) \\ v_4^2(t) \\ v_5^2(t) \\ v_6^2(t) \end{bmatrix} + \begin{bmatrix} w_1^2(t) \\ w_2^2(t) \\ w_3^2(t) \\ w_4^2(t) \\ w_5^2(t) \\ w_6^2(t) \end{bmatrix}. \quad (4.32)
\end{aligned}$$

In order to write $v_4^2(t) = v_{\sigma 2,i}(t)H(\boldsymbol{\sigma}(t-r))\eta_2(t-r)$ and $v_6^2(t) = v_{\sigma 1,j}(t)H(\boldsymbol{\sigma}(t-p))\eta_1(t-p)$ it is necessary that $v_{\sigma,j}(t)$ and $\eta(t)$ are uncorrelated. This condition is reasonable so long as the scheduling parameter is not the system output $y(t)$. If they are correlated the expected value of their product is non-zero and does not satisfy the corollary's assumptions on the stochastic component. The ergodicity of the signals would, therefore, not be provable in this case.

Additionally it is not possible to establish the ergodicity of the signals when the scheduling parameter has the polynomial depen-

dence discussed in (4.8) which is of an order greater than 1. The reason is that we would have higher order moments of the noise term affecting $\sigma(t)$, which would be non-zero mean. This, in turn, would imply that $v_{\sigma,j}(t)$ is non-zero mean and thus $v_4^2(t)$ and $v_6^2(t)$ are non-zero mean, violating the corollary's assumptions again.

With these conditions in mind, we see that amongst the elements of $\mathbf{s}_2(t)\mathbf{s}_2^T(t)$ are all the cross-terms found in (4.31), and thus:

$$\Big\|\frac{1}{N}\sum_{t=0}^{N-1}\big[\sigma_{v1,j}(t)y_{m1}(t-p)v_{\sigma 2,i}(t)y_{m2}(t-r) - E\{\sigma_{v1,j}(t)y_{m1}(t-p)v_{\sigma 2,i}(t)y_{m2}(t-r)\}\big]\Big\|_F \to 0 \quad \text{w.p. 1, as } N\to\infty. \tag{4.33}$$

Now since the vector $\mathbf{v}_\sigma(t)$ is assumed to be zero mean, uncorrelated with the system output and to have independent realisations we have that:

$$E\{\sigma_{v1,j}(t)y_{m1}(t-p)v_{\sigma 2,i}(t)y_{m2}(t-r)\} = E\{\sigma_{v1,j}(t)y_{m1}(t-p)y_{m2}(t-r)\}E\{v_{\sigma 2,i}(t)\} = 0. \tag{4.34}$$

This result implies that:

$$\lim_{N\to\infty}\frac{1}{N}\sum_{t=0}^{N-1}\sigma_{v1,j}(t)y_{m1}(t-p)v_{\sigma 2,i}(t)y_{m2}(t-r) = 0 \text{ w.p. } 1 \tag{4.35}$$

which in turn implies:

$$\lim_{N\to\infty}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\phi}_{m1}^{\boldsymbol{\sigma}_v}(t)[\boldsymbol{\phi}_{m2}^{\mathbf{v}_\sigma}(t)]^T = \mathbf{0} \text{ w.p. } 1. \tag{4.36}$$

We next consider the second matrix in (4.29) $\boldsymbol{\phi}_{m1}^{\boldsymbol{\sigma}_v}(t)\boldsymbol{\phi}_{n2}^T(t)$. This contains the following cross-product type terms:

$$\sigma_{v1,j}(t)y_{m1}(t-p)\sigma_i(t)n_2(t-r). \tag{4.37}$$

Under similar conditions on the scheduling parameter, Corollary 2.1 can be used to show that:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} \sigma_{v1,j}(t) y_{m1}(t-p) \sigma_i(t) n_2(t-r) = 0 \text{ w.p. } 1 \tag{4.38}$$

and so:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} \boldsymbol{\phi}_{m1}^{\boldsymbol{\sigma}_v}(t) \boldsymbol{\phi}_{n2}^{T}(t) = \mathbf{0} \text{ w.p. } 1. \tag{4.39}$$

Results (4.36) and (4.39) imply that Condition (4.29) is satisfied by the IV estimate (4.26) so, when Condition (4.28) is also satisfied, the estimate is consistent despite the contamination of the scheduling parameter measurements by noise.

4.4 Tuning when LPV transfer operators do not commute

In general condition (4.3) is unlikely to be satisfied by a precompensator with the linear parameterisation proposed. In this case, the order of the precompensator can be increased until the condition is approximately satisfied and the method of the previous section can be used. The order required to approximately satisfy (4.3) may be too large to be implemented on the real system in practice, however. A method that can be used in this situation is therefore developed below.

4.4.1 Tuning scheme

The signal $u(t) = F(\boldsymbol{\rho}, \boldsymbol{\sigma}(t)) y_d(t)$ can be expressed as:

$$u(t) = F(\boldsymbol{\rho}, \boldsymbol{\sigma}(t)) y_d(t) = \boldsymbol{\phi}^T(t) \boldsymbol{\rho} \tag{4.40}$$

where

$$\begin{aligned}\boldsymbol{\phi}^T(t) = [&\sigma_0(t)y_d(t), \sigma_1(t)y_d(t), \ldots, \sigma_{n_\sigma}(t)y_d(t),\\ &\sigma_0(t)y_d(t-1), \sigma_1(t)y_d(t-1), \ldots, \sigma_{n_\sigma}(t)y_d(t-1), \ldots,\\ &\sigma_0(t)y_d(t-n_\beta), \sigma_1(t)y_d(t-n_\beta), \ldots, \sigma_{n_\sigma}(t)y_d(t-n_\beta)]. \quad (4.41)\end{aligned}$$

It is now possible to write the output as:

$$\begin{aligned}y(t) &= G(\boldsymbol{\sigma}(t))F(\boldsymbol{\rho}, \boldsymbol{\sigma}(t))y_d(t) + n(t)\\ &= G(\boldsymbol{\sigma}(t))\boldsymbol{\phi}^T(t)\boldsymbol{\rho} + n(t)\\ &= \mathbf{x}^T(t)\boldsymbol{\rho} + n(t), \quad (4.42)\end{aligned}$$

where $\mathbf{x}(t) = G(\boldsymbol{\sigma}(t))\boldsymbol{\phi}(t) \in \mathbb{R}^{n_\rho}$.

The measured tracking error can then be expressed as:

$$e(t) = y_d(t) - \mathbf{x}^T(t)\boldsymbol{\rho} - n(t) \quad (4.43)$$

or in vector form as:

$$\boldsymbol{e} = \mathbf{y_d} - \mathbf{X}\boldsymbol{\rho} - \mathbf{n}, \quad (4.44)$$

where the vector $\boldsymbol{e}$ is given by:

$$\boldsymbol{e} = [e(0), e(1), \ldots, e(N-1)]^T, \quad (4.45)$$

$\mathbf{y_d}$ and $\mathbf{n}$ are defined similarly, and the matrix $\mathbf{X}$ is:

$$\mathbf{X} = [\mathbf{x}(0), \mathbf{x}(1), \ldots, \mathbf{x}(N-1)]^T. \quad (4.46)$$

The aim is to find $\boldsymbol{\rho}$ such that the average tracking error is small, therefore a logical objective is to minimise its mean squared value i.e. to find the $\boldsymbol{\rho}$ that minimises:

$$J_{LMS}^N(\boldsymbol{\rho}) = \frac{1}{2N} E\left\{\boldsymbol{e}^T(\boldsymbol{\rho})\boldsymbol{e}(\boldsymbol{\rho})\right\}, \quad (4.47)$$

where the dependence of $\boldsymbol{e}$ on $\boldsymbol{\rho}$ has been shown explicitly.

The minimiser of (4.47) is given by:

$$\boldsymbol{\rho}^N_{LMS} = \left[\frac{1}{N}\mathbf{X}^T\mathbf{X}\right]^{-1}\frac{1}{N}\mathbf{X}^T\mathbf{y_d}. \tag{4.48}$$

In the case that $G(\boldsymbol{\sigma}(t))$ were known exactly, $\mathbf{X}$ could be calculated, followed by $\boldsymbol{\rho}^N_{LMS}$. $G(\boldsymbol{\sigma}(t))$ is never known exactly, however, and the model uncertainty leads to a non-optimal $\boldsymbol{\rho}$. It is, nevertheless, possible to obtain an estimate $\hat{\mathbf{X}}$ of the matrix $\mathbf{X}$ without the use of a model via a series of experiments on the real system. This can be seen to be the case by noting that the t, jth element of the matrix $\mathbf{X}$ is the output of the system $G(\boldsymbol{\sigma}(t))$ when the jth element of $\boldsymbol{\phi}(t)$ is applied as an input. Thus for each column of $\mathbf{X}$ an experiment can be carried out on the real system i.e. n_ρ experiments in total. In reality an estimate, rather than the exact value, of $\mathbf{X}$ will be found as each experiment will have its own noise realisation $n_j(t)$. The estimate of $\mathbf{X}$ is:

$$\hat{\mathbf{X}} = \mathbf{X} + \mathbf{N}, \tag{4.49}$$

where $\mathbf{N}$ is a matrix whose t, jth element is $n_j(t)$. Substituting $\hat{\mathbf{X}}$ in for $\mathbf{X}$ in (4.48) we have:

$$\begin{aligned}\hat{\boldsymbol{\rho}}^N_{LMS} &= \left[\frac{1}{N}(\mathbf{X}+\mathbf{N})^T(\mathbf{X}+\mathbf{N})\right]^{-1}\frac{1}{N}(\mathbf{X}+\mathbf{N})^T\mathbf{y_d} \\ &= \left[\frac{1}{N}(\mathbf{X}^T\mathbf{X}+\mathbf{N}^T\mathbf{X}+\mathbf{X}^T\mathbf{N}+\mathbf{N}^T\mathbf{N})\right]^{-1}\frac{1}{N}(\mathbf{X}+\mathbf{N})^T\mathbf{y_d}.\end{aligned} \tag{4.50}$$

Therefore when $\hat{\mathbf{X}}$ is used in place of $\mathbf{X}$ in (4.48) the presence of the noise in the experiments performed to find $\hat{\mathbf{X}}$ will mean that the minimising value $\boldsymbol{\rho}^N_{LMS}$ cannot be calculated i.e. $\hat{\boldsymbol{\rho}}^N_{LMS} \neq \boldsymbol{\rho}^N_{LMS}$.

One way of dealing with this problem is to use instrumental variables again. This time two estimates of $\mathbf{X}$, $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_2$, are used. They are obtained from two sets of the n_ρ experiments previously described. The IV estimate is then given by:

$$\hat{\boldsymbol{\rho}}^N_{IV} = \left[\frac{1}{N}\hat{\mathbf{X}}_1^T\hat{\mathbf{X}}_2\right]^{-1}\frac{1}{N}\hat{\mathbf{X}}_1^T\mathbf{y_d}. \tag{4.51}$$

Remarks:

1. The same $G(\boldsymbol{\sigma}(t))$ should be used at each t for all the experiments i.e. when different signals $u(t)$ are applied. It is therefore necessary that the scheduling parameter $\boldsymbol{\sigma}(t)$ be independent of $u(t)$, which may not be the case for certain quasi-LPV systems where the scheduling parameter can be an input-dependent, internal variable. Nonetheless, a number of LPV systems found in practice do satisfy this requirement, such as x-y positioning tables where the dynamics of one stage depend on the position of the other and not their own.
2. The specific choice of instrumental variables obtained from a second set of experiments is used because it can be shown that consistent estimates of the true minimiser of (4.47) are obtained using them, as will be seen in the next subsection. This is not necessarily possible for a general choice of instrumental variables. A disadvantage of this specific choice is that, due to the second set of noise realisations affecting the instrumental variables, the variance of the estimates can be large when the number of data measured is small.

4.4.2 Consistency of the estimate

To demonstrate that the IV estimate (4.51) converges to the true minimiser of (4.47) when $N \to \infty$ we first write:

$$\begin{aligned}\hat{\boldsymbol{\rho}}_{IV}^{N} &= \left[\frac{1}{N}(\mathbf{X}+\mathbf{N}_1)^T(\mathbf{X}+\mathbf{N}_2)\right]^{-1}\frac{1}{N}(\mathbf{X}+\mathbf{N}_1)^T\mathbf{y_d} \\ &= \left[\frac{1}{N}(\mathbf{X}^T\mathbf{X}+\mathbf{N}_1^T\mathbf{X}+\mathbf{X}^T\mathbf{N}_2+\mathbf{N}_1^T\mathbf{N}_2)\right]^{-1}\frac{1}{N}(\mathbf{X}+\mathbf{N}_1)^T\mathbf{y_d}\end{aligned} \tag{4.52}$$

where $\mathbf{N}_1$ and $\mathbf{N}_2$ denote matrices of noise realisations associated with the first and second set of experiments. For $\hat{\boldsymbol{\rho}}_{IV}^{N}$ to converge to $\boldsymbol{\rho}_{LMS}^{N}$ we require $\frac{1}{N}\left[\mathbf{N}_1^T\mathbf{X}+\mathbf{X}^T\mathbf{N}_2+\mathbf{N}_1^T\mathbf{N}_2\right]$ and $\frac{1}{N}\mathbf{N}_1^T\mathbf{y_d}$ to

converge to zero as $N \to \infty$. This can be shown to be the case by noting that the i,jth element of the matrix $\mathbf{N}_1^T\mathbf{X}$ is given by:

$$\begin{aligned}\frac{1}{N}[\mathbf{N}_1^T\mathbf{X}]_{i,j} &= \frac{1}{N}\sum_{t=0}^{N-1} n1_i(t)\mathbf{X}_{t,j} \\ &= \frac{1}{N}\sum_{t=0}^{N-1}[H(\boldsymbol{\sigma}(t))\eta 1_i(t)][G(\boldsymbol{\sigma}(t))\boldsymbol{\phi}_j(t)],\end{aligned} \tag{4.53}$$

where $n1_i(t)$ corresponds to the t,i element of $\mathbf{N}_1$, and $n1_i(t) = H(\boldsymbol{\sigma}(t))\eta 1_i(t)$. Corollary 2.1 is applicable to this time average and can be used to show that:

$$\left\| \frac{1}{N}\sum_{t=0}^{N-1}[n1_i(t)\mathbf{X}_{t,j} - E\{n1_i(t)\mathbf{X}_{t,j}\}] \right\|_F \to 0 \quad \text{w.p. 1, as } N \to \infty, \tag{4.54}$$

and since $E\{n1_i(t)\mathbf{X}_{t,j}\} = 0$ we have

$$\lim_{N\to\infty} \frac{1}{N}\sum_{t=0}^{N-1} n1_i(t)\mathbf{X}_{t,j} = 0 \text{ w.p. 1.}$$

This result can be applied to each element of $\mathbf{N}_1^T\mathbf{X}$, and similar results hold for $\mathbf{X}^T\mathbf{N}_2$, $\mathbf{N}_1^T\mathbf{N}_2$ and $\mathbf{N}_1^T\mathbf{y_d}$. Therefore, $\hat{\boldsymbol{\rho}}_{IV}^N \to \boldsymbol{\rho}_{LMS}^N$ as $N \to \infty$ almost surely.

4.4.3 Noisy scheduling parameters

As was the case for the method in the previous section, it is possible that the values of scheduling parameters used will have been measured and thus contaminated by noise. For example consider the case when the precompensator is tuned to control the lower axis of an x-y positioning table whose dynamics depend on the position of the upper axis. In this case the scheduling parameter would be the position of the upper axis. This can either be the position reference

signal or the measured position. The former may not be a very good measure of the true position and the latter is affected by noise. When the noise is negligible its effect can be neglected and the IV estimate (4.51) can be used. However, when this is not the case, it will be shown that, under certain conditions on the scheduling parameter, by using two independent sets of scheduling parameter measurements, the IV estimate will lead to consistent estimates.

We have that the noisy, measured scheduling parameter vector $\boldsymbol{\sigma}_v(t)$ is expressed as the sum of a noise-free component and a noisy component as in (4.23). When this is used to construct the $\boldsymbol{\phi}(t)$ vector, the noisy $\boldsymbol{\phi}(t)$ for the i$^{\text{th}}$ set of experiments is given by:

$$\boldsymbol{\phi}_i^{\boldsymbol{\sigma}_v}(t) = \boldsymbol{\phi}(t) + \boldsymbol{\phi}_i^{\mathbf{v}_\sigma}(t), \tag{4.55}$$

where $\boldsymbol{\phi}(t)$ is as defined in (4.41) and

$$\boldsymbol{\phi}_i^{\mathbf{v}_\sigma}(t) = [\mathbf{v}_{\sigma i}^T(t)y_d(t), \mathbf{v}_{\sigma i}^T(t)y_d(t-1), \ldots, \mathbf{v}_{\sigma i}^T(t)y_d(t-n_\beta)]^T. \tag{4.56}$$

It is therefore the elements of $\boldsymbol{\phi}_i^{\boldsymbol{\sigma}_v}(t)$ that are now used as the inputs for the n_ρ experiments. This will lead to the new estimate of $\mathbf{X}$:

$$\hat{\mathbf{X}}_i^{\boldsymbol{\sigma}_v} = \mathbf{X} + \mathbf{X}_i^{\mathbf{v}_\sigma} + \mathbf{N}_i. \tag{4.57}$$

The IV estimate is now:

$$\hat{\boldsymbol{\rho}}_{IV}^N = \left[\frac{1}{N}[\hat{\mathbf{X}}_1^{\boldsymbol{\sigma}_v}]^T\hat{\mathbf{X}}_2^{\boldsymbol{\sigma}_v}\right]^{-1}\frac{1}{N}[\hat{\mathbf{X}}_1^{\boldsymbol{\sigma}_v}]^T\mathbf{y_d}. \tag{4.58}$$

Consistency of the IV estimate with noisy scheduling parameters

To demonstrate that the IV estimate (4.58) converges to the true minimiser of (4.47) when $N \to \infty$ we first write:

$$\hat{\boldsymbol{\rho}}_{IV}^{N} = \left[\frac{1}{N}(\mathbf{X}+\mathbf{X}_1^{\mathbf{v}_\sigma}+\mathbf{N}_1)^T(\mathbf{X}+\mathbf{X}_2^{\mathbf{v}_\sigma}+\mathbf{N}_2)\right]^{-1} \frac{1}{N}(\mathbf{X}+\mathbf{X}_1^{\mathbf{v}_\sigma}+\mathbf{N}_1)^T\mathbf{y_d}$$
$$= \left[\frac{1}{N}\big(\mathbf{X}^T\mathbf{X}+\mathbf{X}^T\mathbf{X}_2^{\mathbf{v}_\sigma}+\mathbf{X}^T\mathbf{N}_2+[\mathbf{X}_1^{\mathbf{v}_\sigma}]^T\mathbf{X}+[\mathbf{X}_1^{\mathbf{v}_\sigma}]^T\mathbf{X}_2^{\mathbf{v}_\sigma} + [\mathbf{X}_1^{\mathbf{v}_\sigma}]^T\mathbf{N}_2+\mathbf{N}_1^T\mathbf{X}+\mathbf{N}_1^T\mathbf{X}_2^{\mathbf{v}_\sigma}+\mathbf{N}_1^T\mathbf{N}_2\big)\right]^{-1} \frac{1}{N}\big(\mathbf{X}+\mathbf{X}_1^{\mathbf{v}_\sigma}+\mathbf{N}_1\big)^T\mathbf{y_d}. \tag{4.59}$$

For $\hat{\boldsymbol{\rho}}_{IV}^{N}$ to converge to $\boldsymbol{\rho}_{LMS}^{N}$ we require $\frac{1}{N}(\mathbf{X}^T\mathbf{X}_2^{\mathbf{v}_\sigma}+\mathbf{X}^T\mathbf{N}_2+[\mathbf{X}_1^{\mathbf{v}_\sigma}]^T\mathbf{X}+[\mathbf{X}_1^{\mathbf{v}_\sigma}]^T\mathbf{X}_2^{\mathbf{v}_\sigma}+[\mathbf{X}_1^{\mathbf{v}_\sigma}]^T\mathbf{N}_2+\mathbf{N}_1^T\mathbf{X}+\mathbf{N}_1^T\mathbf{X}_2^{\mathbf{v}_\sigma}+\mathbf{N}_1^T\mathbf{N}_2)$ and $\frac{1}{N}(\mathbf{X}_1^{\mathbf{v}_\sigma}+\mathbf{N}_1)^T\mathbf{y_d}$ to converge to zero as $N\to\infty$. In a similar analysis to that previously presented in Subsection 4.4.2 this can be shown to be the case under the assumptions that the noise affecting the scheduling parameter is not correlated with that affecting the output and that the scheduling parameter's dependence is affine.

4.5 Simulation results

Simulations are used to demonstrate the effectiveness of the proposed methods.

Example 1

The first method, presented in Section 4.3, is applied to a system represented by the following transfer operator:

$$G(\boldsymbol{\sigma}(t),q^{-1}) = \frac{1}{a_0(\boldsymbol{\sigma}(t))+a_1(\boldsymbol{\sigma}(t))q^{-1}+a_2(\boldsymbol{\sigma}(t))q^{-2}} \tag{4.60}$$

where

$$a_0(\boldsymbol{\sigma}(t)) = 1 - 0.2\sigma(t) + 0.7\sigma^2(t),$$
$$a_1(\boldsymbol{\sigma}(t)) = 1 - 0.5\sigma(t) + 0.2\sigma^2(t)$$
and
$$a_2(\boldsymbol{\sigma}(t)) = 1 - 0.7\sigma(t) - 0.1\sigma^2(t).$$

The precompensator satisfying Condition (4.3) and giving perfect mean tracking can be represented exactly using the parameterisation proposed and thus the first method can be employed to find the correct parameters; they being:

$$\boldsymbol{\rho}_0 = [1, -0.2, 0.7, 1, -0.5, 0.2, 1, -0.7, -0.1]^T. \tag{4.61}$$

The desired output $y_d(t)$ is a pseudo random binary signal (PRBS) with an amplitude of 1 and a shift register length of 12, giving $N = 4095$. The noise transfer operator is given by $H(\boldsymbol{\sigma}(t)) = 1 - G(\boldsymbol{\sigma}(t))$. Its input $\eta(t)$ is taken as a zero-mean, normally distributed, stationary random sequence with a variance of 0.015. The scheduling parameter is chosen as $\sigma(t) = 0.5 + 0.4\sin\left(\frac{2\pi t}{3}\right)$. This choice of $\sigma(t)$ satisfies the persistency of excitation conditions in [5] as it visits $(n_\sigma + 1)$ distinct points periodically. The $n(t)$ resulting from these choices has an average standard deviation of 0.1287, where the average standard deviation is calculated as $(\frac{1}{N}\sum_{t=0}^{N-1}\hat{E}\{n^2(t)\})^{1/2}$. The estimated expected value $\hat{E}$ is evaluated using the ensemble average over 200 simulations, each with a new, independent noise realisation.

Both the least squares and IV variants of the first method are tested. The orders corresponding to the precompensator satisfying (4.3) are used i.e. $n_\beta = 2$ and $n_\sigma = 2$. The mean square parametric distance between the true and estimated parameters is presented in Table 4.1, along with the mean square tracking error achieved in the absence of noise using the calculated precompensators.

It can be seen that, due to its reduced sensitivity to the stochastic disturbances, the IV method gives better estimates of the true parameters and thus produces precompensators that give better average tracking performance.

	$\hat{E}\{(\boldsymbol{\rho}_0 - \boldsymbol{\rho}^N)^T(\boldsymbol{\rho}_0 - \boldsymbol{\rho}^N)\}$	$\hat{E}\left\{\frac{1}{N}\sum_{t=0}^{N-1} e^2(t)\right\}$
mLS	0.0457	0.00102
mIV	0.0123	0.00009

Table 4.1. Results for Example 1

Example 2

A second example is carried out to test the effect of noisy scheduling parameter measurements on the precompensator parameters estimated using the first method.

The system used in this simulation is given by:

$$G(\boldsymbol{\sigma}(t), q^{-1}) = \frac{1}{a_0(\boldsymbol{\sigma}(t)) + a_1(\boldsymbol{\sigma}(t))q^{-1}} \tag{4.62}$$

where

$$a_0(\boldsymbol{\sigma}(t)) = 1 - 0.2\sigma(t) \qquad \text{and} \qquad a_1(\boldsymbol{\sigma}(t)) = 1 - 0.5\sigma(t).$$

This system is chosen because its coefficients depend affinely on the scheduling parameter. As discussed in Subsection 4.3.3, it is not possible to obtain consistent estimates when the scheduling parameter measurements are noisy if the dependency is not affine. The desired output $y_d(t)$ is the same signal as that used in Simulation 1. The true scheduling parameter is given by $\sigma(t) = 0.5 + 0.4\cos(\pi t)$, which satisfies the persistency of excitation conditions in [5]. The noise $\mathbf{v}_\sigma(t)$ contaminating the measurements of the scheduling parameter signal is a zero-mean, normally distributed, stationary random sequence with a variance of 0.01. The disturbance $n(t)$ affecting the output is set to zero in order to better emphasise the effect of the scheduling parameter noise. Both the least squares and IV variants of the first method are tested. This time the IV vector uses $\sigma(t)$ measured during the second experiment. The values $n_\beta = 1$ and

$n_\sigma = 1$ are used as they correspond to the precompensator giving perfect mean tracking whose parameters are:

$$\boldsymbol{\rho}_0 = [1, -0.2, 1, -0.5]^T. \tag{4.63}$$

The results are shown in Table 4.2. Again, the expected value is estimated using the ensemble average over 200 simulations.

	$\hat{E}\{(\boldsymbol{\rho}_0 - \boldsymbol{\rho}^N)^T(\boldsymbol{\rho}_0 - \boldsymbol{\rho}^N)\}$	$\hat{E}\left\{\frac{1}{N}\sum_{t=0}^{N-1} e^2(t)\right\}$
mLS	0.00207	0.00113
mIV	0.00002	0.00002

Table 4.2. Results for Example 2

It is clearly evident that the IV variant gives superior parameter estimates over the LS method. In turn this leads to much better tracking performance.

Example 3

A third example is performed to see how the methods behave when Condition (4.3) cannot be satisfied using the parameterisation proposed. The system used in this simulation is given by:

$$G(\boldsymbol{\sigma}(t), q^{-1}) = \frac{b_0(\boldsymbol{\sigma}(t)) + b_1(\boldsymbol{\sigma}(t))q^{-1}}{a_0(\boldsymbol{\sigma}(t)) + a_1(\boldsymbol{\sigma}(t))q^{-1}} \tag{4.64}$$

where

$$\begin{aligned} b_0(\boldsymbol{\sigma}(t)) &= 0.5 - 0.4\sigma(t), & b_1(\boldsymbol{\sigma}(t)) &= 0.2 - 0.3\sigma(t) \\ a_0(\boldsymbol{\sigma}(t)) &= 1 - 0.2\sigma(t), & a_1(\boldsymbol{\sigma}(t)) &= 1 - 0.5\sigma(t). \end{aligned} \tag{4.65}$$

The desired output $y_d(t)$ and scheduling parameter $\sigma(t)$ are the same signals as those used in Simulation 2. The values $n_\beta = 1$ and $n_\sigma = 1$ are chosen.

A simulation is first carried out without any disturbances to test the methods. Since it is not possible to satisfy Condition (4.3) with the finite order precompensator used, the first method is not theoretically applicable. It is tested, nonetheless, to see how it performs. The results are shown in Table 4.3. As the parameters corresponding to the ideal precompensator do not exist in this case, the distance between those minimising the LMS criterion, calculated using (4.48), and those estimated is shown instead. The IV methods are not tested because in the absence of stochastic disturbances they give the same values.

	$(\boldsymbol{\rho}^N_{LMS} - \boldsymbol{\rho}^N)^T(\boldsymbol{\rho}^N_{LMS} - \boldsymbol{\rho}^N)$	$\frac{1}{N}\sum_{t=0}^{N-1} e^2(t)$
Without precomp.	–	0.753
mLS	10.802	1.857
LMS	0	0.325

Table 4.3. Results for Example 3 in the absence of disturbances

From Table 4.3 it can be observed that, as expected, the precompensator calculated with the general method, presented in Section 4.4, gives a smaller tracking error. Furthermore the first method does not find the parameters that minimise the LMS criterion and actually estimates parameters that produce a larger tracking error than that obtained without a precompensator, clearly illustrating the need for the general method.

The general method is then tested in the presence of disturbances. The noise transfer operator is, again, given by $H(\boldsymbol{\sigma}(t)) = 1 - G(\boldsymbol{\sigma}(t))$ and $\eta(t)$ is a zero-mean, normally distributed random sequence with a variance of 0.015. The average standard deviation of $n(t)$ this time

is 0.1063. The simulations are again carried out 200 times. The results are shown in Table 4.4.

	$\hat{E}\{(\boldsymbol{\rho}_{LMS}^N - \hat{\boldsymbol{\rho}}^N)^T(\boldsymbol{\rho}_{LMS}^N - \hat{\boldsymbol{\rho}}^N)\}$	$\hat{E}\left\{\frac{1}{N}\sum_{t=0}^{N-1} e^2(t)\right\}$
LMS	35.0564	0.642
IV	7.9891	0.400

Table 4.4. Results for Example 3 in the presence of disturbances

We see that the presence of the stochastic disturbances causes a large bias in the parameter estimates found using the LMS technique. This bias, in turn, greatly deteriorates the tracking performance obtained. The IV technique, however, is less sensitive to the presence of the disturbances and the parameters calculated are much closer to the true minimising parameters. In turn, an average tracking error that is much closer to the value found in the absence of disturbances is obtained.

4.6 Conclusions

Direct data-driven tuning methods of precompensators for LPV systems have been developed in this chapter.

Two techniques have been proposed. The first one, applicable when the precompensator and the system commute, only requires two experiments in order to obtain consistent estimates of the parameters that give perfect mean tracking. The second one, which does not require that the precompensator and the system commute, uses a number of experiments equal to twice the number of precompensator parameters. It is demonstrated that the computed parameters converge to those minimising the mean squared tracking

error in the presence of noise. The methods's effectiveness has been demonstrated in simulation.

The main benefit of the first method over the second method is clearly the reduced number of experiments required. However, it is only applicable to the reduced class of systems that commute with the precompensator. The second method is more general as does not have this condition and thus is more practically applicable. The use of sufficiently long data lengths is, however, important in order to keep the variance of the parameter estimates small.

The second method is, in fact, applicable to a more general system class than LPV systems, that of stable, nonlinear systems whose input space is extended by a scheduling parameter. The noise filter must, however, remain linear in order for the consistency of the estimates to be demonstrable.

Appendix

4.A Input weighting

As discussed for the LTI precompensator tuning case, in practice it may be of interest to be able to weight an input in the system in order to stop it becoming too large when the precompensator is used. Similarly to the LTI case, we have that in general the input concerned can be described as:

$$\begin{aligned} u_W(t) &= U(\boldsymbol{\sigma}(t))u(t) + H_u(\boldsymbol{\sigma}(t))n(t) \\ &= U(\boldsymbol{\sigma}(t))u(t) + n_u(t). \end{aligned} \tag{4.66}$$

It is possible to affect the magnitude of $u_W(t)$ by including a weighting term on it in the cost function:

$$J^N_{LMS}(\boldsymbol{\rho}) = \frac{1}{2N}\left[E\left\{\boldsymbol{e}^T(\boldsymbol{\rho})\mathbf{W}_e\boldsymbol{e}(\boldsymbol{\rho})\right\} + E\left\{\mathbf{u}^T_W(\boldsymbol{\rho})\mathbf{W}_u\mathbf{u}_W(\boldsymbol{\rho})\right\}\right], \tag{4.67}$$

where $\mathbf{W}_e$ and $\mathbf{W}_u$ are positive- and positive-semidefinite weighting matrices, respectively, and $\mathbf{u}_W$ is defined similarly to $\boldsymbol{e}$ in (4.45).

Additionally, similarly to (4.42), we can write:

$$\begin{aligned} u_W(t) &= U(\boldsymbol{\sigma}(t))\boldsymbol{\phi}^T(t)\boldsymbol{\rho} + n_u(t) \\ &= \mathbf{x}^T_u(t)\boldsymbol{\rho} + n_u(t). \end{aligned} \tag{4.68}$$

Therefore we have:

$$\begin{aligned} \mathbf{u}_W(\boldsymbol{\rho}) &= \begin{bmatrix} \mathbf{x}^T_u(0) \\ \mathbf{x}^T_u(1) \\ \vdots \\ \mathbf{x}^T_u(N-1) \end{bmatrix}\boldsymbol{\rho} + \mathbf{n}_u \\ &= \mathbf{X}_u\boldsymbol{\rho} + \mathbf{n}_u. \end{aligned} \tag{4.69}$$

The minimiser of (4.67) is thus:

$$\boldsymbol{\rho}_{LMS}^{N} = \left[\frac{1}{N}(\mathbf{X}^T\mathbf{W}_e\mathbf{X} + \mathbf{X}_u^T\mathbf{W}_u\mathbf{X}_u)\right]^{-1}\frac{1}{N}\mathbf{X}^T\mathbf{y_d}. \tag{4.70}$$

In the same way that $\mathbf{X}$ is not known exactly, neither is $\mathbf{X}_u$, but an estimate $\hat{\mathbf{X}}_u$ can be obtained using the same experiments used to estimate $\mathbf{X}$. This time not only the system output $y(t)$ should be measured during the experiments but also the input to be weighted $u_W(t)$. Because this input can also be affected by noise, it is necessary to use measurements from two experiments again and use the IV estimate:

$$\hat{\boldsymbol{\rho}}_{IV}^{N} = \left[\frac{1}{N}(\hat{\mathbf{X}}_1^T\mathbf{W}_e\hat{\mathbf{X}}_2 + \hat{\mathbf{X}}_{u1}^T\mathbf{W}_u\hat{\mathbf{X}}_{u2})\right]^{-1}\frac{1}{N}\hat{\mathbf{X}}_1^T\mathbf{y_d}. \tag{4.71}$$

It should be noted that input weighting cannot be considered for the method in Section 4.3 where the exact system inverse exists in the precompensator parameterisation. This is because there the aim is to identify the precompensator that satisfies Condition (4.3) exactly. This would not be possible if a trade-off is made with the reduction of the input signal amplitude.

5

ILC for LTI systems – A statistical analysis of certain ILC algorithms

5.1 Introduction

In the previous two chapters data-driven precompensator tuning has been considered for LTI and LPV systems in order to improve the system's tracking performance for general movements. In the next three chapters ILC is considered to improve the system's tracking for the specific case where the movement is repetitive.

The aim of this chapter is to compare a number of different existing ILC algorithms for LTI systems, firstly by a statistical analysis, then in simulation and finally by their application to the LPMSM. A recursive formulation for the controlled error is used to develop a new transfer function relationship in the iteration domain. This formulation allows novel expressions for both the mean and variance of the error for each algorithm to be developed.

In [18] analytical expressions for the covariance matrix of the controlled error have been developed for high order ILC algorithms with load and measurement disturbances separately. These expressions together are similar to the complete variance expression derived in this chapter, though here transfer operators are used instead of matrix operators allowing certain insightful frequency-domain expressions to be derived. The contributions differ, also, as here a different

analysis is made including expressions for the mean controlled error and analytical and experimental comparisons of specific algorithms are given.

The algorithms that are compared are:

1. A deterministic algorithm e.g. [4].
2. An algorithm with a forgetting factor e.g. [3, 22].
3. An algorithm with an iteration decreasing gain e.g. [46, 56].
4. A filtered algorithm e.g. [44].

The chapter is organised as follows. In Section 5.2 the ILC problem is formulated and an insight is given into how stochastic, iteration-varying disturbances affect the error convergence. Additionally, the assumptions used during the analysis are given. In Section 5.3 expressions for the mean and variance of the error are developed. In Section 5.4 the expressions are used to analyse the different ILC algorithms. Section 5.5 tests the algorithms in simulation. They are then applied experimentally to the LPMSM in Section 5.6. Finally in Section 5.7 the chapter conclusions are made.

5.2 Problem formulation

In this chapter we consider repetitive processes. The signals are now, therefore, indexed by the repetition number k i.e. the controlled output at repetition k is given by:

$$z(t,k) = G(q^{-1})u(t,k) + d(t,k), \tag{5.1}$$

and the measured output by:

$$y(t,k) = z(t,k) + n(t,k). \tag{5.2}$$

A general form of an ILC command is given by:

$$u(t,k+1) = Q(q^{-1})\left(u(t,k) + q^m L(q^{-1})e(t,k)\right) \tag{5.3}$$

where $Q(q^{-1})$ and $L(q^{-1})$ are linear, discrete, and possibly non-causal, filters, m is the relative degree of $G(q^{-1})$ and the measured error signal is given by:

$$e(t,k) = y_d(t) - y(t,k). \tag{5.4}$$

The desired output $y_d(t)$ is now defined over the finite repetition duration for $t = 0, \ldots, N-1$. The controlled error signal is given as:

$$\epsilon(t,k) = y_d(t) - z(t,k). \tag{5.5}$$

By combining equations (5.1) - (5.5), a recursive expression for the controlled error evolution equation is found as:

$$\begin{aligned}\epsilon(t,k+1) = G(q^{-1})Q(q^{-1})[1 - q^m L(q^{-1})G(q^{-1})]G^{-1}(q^{-1})\epsilon(t,k) \\ + [1 - G(q^{-1})Q(q^{-1})G^{-1}(q^{-1})]y_d(t) \\ + G(q^{-1})Q(q^{-1})G^{-1}(q^{-1})d(t,k) - d(t,k+1) \\ + G(q^{-1})Q(q^{-1})q^m L(q^{-1})n(t,k).\end{aligned} \tag{5.6}$$

It should be noted that in the finite-time setting, causal and non-causal operators do not commute in general. If, however, $Q(q^{-1})$ and $L(q^{-1})$ are causal or N is sufficiently large that the effects of finite time are negligible then (5.6) simplifies to:

$$\begin{aligned}\epsilon(t,k+1) = Q(q^{-1})[1 - G(q^{-1})q^m L(q^{-1})]\epsilon(t,k) + [1 - Q(q^{-1})]y_d(t) \\ + Q(q^{-1})d(t,k) - d(t,k+1) \\ + G(q^{-1})Q(q^{-1})q^m L(q^{-1})n(t,k).\end{aligned} \tag{5.7}$$

It is clear from this expression that, even if $q^m L(q^{-1})$ is chosen such that $q^m L(q^{-1}) = G^{-1}(q^{-1})$, the presence of iteration-varying disturbances and/or the use of a filter $Q(q^{-1}) \neq 1$ mean that a steady converged error value equal to zero is not achievable.

In the absence of disturbances it can be seen that the quickest convergence of the error can be achieved by taking:

$$q^m L(q^{-1}) = G^{-1}(q^{-1}). \tag{5.8}$$

Finding even a good approximation of $G^{-1}(q^{-1})$ is a laborious task, however, and identifying $G^{-1}(q^{-1})$ exactly is impossible. A frequently encountered, sufficient condition for monotonic convergence of the 2-norm of the error between iterations, when $Q(q^{-1})[1 - G(q^{-1})q^m L(q^{-1})]$ is stable and causal, is the following, see e.g. [45]:

$$\sup_{\omega\in[0,\pi]} \left|Q(e^{-j\omega})\left(1 - G(e^{-j\omega})e^{jm\omega}L(e^{-j\omega})\right)\right| < 1. \tag{5.9}$$

It should be noted that, when the repetition length is long enough that $N = \infty$ can be assumed, this condition also assures monotonic convergence of the 2-norm of the controlled error if $Q(q^{-1})[1 - G(q^{-1})q^m L(q^{-1})]$ is stable and non-causal. This fact can be seen using Parseval's theorem.

Often $q^m L(q^{-1})$ is chosen to satisfy condition (5.9) and can be expressed generally as:

$$\begin{aligned} q^m L(q^{-1}) &= \hat{G}^{-1}(q^{-1}) \\ &= \left(1 + \Delta(q^{-1})\right) G^{-1}(q^{-1}), \end{aligned} \tag{5.10}$$

where $\hat{G}(q^{-1})$ is a model of the system and $\Delta(q^{-1})$ represents the multiplicative uncertainty due to the unmodelled dynamics i.e.

$$G(q^{-1}) = [1 + \Delta(q^{-1})]\hat{G}(q^{-1}). \tag{5.11}$$

It should be noted that the direct use of the model inverse is only one choice for $q^m L(q^{-1})$ from a number of approaches proposed in the literature. It will be used in this chapter, however, because many of these other choices can be seen as approximations to the inverse model, each with their own associated $\Delta(q^{-1})$. Moreover it can be obtained using the method presented in Chapter 3 since the precompensator is an approximation of the inverse system.

In [39] a forward-shift iteration-domain operator, w, is defined. It has the property:

$$v(t,k+1) = wv(t,k) \tag{5.12}$$

where $v(t)$ is an arbitrary variable. Using this operator it is possible to rewrite equation (5.7) as:

$$\epsilon(t,k) = \frac{1}{w - Q(q^{-1})[1 - G(q^{-1})q^m L(q^{-1})]}\big([1 - Q(q^{-1})]y_d(t) + [Q(q^{-1}) - w]d(t,k) + G(q^{-1})Q(q^{-1})q^m L(q^{-1})n(t,k)\big). \tag{5.13}$$

This expression will be used later to analyse the effect of disturbances.

5.2.1 Assumptions

In order to make a detailed analysis of the effect of the disturbances on ILC algorithms, it is necessary to first make some assumptions about the nature of the disturbances. In general the load and measurement disturbances can be either iteration repetitive or iteration-varying. Repetitive disturbances will be learnt, in a similar way to the desired output, and thus will not be considered here. The following assumptions are thus made:

A5.1: $d(t,k)$ and $n(t,k)$ are zero-mean, weakly stationary random sequences, which are uncorrelated between iterations and have variances equal to σ_d^2 and σ_n^2 respectively i.e.

$$E\{d(t,k)\} = 0, \qquad E\{n(t,k)\} = 0,$$

$$E\{d(t,k)d(t,k+r)\} = \begin{cases} \sigma_d^2 & r = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.14}$$

$$E\{n(t,k)n(t,k+r)\} = \begin{cases} \sigma_n^2 & r = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.15}$$

and

$$E\{d(t,k)n(t,k+r)\} = \begin{cases} r_{dn}(0) & r=0 \\ 0 & \text{otherwise,} \end{cases} \tag{5.16}$$

where r is a lag.

A5.2: $d(t,k)$ and $n(t,k)$ are assumed to be uncorrelated with the system input $u(t,k)$ i.e.

$$E\{u(t,k)d(t+r,k)\} = 0 \qquad \forall r, \tag{5.17}$$

$$E\{u(t,k)n(t+r,k)\} = 0 \qquad \forall r. \tag{5.18}$$

A5.3: $Q(q^{-1})$ and $G(q^{-1})Q(q^{-1})q^m L(q^{-1})$ are stable and condition (5.9) is satisfied.

5.3 Mean and variance expressions

It is desirable to calculate the statistical properties of an ILC algorithm in order to see how the presence of noise affects the achievable error. To this end, expressions for the mean and variance of the controlled error are given below.

Theorem 5.1 *For the system described by equations (5.1)-(5.2) using the ILC algorithm (5.3), and respecting Assumptions (A5.1)-(A5.3), the converged mean controlled error signal is:*

$$\lim_{k\to\infty} E\{\epsilon(t,k)\} = \frac{[1-Q(q^{-1})]}{1-Q(q^{-1})[1-G(q^{-1})q^m L(q^{-1})]} y_d(t). \tag{5.19}$$

Proof: From equation (5.13) we have:

$$\begin{aligned} E\{\epsilon(t,k)\} =& \frac{1}{w-Q(q^{-1})[1-G(q^{-1})q^m L(q^{-1})]} \Big([1-Q(q^{-1})]E\{y_d(t)\} \\ &+ [Q(q^{-1})-w]E\{d(t,k)\} \\ &+ G(q^{-1})Q(q^{-1})q^m L(q^{-1})E\{n(t,k)\} \Big) \\ =& \frac{[1-Q(q^{-1})]}{w-Q(q^{-1})[1-G(q^{-1})q^m L(q^{-1})]} y_d(t). \end{aligned} \tag{5.20}$$

The z-transform of equation (5.20) in the iteration domain gives:

$$\mathcal{Z}\{E\{\epsilon(t,k)\}\} = \frac{[1-Q(q^{-1})]}{z-Q(q^{-1})[1-G(q^{-1})q^m L(q^{-1})]} \cdot \frac{y_d(t)z}{z-1} \quad (5.21)$$

where $y_d(t)$, being constant from one iteration to the next, acts as a step input at iteration $k=0$. In order to find the value of the error signal as the iteration number tends to infinity it is possible to use the standard Final Value Theorem for discrete-time systems, since Assumption (A5.3) implies that the system's poles in the iteration domain are within the unit circle. This gives:

$$\begin{aligned}\lim_{k\to\infty} E\{\epsilon(t,k)\} &= \lim_{z\to 1} \frac{(z-1)[1-Q(q^{-1})]}{z-Q(q^{-1})[1-G(q^{-1})q^m L(q^{-1})]} \cdot \frac{y_d(t)z}{z-1} \\ &= \frac{[1-Q(q^{-1})]}{1-Q(q^{-1})[1-G(q^{-1})q^m L(q^{-1})]} y_d(t),\end{aligned}$$

which is the expression in Theorem 5.1. ∎

Theorem 5.2 *For the system described by equations (5.1)-(5.2) using the ILC algorithm (5.3), and respecting Assumptions (A5.1)-(A5.3), the variance of the controlled error signal at iteration $k+1$ is:*

$$\begin{aligned}E\{\tilde{\epsilon}^2(t,k+1)\} &= E\{[I(q^{-1})\tilde{\epsilon}(t,k)]^2\} + E\{[Q(q^{-1})d(t,k)]^2\} \\ &+ E\{d^2(t,k+1)\} + E\{[G(q^{-1})Q(q^{-1})q^m L(q^{-1})n(t,k)]^2\} \\ &- 2E\{[I(q^{-1})d(t,k)][Q(q^{-1})d(t,k)]\} \\ &- 2E\{[I(q^{-1})d(t,k)][G(q^{-1})Q(q^{-1})q^m L(q^{-1})n(t,k)]\} \\ &+ 2E\{[Q(q^{-1})d(t,k)][G(q^{-1})Q(q^{-1})q^m L(q^{-1})n(t,k)]\}, \quad (5.22)\end{aligned}$$

where $I(q^{-1}) = Q(q^{-1})[1-G(q^{-1})q^m L(q^{-1})]$.

Proof: The variance of the controlled error at iteration k is defined as:

$$E\{\tilde{\epsilon}^2(t,k)\} = E\{[\epsilon(t,k) - E\{\epsilon(t,k)\}]^2\}. \tag{5.23}$$

Using equations (5.13) and (5.20) we find:

$$\tilde{\epsilon}(t,k) = \frac{[Q(q^{-1}) - w]d(t,k) + G(q^{-1})Q(q^{-1})q^m L(q^{-1})n(t,k)}{w - I(q^{-1})}. \tag{5.24}$$

Expanding equation (5.24) into its recursive form in the iteration domain gives:

$$\tilde{\epsilon}(t,k+1) = I\tilde{\epsilon}(t,k) + Qd(t,k) - d(t,k+1) + GQL_m n(t,k), \tag{5.25}$$

where q^{-1} has been omitted to simplify the presentation and $L_m(q^{-1}) = q^m L(q^{-1})$. Taking the square of equation (5.25) gives:

$$\begin{aligned}\tilde{\epsilon}^2(t,k+1) = {} & [I\tilde{\epsilon}(t,k)]^2 + [Qd(t,k)]^2 + d^2(t,k+1) + [GQL_m n(t,k)]^2 \\ & + 2\big[[I\tilde{\epsilon}(t,k)][Qd(t,k)] - d(t,k+1)[I\tilde{\epsilon}(t,k)] \\ & + [I\tilde{\epsilon}(t,k)][GQL_m n(t,k)] - d(t,k+1)[Qd(t,k)] \\ & + [Qd(t,k)][GQL_m n(t,k)] - d(t,k+1)[GQL_m n(t,k)]\big],\end{aligned} \tag{5.26}$$

which on applying the expectation operator leads to:

$$\begin{aligned}E\{\tilde{\epsilon}^2(t,k+1)\} = {} & E\{[I\tilde{\epsilon}(t,k)]^2\} + E\{[Qd(t,k)]^2\} + E\{d^2(t,k+1)\} \\ & + E\{[GQL_m n(t,k)]^2\} + 2\big[E\{[I\tilde{\epsilon}(t,k)][Qd(t,k)]\} \\ & + E\{[I\tilde{\epsilon}(t,k)][GQL_m n(t,k)]\} + E\{[Qd(t,k)][GQL_m n(t,k)]\}\big],\end{aligned} \tag{5.27}$$

where certain cross terms are lost as they are uncorrelated. The other cross terms can be further evaluated. For example, $E\{[I\tilde{\epsilon}(t,k)][Qd(t,k)]\}$ can be found by filtering equation (5.25), evaluated at iteration k, by $I(q^{-1})$ and then multiplying by $Q(q^{-1})d(t,k)$ and taking the expected value, this gives:

$$\begin{aligned}E\{[I\tilde{\epsilon}(t,k)][Qd(t,k)]\} = {} & E\{[I^2\tilde{\epsilon}(t,k-1)][Qd(t,k)]\} \\ & + E\{[IQd(t,k-1)][Qd(t,k)]\} - E\{[Id(t,k)][Qd(t,k)]\} \\ & + E\{[IGQL_m n(t,k-1)][Qd(t,k)]\} \\ = {} & -E\{[Id(t,k)][Qd(t,k)]\}.\end{aligned} \tag{5.28}$$

Similar manipulations give:

$$E\{[I\tilde{\epsilon}(t,k)][GQL_m n(t,k)]\} = -E\{[Id(t,k)][GQL_m n(t,k)]\}. \quad (5.29)$$

Substituting these results into equation (5.27) gives the expression in Theorem 5.2. ∎

Remarks:

1. It is clear from Theorem 5.1, as remarked earlier, that the use of a filter $Q(q^{-1}) \neq 1$ does not allow a zero-mean controlled error value to be reached.
2. Using the fact that the variance can be expressed as:

$$E\{\tilde{\epsilon}^2(t,k)\} = E\{\epsilon^2(t,k)\} - E\{\epsilon(t,k)\}^2, \quad (5.30)$$

we get that:

$$E\{\epsilon^2(t,k)\} = E\{\tilde{\epsilon}^2(t,k)\} + E\{\epsilon(t,k)\}^2. \quad (5.31)$$

The RMS value of the controlled error is then defined as:

$$\sqrt{E\left\{\frac{1}{N}\sum_{t=0}^{N-1}\epsilon^2(t,k)\right\}} = \sqrt{\frac{1}{N}\sum_{t=0}^{N-1}[E\{\tilde{\epsilon}^2(t,k)\} + E\{\epsilon(t,k)\}^2]}. \quad (5.32)$$

This illustrates that a small RMS value is only achievable when both the variance and the expected value of the controlled error are small, thus motivating the need for an analysis of these quantities.

5.4 Analysis of the algorithms

In this section the different algorithms are analysed using the expressions developed in the previous section. Throughout the section the assumption of perfect system knowledge is made because, for this case, the expressions can be intuitively interpreted. Expressions for the general case are given in the chapter's appendix.

5.4.1 A deterministic algorithm

In the standard case, where the ILC law is designed without consideration for stochastic disturbances, we have $Q(q^{-1}) = 1$ and $q^m L(q^{-1}) = G^{-1}(q^{-1})$. With these choices Theorem 5.1 gives:

$$\lim_{k\to\infty} E\{\epsilon(t,k)\} = 0. \tag{5.33}$$

The mean error, therefore, eventually converges to 0, as desired. However, Theorem 5.2 gives the variance of the error signal as:

$$\begin{aligned} E\{\tilde{\epsilon}^2(t,k+1)\} &= E\left\{d^2(t,k)\right\} + E\left\{d^2(t,k+1)\right\} \\ &\quad + E\{n^2(t,k)\} + 2E\{d(t,k)n(t,k)\} \\ &= 2\sigma_d^2 + \sigma_n^2 + 2r_{dn}(0), \end{aligned} \tag{5.34}$$

where σ_d^2 is the variance of both $d(t,k)$ and $d(t,k+1)$ due to the stationarity assumption made on $d(t,k)$. In general $r_{dn}(0)$ can be positive or negative so it is difficult to comment on it. However, in the case that $d(t,k)$ and $n(t,k)$ are uncorrelated, the variance of the error can be seen to be the sum of twice the variance of the load disturbance and the variance of the measurement disturbance. The fact that the load disturbance's variance is doubled corresponds to results found in [56] and [14], though in both only a single perturbation source is considered. It demonstrates how the presence of non-repetitive disturbances can be particularly detrimental to the tracking performance achievable using ILC as they are fed back from the previous iteration.

Remark: If, instead of taking $q^m L(q^{-1})$ as above, it is taken as $q^m L(q^{-1}) = \kappa G^{-1}(q^{-1})$ with $0 < \kappa < 1$, and $Q(q^{-1}) = 1$ still, the mean error converges to zero in the limit but the variance of the error is now:

$$\begin{aligned} E\{\tilde{\epsilon}^2(t,k+1)\} &= (1-\kappa)^2 E\{\tilde{\epsilon}^2(t,k)\} - (1-2\kappa)E\{d^2(t,k)\} \\ &\quad + E\{d^2(t,k+1)\} + \kappa^2 E\{n^2(t,k)\} + 2\kappa^2 r_{dn}(0). \end{aligned} \tag{5.35}$$

Since $(1-\kappa)^2 < 1$,

$$\lim_{k\to\infty} E\{\tilde{\epsilon}^2(t,k+1)\} = E\{\tilde{\epsilon}^2(t,k)\} = E\{\tilde{\epsilon}^2(t,\infty)\}, \tag{5.36}$$

where

$$E\{\tilde{\epsilon}^2(t,\infty)\} = \frac{2}{(2-\kappa)}\sigma_d^2 + \frac{\kappa}{(2-\kappa)}\sigma_n^2 + \frac{2\kappa}{(2-\kappa)}r_{nd}(0). \tag{5.37}$$

So, considering the case where $d(t,k)$ and $n(t,k)$ are uncorrelated, the limit variance of the error can be reduced compared to that of the standard case when $\kappa = 1$. This reduction in the error variance entails, nonetheless, a reduction in the rate of convergence of the deterministic error and thus a compromise must be made. The expression for the component of the error variance due to the load disturbances is in agreement with an expression found in Section 6.2 of [18].

5.4.2 An algorithm with a forgetting factor

An ILC law with a forgetting factor is given by:

$$u(t,k+1) = (1-\alpha)u(t,k) + q^m L'(q^{-1})e(t,k), \tag{5.38}$$

where $0 \leq \alpha < 1$ is the forgetting factor. The objective of introducing the forgetting factor is to increase the learning algorithm's robustness to initialisation errors, fluctuations of the dynamics and random disturbances. The reasoning behind this is that when considering the input at the k'th iteration the previous inputs will be multiplied by $(1-\alpha)^k$. Thus when α is chosen such that $1-\alpha < 1$ their influence on the current input should be diminished, and so will that of the disturbances from previous iterations that are fed back in the inputs. The law (5.38) will now be investigated using the framework presented above.

By comparing the ILC laws (5.3) and (5.38), it is possible to see that $Q(q^{-1}) = 1-\alpha$ and $q^m L(q^{-1}) = \frac{q^m L'(q^{-1})}{1-\alpha}$. If $q^m L'(q^{-1}) = G^{-1}(q^{-1})$ is taken, Theorem 5.1 gives:

$$\lim_{k\to\infty} E\{\epsilon(t,k)\} = \frac{\alpha}{\alpha+1} y_d(t). \tag{5.39}$$

So when $\alpha \neq 0$ the mean error cannot converge to zero. Examining the variance, Theorem 5.2 gives:

$$\begin{aligned} E\{\tilde{\epsilon}^2(t,k+1)\} =& \alpha^2 E\{\tilde{\epsilon}^2(t,k)\} + (1-\alpha)^2 E\{d^2(t,k)\} \\ &+ E\{d^2(t,k+1)\} + E\{n^2(t,k)\} \\ &+ 2\alpha(1-\alpha)E\{d^2(t,k)\} + 2\alpha E\{d(t,k)n(t,k)\} \\ &+ 2(1-\alpha)E\{d(t,k)n(t,k)\}, \end{aligned} \tag{5.40}$$

which in the limit $k \to \infty$ reduces to:

$$E\{\tilde{\epsilon}^2(t,\infty)\} = \sigma_d^2 + \frac{1}{1-\alpha^2}(\sigma_d^2 + \sigma_n^2 + 2r_{dn}(0)). \tag{5.41}$$

Thus, for the case that $d(t,k)$ and $n(t,k)$ are uncorrelated, the forgetting factor that minimises the limit variance of the error can be seen to be $\alpha = 0$ i.e. no forgetting factor should be used, giving the deterministic algorithm. This result is similar to a result found in [50], which uses a different analysis framework but concludes that the optimal forgetting matrix is zero when the trace of the input error covariance matrix is to be minimised.

It is interesting to note that if $q^m L(q^{-1}) = q^m L'(q^{-1}) = G^{-1}(q^{-1})$ is used instead and $Q(q^{-1})$ is as before, i.e. the forgetting factor affects the entire algorithm, a different conclusion is reached. In this case Theorem 5.1 gives:

$$\lim_{k\to\infty} E\{\epsilon(t,k)\} = \alpha y_d(t). \tag{5.42}$$

So again we see that in order to minimise $\lim_{k\to\infty} E\{\epsilon(t,k)\}$ the optimal α is $\alpha = 0$. But Theorem 5.2 gives:

$$\begin{aligned} E\{\tilde{\epsilon}^2(t,k+1)\} &= E\left\{d^2(t,k+1)\right\} \\ &\quad + (1-\alpha)^2\left(E\left\{d^2(t,k)\right\} + E\left\{n^2(t,k)\right\}\right) \\ &= \sigma_d^2 + (1-\alpha)^2\left(\sigma_d^2 + \sigma_n^2 + 2r_{dn}(0)\right). \end{aligned} \tag{5.43}$$

This time the optimal value of α to minimise the variance of the error, when the $d(t,k)$ and $n(t,k)$ are uncorrelated, is $\alpha = 1$ and leads to $E\{\tilde{\epsilon}^2(t,k+1)\} = \sigma_d^2$. This value makes sense as it means that the previous input is not fed back at all so only the load disturbance during the current iteration affects the error. A compromise therefore needs to be made between minimising the variance of the error and keeping its converged, mean value small.

5.4.3 An algorithm with an iteration decreasing gain

An ILC command with an iteration decreasing learning gain has the form:

$$u(t,k+1) = u(t,k) + \frac{q^m L'(q^{-1})}{k+1} e(t,k). \tag{5.44}$$

With this law $Q(q^{-1}) = 1$ and $q^m L(q^{-1}) = \frac{q^m L'(q^{-1})}{k+1}$ and Theorem 5.1 gives:

$$\lim_{k\to\infty} E\{\epsilon(t,k)\} = 0, \tag{5.45}$$

and Theorem 5.2, with $q^m L'(q^{-1}) = G^{-1}(q^{-1})$, gives:

$$E\{\tilde{\epsilon}^2(t,k+1)\} = \frac{1}{k+1}\left(\sigma_n^2 + r_{dn}(0)\right) + \frac{1+2/k}{1+1/k}\sigma_d^2. \tag{5.46}$$

In the limit (5.46) gives:

$$\lim_{k\to\infty} E\{\tilde{\epsilon}^2(t,k+1)\} = \sigma_d^2. \tag{5.47}$$

Results (5.45) and (5.47) show that, when a decreasing learning gain is used, the mean error converges to zero and the variance of the error converges to just that of the load disturbance. This performance is the best that can be achieved. The disadvantage of this algorithm is that the error contraction rate reduces with iteration number and eventually the learning practically stops so it cannot react to slow changes in the desired output or the repetitive disturbances affecting the system.

5.4.4 A filtered algorithm

A filtered ILC law is that given by the general form (5.3):

$$u(t,k+1) = Q(q^{-1})[u(t,k) + q^m L(q^{-1})e(t,k)],$$

where $Q(q^{-1})$ is the filter referred to. When $q^m L(q^{-1}) = G^{-1}(q^{-1})$ is used, Theorem 5.1 gives:

$$\lim_{k\to\infty} E\{\epsilon(t,k)\} = [1 - Q(q^{-1})]y_d(t). \tag{5.48}$$

Since ILC is defined over a finite time duration, N, the infinite-time Fourier transform cannot be applied in order to work in the frequency domain. However, when N is large compared to the settling time of $G(q^{-1})$, the finite-time Fourier transform can be used to make a reasonably accurate frequency-domain analysis. The magnitude response of the Fourier transform of equation (5.48) is:

$$\left|\mathcal{F}\left\{\lim_{k\to\infty} E\{\epsilon(t,k)\}\right\}\right|^2 = |1 - Q(e^{-j\omega})|^2 \Phi_{y_d}(\omega).$$

It is clear that in order to converge to a zero-mean error it is necessary to use a filter that has a magnitude of 1 and zero phase shift at frequencies where $\Phi_{y_d}(\omega)$ is non-zero.

Now considering the error variance, Theorem 5.2 gives:

$$\begin{aligned} E\{\tilde{\epsilon}^2(t,k+1)\} = \sigma_d^2 + E\left\{[Q(q^{-1})d(t,k)]^2\right\} + E\left\{[Q(q^{-1})n(t,k)]^2\right\} \\ + 2E\{[Q(q^{-1})d(t,k)][Q(q^{-1})n(t,k)]\}. \end{aligned} \tag{5.49}$$

Since $d(t,k)$ and $n(t,k)$ are stationary so is $\epsilon(t,k+1)$, therefore, (5.49) can also be expressed as:

$$\begin{aligned} E\{\tilde{\epsilon}^2(t,k+1)\} = \sigma_d^2 + \frac{1}{2\pi}\int_{-\pi}^{\pi} |Q(e^{-j\omega})|^2 [\Phi_d(\omega) + \Phi_n(\omega) \\ + 2\Phi_{nd}(\omega)]d\omega. \end{aligned} \tag{5.50}$$

When $\Phi_{nd}(\omega) = 0$, it is possible to see that the variance of the error can be reduced below that obtained using the deterministic algorithm by choosing $Q(q^{-1})$ to have a magnitude of less than one at frequencies where the disturbance power spectra are large.

A compromise, therefore, needs to be made again, between filtering in order to reduce the error variance, but not filtering at frequencies important to $y_d(t)$ so as to allow a reasonable converged mean error to be achieved. Fortunately low-frequency signals are often used for $y_d(t)$, whilst $\Phi_d(\omega)$ and $\Phi_n(\omega)$ tend to be large at high frequencies. This means that the minimisation of the converged error and the error variance are usually not conflicting aims, and $Q(q^{-1})$ can be taken as a low-pass filter with a sensibly chosen cut-off frequency.

5.5 Simulation results

A simulation is carried out to illustrate the theoretical results. The real, continuous-time system, $G_c(s)$, and its identified model, $\hat{G}_c(s)$, are:

$$G_c(s) = \frac{20}{(s+1)(s+20)} \qquad \text{and} \qquad \hat{G}_c(s) = \frac{1}{(s+1)}. \tag{5.51}$$

$G_c(s)$ and $\hat{G}_c(s)$ are discretised using a sampling period of $h = 0.1$s and a zero-order hold to give the discrete-time systems $G(q^{-1})$ and $\hat{G}(q^{-1})$ respectively. $y_d(t)$ is defined by:

$$y_d(t) = \begin{cases} 1 - \cos(0.1\pi t) & 0 \le t \le 20\text{s} \\ 0 & 20.1 \le t \le 30\text{s} \end{cases} \tag{5.52}$$

Using this $y_d(t)$ and the specified sampling period give $N = 301$. The load disturbance, $d(t,k)$, is taken as a zero-mean, normally distributed, random sequence with $\sigma_d^2 = 0.05^2$. The measurement disturbance, $n(t,k)$, is taken as a zero-mean, normally distributed, random sequence with variance equal to 0.05^2 filtered by a 5th order

Butterworth high-pass filter with a cut-off frequency of 2Hz, to simulate high frequency measurement noise. It has an estimated variance of 0.0387^2. The different algorithms analysed in the previous section are tested. 10 iterations are carried out for each algorithm and, in order to obtain an estimate of the mean and variance of the error at a specific time, each simulation is repeated 200 times. The expected value and variance at $t = 15$s, and the RMS value are estimated for the error at $k = 10$ for the 200 simulations. The iteration number is chosen to allow the algorithm to have converged to a point where the errors due to the disturbances are dominant over the deterministic errors. The time $t = 15$s is chosen arbitrarily. Although the disturbances affecting the system are different for the 200 simulations, the same disturbance signals are used for each of the different algorithms. This means a direct comparison can be made of how each algorithm performs in the presence of the same disturbances.

Table 5.1 shows the simulation results for the different algorithms. Certain clarifications are perhaps necessary. The algorithm with the forgetting factor is tested using different values for the forgetting factor. Additionally, the different ways of implementing the forgetting factor are tested, firstly when the factor only affects the previous input i.e. $q^m L(q^{-1}) = \frac{q^m L'(q^{-1})}{1-\alpha}$, and secondly when it affects the entire algorithm i.e. $q^m L(q^{-1}) = q^m L'(q^{-1})$. Two versions of the filtered algorithm are tested, a causal and a non-causal one. The non-causal version can be justified by making the assumption that the repetition length is sufficiently long that the effects of finite time are negligible. Both versions use a 5th order Butterworth low-pass filter with a cut-off frequency of 0.3Hz for $Q(q^{-1})$. The filter's cut-off frequency is chosen to be above that of the highest frequency component of $y_d(t)$, which is at 0.05Hz. The non-causal version is implemented by first filtering the ILC command normally, then filtering the result in reverse-time, before flipping the new result so that it is in forward-time again. These operations mean the filtered signal has zero phase change compared to the initial signal.

Algorithm	$\left\lvert\hat{E}\{\epsilon(15,10)\}\right\rvert$	$\hat{E}\{\tilde{\epsilon}^2(15,10)\}$	$\sqrt{\hat{E}\left\{\frac{1}{N}\sum_{t=0}^{N-1}\epsilon^2(t,10)\right\}}$
Deterministic	0.0065	0.0050	0.0683
Forgetting factor ($\alpha = 0.1$)			
$L = \frac{L'}{1-\alpha}$	0.0987	0.0048	0.1124
$L = L'$	0.1081	0.0044	0.1184
Forgetting factor ($\alpha = 0.5$)			
$L = \frac{L'}{1-\alpha}$	0.3474	0.0048	0.3395
$L = L'$	0.5177	0.0032	0.5018
Forgetting factor ($\alpha = 0.9$)			
$L = \frac{L'}{1-\alpha}$	0.8184	0.0071	0.6461
$L = L'$	0.9317	0.0027	0.8996
Decreasing gain	0.0266	0.0030	0.0557
Low-pass filter			
- Causal	0.5113	0.0029	0.3141
- Noncausal	0.0045	0.0027	0.0513

Table 5.1. Estimated mean and variance of $\epsilon(15, 10)$ and the RMS values of $\epsilon(t, 10)$ found over 200 simulations

From the table the compromise between minimising the variance of the error whilst keeping its mean value small is clearly seen. The deterministic algorithm has the second smallest mean error value but the second largest error variance. The algorithms using the forgetting factor show the tendencies anticipated from the theoretical results. In the case of the decreasing gain, we see that the variance is much smaller than that achieved with the deterministic algorithm, however, the mean value of the error is not so small. This is because, as previously mentioned, the learning rate reduces with each iteration so the error has not converged as quickly as that of the determinis-

tic algorithm. The algorithm using the causal filter produces a low error variance, as predicted by the analysis, but a large mean error value due to the phase shift introduced by the filter. The non-causal filtered algorithm has the lowest mean error and joint lowest variance, it also has the smallest RMS value. This good performance is because the $y_d(t)$ is limited to low frequencies so allows a low-pass filter to effectively filter out the disturbances at higher frequencies. Additionally, the non-causal implementation means that it does not suffer from the phase-shift problems of the causal filter.

5.6 Experimental results

The different ILC algorithms are applied to the tracking control of the LPMSM. The two-degree-of-freedom controller operating at a sampling frequency of 2kHz is used. The sampling frequency is imposed by the way ILC is implemented on the system.

The input, $u(t,k)$, computed by the ILC algorithms is used as the reference signal of the closed-loop system, meaning the transfer function $G(q^{-1})$ represents the closed-loop motor system. The desired output position, $y_d(t)$, is a series of three Macromotions in the positive direction, followed by a similar series in the negative direction, as can be seen in Figure 5.1 and has $N = 8192$.

An approximate Box-Jenkins model of the closed-loop motor system is identified using standard identification techniques and a PRBS as an input signal. The PRBS is created using a shift register of 10 bits and the resulting signal is repeated eight times giving a total length of 8184 points. The identified model is:

$$\hat{G}(q^{-1}) = \frac{q^{-1}\left(0.00201 + 0.00092q^{-1} + 0.01972q^{-2} - 0.00938q^{-3}\right)}{1 - 3.104q^{-1} + 3.739q^{-2} - 2.058q^{-3} + 0.4364q^{-4}}. \tag{5.53}$$

It is used to calculate a phase lead compensator as an approximation to $\hat{G}^{-1}(q^{-1})$ [33]. This gives $\hat{G}^{-1}(q^{-1}) = 0.85q^6$, which satisfied condition (5.9) up to a frequency of 424Hz, with $Q(q^{-1}) = 1$, see

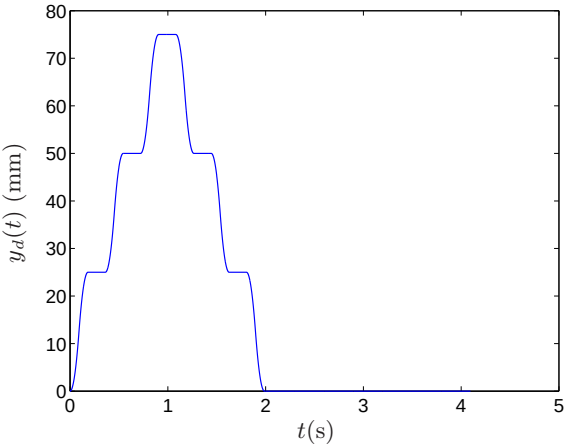

Fig. 5.1. Desired output position

Figure 5.2. To assure monotonic convergence it is thus necessary to take $Q(q^{-1})$ as a low-pass filter with a cut-off frequency below 424Hz, 400Hz was chosen. The filter is implemented in a non-causal way so as to give zero-phase shift. A fifth order Butterworth filter is used. Despite the phase lead compensator not being an accurate inverse of the system model it allows reasonably rapid convergence to be attained without going through the laborious process of very accurate modelling. Because a low-pass filter is necessary for deterministic convergence the standard, deterministic algorithm is not implementable experimentally. The other algorithms, however, are implemented, all being filtered. For each experiment 100 iterations are carried out and each experiment is repeated four times. It is, obviously, not possible to measure $\epsilon(t,k)$ in real experiments so the measured error $e(t,k)$ is used for comparisons. For all experiments $u(t,0) = y_d(t)$ is taken.

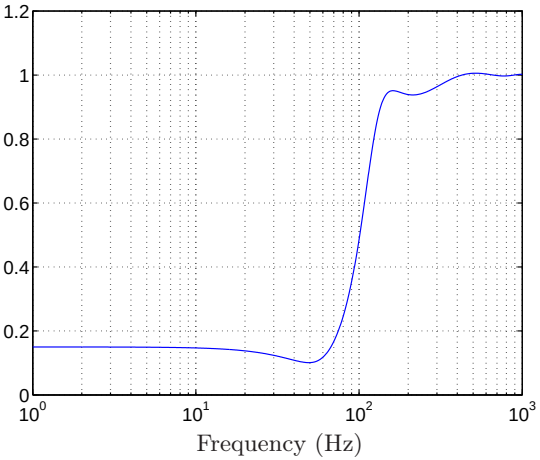

Fig. 5.2. $\left|Q(e^{-j\omega h})\left(1 - G(e^{-j\omega h})\hat{G}^{-1}(e^{-j\omega h})\right)\right|$ with $Q(q^{-1}) = 1$ and $\hat{G}^{-1}(q^{-1}) = 0.85q^6$

For the forgetting factor algorithms a forgetting factor of $\alpha = 10^{-6}$ is used. This value is chosen in order to allow a reasonable RMS value of the converged measured error, $e(t, \infty)$, to be obtained. Expressions (5.32), (5.39) and (5.42) are used in the noiseless case i.e. $e(t, k) = \epsilon(t, k)$ to find lower bounds on the RMS value of $e(t, \infty)$. The desired RMS value of $e(t, \infty)$ is taken as that achieved with the low-pass filtered algorithm. The forgetting factor is so small that $L' \approx \frac{L'}{1-\alpha}$, meaning the two variations become, essentially, the same.

Table 5.2 shows the RMS values of $e(t, 100)$ obtained with the different algorithms. It is seen that the value achieved by the decreasing gain algorithm is about 1.4 times greater than that achieved with the filtered version and the forgetting factor algorithm gives a value approximately five times larger.

Figure 5.3 shows the mean tracking performance at $k = 100$ for the forgetting factor algorithm for a small section of the trajectory.

Algorithm	$\sqrt{\hat{E}\left\{\frac{1}{N}\sum_{t=0}^{N-1} e^2(t,100)\right\}}$ (mm)
Low-pass filter	7.4257×10^{-5}
Forgetting factor	3.8603×10^{-4}
Decreasing gain	1.0673×10^{-4}

Table 5.2. RMS values for $e(t, 100)$ found over 4 experiments

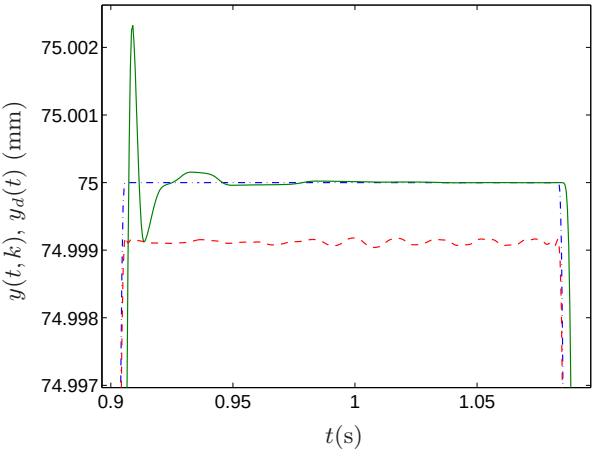

Fig. 5.3. Mean system output for the forgetting factor algorithm at $k = 0$ (solid) and $k = 100$ (dashed), and $y_d(t)$ (dot-dash)

It is clear that even with the small α used a constant error still occurs, explaining the much larger RMS value. Figure 5.4 shows a zoom for the other two algorithms. The decreasing gain algorithm has slightly more oscillation in the overshoot region due to its slower learning of the deterministic errors. This is probably the cause of its larger RMS value. However, it does give less oscillatory tracking in the steady-state region. This, perhaps, is because noise exists at

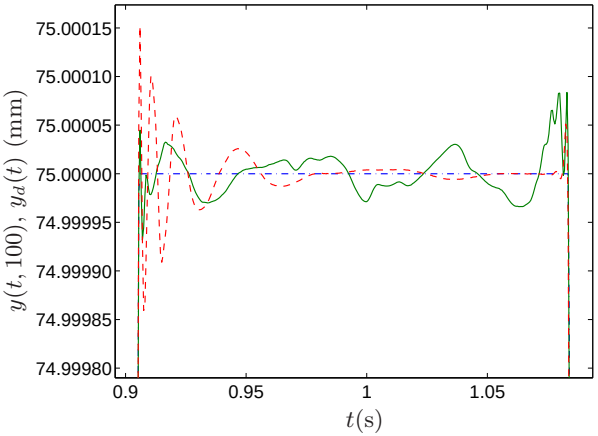

Fig. 5.4. Mean system output at $k = 100$ for the algorithms: decreasing gain (dashed) and low-pass filtered (solid), and $y_d(t)$ (dot-dash)

frequencies below the filter's cut-off frequency so the decreasing gain algorithm helps reduce its detrimental effect. Reducing the filter cut-off frequency would help reduce the algorithms sensitivity to noise but would cause a larger expected error to occur for this $y_d(t)$.

These observations are confirmed in Table 5.3, where the mean and variance of the error for a certain time in this region are smaller for the decreasing gain algorithm.

5.7 Conclusions

In this chapter expressions for the mean and variance of the controlled tracking error in the presence of stochastic disturbances have been developed in general, and for several specific ILC algorithms.

It has been shown that a trade off between minimising the mean and variance of the error commonly occurs. This can be recognised

Algorithm	$\left\lvert \hat{E}\{e(1,100)\} \right\rvert$ (mm)	$\hat{E}\{\tilde{e}^2(1,100)\}$ (mm^2)
Low-pass filter	1.3672×10^{-5}	9.1445×10^{-10}
Forgetting factor	8.6169×10^{-4}	2.7705×10^{-9}
Decreasing gain	1.4038×10^{-7}	6.1343×10^{-12}

Table 5.3. Estimated mean and variance of $e(1, 100)$ over 4 experiments

as the well-known bias-variance tradeoff, which commonly occurs in estimation theory. When the spectra of the noise and desired output are situated in different frequency regions it has been found that a filtered algorithm can give good tracking performance. If the spectra overlap too much, however, an algorithm with a decreasing learning gain has been shown to give good robustness to noise and small tracking errors, although it has a slower error convergence rate.

Appendix

5.A Derivation of expression (5.6)

The controlled error at iteration $k+1$ is given by:

$$\begin{aligned}\epsilon(t,k+1) =& y_d(t) - G(q^{-1})u(t,k+1) - d(t,k+1)\\ =& y_d(t) - G(q^{-1})Q(q^{-1})\left[u(t,k) + q^m L(q^{-1})e(t,k)\right]\\ & - d(t,k+1)\\ =& G(q^{-1})Q(q^{-1})[q^m L(q^{-1})G(q^{-1}) - 1]u(t,k)\\ & + [1 - G(q^{-1})Q(q^{-1})q^m L(q^{-1})]y_d(t) - d(t,k+1)\\ & + G(q^{-1})Q(q^{-1})q^m L(q^{-1})[d(t,k) + n(t,k)]. \quad (5.54)\end{aligned}$$

This expression gives:

$$\begin{aligned}u(t,k) = &\left[G(q^{-1})Q(q^{-1})[q^m L(q^{-1})G(q^{-1}) - 1]\right]^{-1}[\epsilon(t,k+1)\\ & - [1 - G(q^{-1})Q(q^{-1})q^m L(q^{-1})]y_d(t) + d(t,k+1)\\ & - G(q^{-1})Q(q^{-1})q^m L(q^{-1})[d(t,k) + n(t,k)]]. \quad (5.55)\end{aligned}$$

We also have that:

$$u(t,k) = G^{-1}(q^{-1})[y_d(t,k) - \epsilon(t,k) - d(t,k)]. \quad (5.56)$$

Equating (5.55) and (5.56) gives:

$$\begin{aligned}&G(q^{-1})Q(q^{-1})[q^m L(q^{-1})G(q^{-1}) - 1]G^{-1}(q^{-1})[y_d(t,k) - \epsilon(t,k)\\ &- d(t,k)] = \epsilon(t,k+1) - [1 - G(q^{-1})Q(q^{-1})q^m L(q^{-1})]y_d(t)\\ &+ d(t,k+1) - G(q^{-1})Q(q^{-1})q^m L(q^{-1})[d(t,k) + n(t,k)], \quad (5.57)\end{aligned}$$

from which we have:

$$\begin{aligned}\epsilon(t,k+1) = & G(q^{-1})Q(q^{-1})[1 - q^m L(q^{-1})G(q^{-1})]G^{-1}(q^{-1})\epsilon(t,k)\\ & + [1 - G(q^{-1})Q(q^{-1})G^{-1}(q^{-1})]y_d(t) - d(t,k+1)\\ & + G(q^{-1})Q(q^{-1})G^{-1}(q^{-1})d(t,k) + G(q^{-1})Q(q^{-1})q^m L(q^{-1})n(t,k),\end{aligned}$$

which is equal to (5.6).

5.B Derivation of expression (5.46)

Theorem 5.2, with $Q(q^{-1}) = 1$ and $q^m L(q^{-1}) = \frac{G^{-1}(q^{-1})}{k+1}$, gives:

$$
\begin{aligned}
&E\{\tilde{\epsilon}^2(t,k+1)\} \\
=&\frac{k^2}{(k+1)^2}E\{\tilde{\epsilon}^2(t,k)\} + E\{d^2(t,k)\} + E\{d^2(t,k+1)\} \\
&\quad + \frac{1}{(k+1)^2}E\left\{n^2(t,k)\right\} - 2\left(1-\frac{1}{k+1}\right)E\left\{d^2(t,k)\right\} \\
&\quad - 2\left(\left(1-\frac{1}{k+1}\right)\frac{1}{k+1} - \frac{1}{k+1}\right)E\left\{d(t,k)n(t,k)\right\} \\
=&\frac{k^2}{(k+1)^2}E\{\tilde{\epsilon}^2(t,k)\} + \frac{1}{(k+1)^2}\left(\sigma_n^2 + 2r_{dn}(0)\right) + \frac{2(k+1)}{(k+1)^2}\sigma_d^2 \\
=&\frac{k^2}{(k+1)^2}\left[\frac{(k-1)^2}{k^2}E\{\tilde{\epsilon}^2(t,k-1)\} + \frac{1}{k^2}\left(\sigma_n^2 + 2r_{dn}(0)\right) + \frac{2k}{k^2}\sigma_d^2\right] \\
&\quad + \frac{1}{(k+1)^2}\left(\sigma_n^2 + 2r_{dn}(0)\right) + \frac{2(k+1)}{(k+1)^2}\sigma_d^2 \\
=&\frac{k^2}{(k+1)^2}\left[\frac{(k-1)^2}{k^2}\left[\frac{(k-2)^2}{(k-1)^2}E\{\tilde{\epsilon}^2(t,k-2)\}\right.\right. \\
&\quad \left.+\frac{1}{(k-1)^2}\left(\sigma_n^2 + 2r_{dn}(0)\right) + \frac{2(k-1)}{(k-1)^2}\sigma_d^2\right] \\
&\quad \left.+ \frac{1}{k^2}\left(\sigma_n^2 + 2r_{dn}(0)\right) + \frac{2k}{k^2}\sigma_d^2\right] \\
&\quad + \frac{1}{(k+1)^2}\left(\sigma_n^2 + 2r_{dn}(0)\right) + \frac{2(k+1)}{(k+1)^2}\sigma_d^2
\end{aligned}
$$

$$
\begin{aligned}
=&\frac{k^2}{(k+1)^2}\frac{(k-1)^2}{k^2}\frac{(k-2)^2}{(k-1)^2}\cdots\frac{0^2}{1^2}E\{\tilde{\epsilon}^2(t,0)\}\\
&+\left[\frac{k^2}{(k+1)^2}\frac{(k-1)^2}{k^2}\frac{(k-2)^2}{(k-1)^2}\cdots\frac{1}{1^2}+\cdots\right.\\
&\qquad+\frac{k^2}{(k+1)^2}\frac{(k-1)^2}{k^2}\frac{1}{(k-1)^2}+\frac{k^2}{(k+1)^2}\frac{1}{k^2}+\cdots\\
&\qquad\left.+\frac{1}{(k+1)^2}\right]\left(\sigma_n^2+2r_{dn}(0)\right)\\
&+\left[\frac{k^2}{(k+1)^2}\frac{(k-1)^2}{k^2}\frac{(k-2)^2}{(k-1)^2}\cdots\frac{2}{1^2}+\cdots\right.\\
&+\frac{k^2}{(k+1)^2}\frac{(k-1)^2}{k^2}\frac{2(k-1)}{(k-1)^2}+\frac{k^2}{(k+1)^2}\frac{2k}{k^2}+\cdots+\left.\frac{2(k+1)}{(k+1)^2}\right]\sigma_d^2\\
=&(k+1)\frac{1}{(k+1)^2}\left(\sigma_n^2+r_{dn}(0)\right)+[1+\cdots+k+(k+1)]\frac{2}{(k+1)^2}\sigma_d^2\\
=&\frac{1}{k+1}\left(\sigma_n^2+r_{dn}(0)\right)+\frac{(k+1)(k+2)}{2}\frac{2}{(k+1)^2}\sigma_d^2\\
=&\frac{1}{k+1}\left(\sigma_n^2+r_{dn}(0)\right)+\frac{1+2/k}{1+1/k}\sigma_d^2.
\end{aligned}
$$

5.C Full mean and variance expressions

Here the expressions for the controlled error's expected value and variance, in the case of uncertain system knowledge, are given.

5.C.1 A deterministic algorithm

We have $Q(q^{-1}) = 1$ and $q^m L(q^{-1}) = [1 + \Delta(q^{-1})]G^{-1}(q^{-1})$, so Theorem 5.1 gives:

$$\lim_{k\to\infty} E\{\epsilon(t,k)\} = 0 \tag{5.58}$$

and Theorem 5.2 gives:

$$E\{\tilde{\epsilon}^2(t,k{+}1)\} = E\{[\Delta(q^{-1})\tilde{\epsilon}(t,k)]^2\}{+}E\{d^2(t,k)\}{+}E\{d^2(t,k{+}1)\}$$
$$+ E\{[(1+\Delta(q^{-1}))n(t,k)]^2\} + 2E\{d(t,k)[\Delta(q^{-1})d(t,k)]\}$$
$$+ 2E\{[\Delta(q^{-1})d(t,k)][(1+\Delta(q^{-1}))n(t,k)]\}$$
$$+ 2E\{d(t,k)[(1+\Delta(q^{-1}))n(t,k)]\}. \quad (5.59)$$

5.C.2 An algorithm with a forgetting factor

For the first variant of the forgetting factor algorithm we have $Q(q^{-1}) = 1-\alpha$ and $q^m L(q^{-1}) = \frac{q^m L'(q^{-1})}{1-\alpha}$. When $q^m L'(q^{-1}) = [1+\Delta(q^{-1})]G^{-1}(q^{-1})$ is used, Theorem 5.1 gives:

$$\lim_{k\to\infty} E\{\epsilon(t,k)\} = \frac{\alpha}{\alpha + [1+\Delta(q^{-1})]} y_d(t) \quad (5.60)$$

and from Theorem 5.2 we have:

$$E\{\tilde{\epsilon}^2(t,k+1)\} = E\{[(\alpha+\Delta(q^{-1}))\tilde{\epsilon}(t,k)]^2\} + (1-\alpha)^2 E\{d^2(t,k)\}$$
$$+ E\{d^2(t,k+1)\} + E\{[(1+\Delta(q^{-1}))n(t,k)]^2\}$$
$$+ 2(1-\alpha)E\{d(t,k)[(\alpha+\Delta(q^{-1}))d(t,k)]\}$$
$$+ 2E\{[(\alpha+\Delta(q^{-1}))d(t,k)][(1+\Delta(q^{-1}))n(t,k)]\}$$
$$+ 2(1-\alpha)E\{d(t,k)[(1+\Delta(q^{-1}))n(t,k)]\}. \quad (5.61)$$

For the alternative forgetting factor algorithm $q^m L'(q^{-1}) = q^m L(q^{-1}) = [1+\Delta(q^{-1})]G^{-1}(q^{-1})$ and $Q(q^{-1}) = 1$, and Theorem 5.1 gives:

$$\lim_{k\to\infty} E\{\epsilon(t,k)\} = \frac{\alpha}{1+(1-\alpha)\Delta(q^{-1})} y_d(t), \quad (5.62)$$

and Theorem 5.2 leads to:

$$E\{\tilde{\epsilon}^2(t,k+1)\} = (1-\alpha)^2 E\{[\Delta(q^{-1})\tilde{\epsilon}(t,k)]^2\} + (1-\alpha)^2 E\{d^2(t,k)\} + E\{d^2(t,k+1)\} + (1-\alpha)^2 E\{[(1+\Delta(q^{-1}))n(t,k)]^2\} + 2(1-\alpha)^2 E\{d(t,k)[\Delta(q^{-1})d(t,k)]\} + 2(1-\alpha)^2 E\{[\Delta(q^{-1})d(t,k)][(1+\Delta(q^{-1}))n(t,k)]\} + 2(1-\alpha)^2 E\{d(t,k)[(1+\Delta(q^{-1}))n(t,k)]\}. \quad (5.63)$$

5.C.3 An algorithm with an iteration decreasing gain

For the ILC command with an iteration decreasing learning gain we have $Q(q^{-1}) = 1$ and $q^m L(q^{-1}) = \frac{[1+\Delta(q^{-1})]G^{-1}(q^{-1})}{k+1}$ thus Theorem 5.1 leads:

$$\lim_{k\to\infty} E\{\epsilon(t,k)\} = 0, \quad (5.64)$$

and Theorem 5.2:

$$E\{\tilde{\epsilon}^2(t,k+1)\} = \frac{1}{(k+1)^2} E\{[(k-\Delta(q^{-1}))\tilde{\epsilon}(t,k)]^2\} + E\{d^2(t,k)\} + E\{d^2(t,k+1)\} + \frac{1}{(k+1)^2} E\{[(1+\Delta(q^{-1}))n(t,k)]^2\} - \frac{2}{k+1} E\{[(k-\Delta(q^{-1}))d(t,k)]d(t,k)\} - \frac{2}{(k+1)^2} E\{[(k-\Delta(q^{-1}))d(t,k)][(1+\Delta(q^{-1}))n(t,k)]\} + \frac{2}{k+1} E\{d(t,k)[(1+\Delta(q^{-1}))n(t,k)]\}. \quad (5.65)$$

5.C.4 A filtered algorithm

Theorem 5.1 gives, for the filtered ILC law with $q^m L(q^{-1}) = [1 + \Delta(q^{-1})]G^{-1}(q^{-1})$:

$$\lim_{k\to\infty} E\{\epsilon(t,k)\} = \frac{[1-Q(q^{-1})]}{1+Q(q^{-1})\Delta(q^{-1})} y_d(t). \quad (5.66)$$

The frequency domain interpretation is therefore:

$$\left|\mathcal{F}\left\{\lim_{k\to\infty} E\{\epsilon(t,k)\}\right\}\right|^2 = \frac{|1-Q(e^{-j\omega})|^2}{|1+Q(e^{-j\omega})\Delta(e^{-j\omega})|^2}\Phi_{y_d}(\omega).$$

Theorem 5.2 gives the variance as:

$$\begin{aligned}E\{\tilde{\epsilon}^2(t,k+1)\} &= E\{[Q(q^{-1})\Delta(q^{-1})\tilde{\epsilon}(t,k)]^2\}+E\{[Q(q^{-1})d(t,k)]^2\}\\ &+ E\{d^2(t,k+1)\}+E\{[Q(q^{-1})[1+\Delta(q^{-1})]n(t,k)]^2\}\\ &+2E\{[Q(q^{-1})\Delta(q^{-1})d(t,k)][Q(q^{-1})d(t,k)]\}\\ &+2E\{[Q(q^{-1})\Delta(q^{-1})d(t,k)][Q(q^{-1})[1+\Delta(q^{-1})]n(t,k)]\}\\ &+2E\{[Q(q^{-1})d(t,k)][Q(q^{-1})[1+\Delta(q^{-1})]n(t,k)]\}. \quad (5.67)\end{aligned}$$

6

ILC for LTI systems – ILC based on stochastic approximation

6.1 Introduction

The main contribution of this chapter is to show how ILC for linear systems affected by stochastic disturbances fits into the stochastic approximation theory framework. Using SA theory it is possible to derive conditions for well-known ILC algorithms to converge almost surely to the optimal input signal in the presence of stochastic disturbances. In addition, the important practical issues of monotonic convergence of the error signal and robustness to system uncertainty are addressed. Also two choices of learning matrix based on an uncertain model are studied, as well as a model-free choice.

In [11] SA theory is used to develop an ILC algorithm for linear systems. There the input produced by the algorithm at each iteration is randomly perturbed and applied to the system in a second experiment in order to estimate the gradient of the proposed cost function. In contrast, here either an uncertain system model or a second special experiment is considered. These choices will typically lead to faster convergence.

Steepest descent algorithms have been applied to ILC for the discrete-time case in [21]. Although certain similarities exist between the algorithms considered in this chapter and steepest descent

algorithms, the major difference is the conditions SA sets on the step sizes between iterations. These conditions are necessary to ensure almost sure convergence to the optimal input in the presence of stochastic disturbances.

This chapter is organised as follows. In Section 6.2 the notational framework used is defined and the assumptions are stated. In Section 6.3 ILC is considered from an SA perspective. Then in Section 6.4 possible choices of the learning matrix are considered. Simulations are carried out in Section 6.5. In Section 6.6 experimental results obtained on the LPMSM are presented. Finally in Section 6.7 the chapter conclusions are made.

6.2 Problem formulation

As the signals in ILC are defined over a finite duration, it is possible to express a discrete-time system's input-output relationship by a matrix representation. Taking advantage of the non-causal filtering possibilities of ILC, the lifted-system representation is used. For a system with a relative degree m we define the vectors:

$$\mathbf{u}(k) = [u(0,k), u(1,k), \ldots, u(N-m-1,k)]^T \tag{6.1}$$

and

$$\mathbf{z}(k) = [z(m,k), z(m+1,k), \ldots, z(N-1,k)]^T. \tag{6.2}$$

The vectors $\mathbf{y}(k)$, $\mathbf{d}(k)$, $\mathbf{n}(k)$ and $\mathbf{y_d}$ are defined similarly to $\mathbf{z}(k)$. Using these vectors, the measured output of the system is:

$$\mathbf{y}(k) = \mathbf{G}\mathbf{u}(k) + \mathbf{d}(k) + \mathbf{n}(k), \tag{6.3}$$

where $\mathbf{G}$ is:

$$\mathbf{G} = \begin{bmatrix} g_m & 0 & \ldots & 0 \\ g_{m+1} & g_m & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_m \end{bmatrix}, \tag{6.4}$$

g_i being the ith Markov parameter of $G(q^{-1})$. The controlled error vector is:

$$\boldsymbol{\epsilon}(k, \mathbf{u}(k)) = \mathbf{y_d} - \mathbf{z}(k) = \mathbf{y_d} - \mathbf{G}\mathbf{u}(k) - \mathbf{d}(k) \tag{6.5}$$

and the measured error vector:

$$\mathbf{e}(k, \mathbf{u}(k)) = \mathbf{y_d} - \mathbf{y}(k) = \boldsymbol{\epsilon}(k, \mathbf{u}(k)) - \mathbf{n}(k), \tag{6.6}$$

where the errors' dependence on $\mathbf{u}(k)$ is explicitly stated.

Furthermore, we have that the real system can be represented in lifted-system form as:

$$\mathbf{G} = \hat{\mathbf{G}}[\mathbf{I} + \boldsymbol{\Delta}] \tag{6.7}$$

where $\mathbf{I}$ is the identity matrix, and $\hat{\mathbf{G}}$ and $\mathbf{I} + \boldsymbol{\Delta}$ are Toeplitz matrices formed similarly to (6.4) from the Markov parameters of $q^{\hat{m}}\hat{G}(q^{-1})$ and $q^{m-\hat{m}}[1 + \Delta(q^{-1})]$, respectively. $\hat{m}$ is the relative degree of $\hat{G}(q^{-1})$.

Definition: A real, square matrix $\mathbf{M}$ (not necessarily symmetric) is called positive definite $\mathbf{M} > 0$ if and only if all the eigenvalues of its symmetric part $(\mathbf{M} + \mathbf{M}^T)/2$ are positive.

6.2.1 Assumptions

A6.1: The ideal input:

$$\mathbf{u}_0 = \mathbf{G}^{-1}\mathbf{y_d} \tag{6.8}$$

is realisable.

A6.2: The system uncertainty satisfies:

$$\mathbf{I} + \boldsymbol{\Delta} > 0. \tag{6.9}$$

A6.3: The disturbances $\mathbf{d}(k)$ and $\mathbf{n}(k)$ are zero-mean, weakly stationary random vectors with bounded, unknown covariance matrices $\mathbf{R_d}$ and $\mathbf{R_n}$, respectively. Additionally, they have bounded, unknown cross-covariance matrices $\mathbf{R_{dn}}$ and $\mathbf{R_{nd}}$. Moreover, different realisations of $\mathbf{d}(k)$ and $\mathbf{n}(k)$ between iterations are mutually independent.

A6.4: The mean input is bounded for all iterations:

$$E\{\mathbf{u}(k)\} < \infty \quad \forall k.$$

Remarks:

1) It is shown in [20] that a sufficient condition for (6.9) is that the filter $q^{m-\hat{m}}[1 + \Delta(q^{-1})]$ is strictly positive real (SPR). So when $m = \hat{m}$, Assumption A6.2 is satisfied when $\|\Delta\|_\infty < 1$. This condition occurs frequently in the model uncertainty representation and so is a reasonable assumption.
2) The validity of Assumption A6.4 will be discussed later in the chapter.

6.3 ILC from a stochastic approximation viewpoint

The ideal aim of tracking control is to achieve zero controlled error. When stochastic disturbances affect a system this objective is not possible. A reasonable aim, in this case, is to set the mean controlled error equal to zero. We can state a goal of an ILC algorithm, thus, as to iteratively calculate the optimal input signal $\mathbf{u}_0$ such that:

$$E\{\mathbf{L}\boldsymbol{\epsilon}(k, \mathbf{u}_0)\} = E\{\mathbf{L}\mathbf{e}(k, \mathbf{u}_0)\} = 0, \tag{6.10}$$

where $\mathbf{L} \in \mathbb{R}^{(N-m)\times(N-m)}$ is a non-singular matrix.

It is straightforward to see that the solution to criterion (6.10) is (6.8). However, in order to calculate the ideal input $\mathbf{u}_0$ directly exact knowledge of $\mathbf{G}$ is needed, which is not available. Nevertheless, $\mathbf{u}_0$ can be found using an iterative SA procedure, such as the Robbins-Monro algorithm, which does not require exact system knowledge. This algorithm calculates the input iteratively as:

$$\mathbf{u}(k+1) = \mathbf{u}(k) + \gamma(k)\mathbf{L}\mathbf{e}(k, \mathbf{u}(k)). \tag{6.11}$$

This algorithm clearly has the form of a standard P-type ILC law with an iteration varying learning gain $\gamma(k)$. In the next subsection conditions will be given that, according to SA theory, ensure almost sure convergence of the algorithm to the ideal input.

6.3.1 Almost sure convergence

Theorem 6.1 *Under the Assumptions A6.1, A6.3 and A6.4, the iterative update algorithm (6.11) converges almost surely to the solution* $\mathbf{u}_0$ *of (6.10) when* $k \to \infty$ *if:*

i) The sequence $\gamma(k)$ *of positive steps satisfies:*

$$\sum_{k=0}^{\infty} \gamma(k) = \infty \quad \textit{and} \quad \sum_{k=0}^{\infty} \gamma^2(k) < \infty. \tag{6.12}$$

ii) $E\{\mathbf{Le}(k, \mathbf{u}(k))\}$ *is monotonically decreasing:*

$$\mathbf{Q}(\mathbf{u}(k)) = \frac{d}{d\mathbf{u}(k)} E\{\mathbf{Le}(k, \mathbf{u}(k))\} < 0. \tag{6.13}$$

Proof: The proof follows by applying Theorem 2.1. The conditions on the step sizes in (6.12) are those in Condition i of Theorem 2.1. Setting $c(\mathbf{u}) = E\{\mathbf{Le}(k, \mathbf{u})\}^T E\{\mathbf{Le}(k, \mathbf{u})\}$ automatically satisfies Condition ii. Conditions iii and iv follow from (6.13). Condition v is satisfied by Assumptions A6.1, A6.3 and A6.4. ■

Condition (6.12) should be fulfilled by an appropriate choice of the sequence $\gamma(k)$. $\mathbf{Q}(\mathbf{u}(k))$, in Condition (6.13), can be rewritten as:

$$\begin{aligned} \mathbf{Q}(\mathbf{u}(k)) &= \frac{d}{d\mathbf{u}(k)} E\{\mathbf{Le}(k, \mathbf{u}(k))\} \\ &= \frac{d}{d\mathbf{u}(k)} \; E\{\mathbf{Ly_d} - \mathbf{LGu}(k) + \mathbf{Ld}(k) + \mathbf{Lv}(k)\} \\ &= -\mathbf{LG} = -\mathbf{L}\hat{\mathbf{G}}[\mathbf{I} + \mathbf{\Delta}] \end{aligned} \tag{6.14}$$

and so Condition (6.13) becomes:

$$\mathbf{L}\hat{\mathbf{G}}[\mathbf{I}+\mathbf{\Delta}] > 0. \tag{6.15}$$

Remark: By combining equations (6.5), (6.6), (6.8) and (6.11) we can obtain the input error evolution as:

$$\begin{aligned}\mathbf{e}_u(k+1) &= \mathbf{u}_0 - \mathbf{u}(k+1)\\ &= (\mathbf{I}-\gamma(k)\mathbf{LG})\mathbf{e}_u(k) + \gamma(k)\mathbf{L}(\mathbf{y_d} - \mathbf{d}(\mathbf{k}) - \mathbf{n}(\mathbf{k})).\end{aligned} \tag{6.16}$$

A necessary, but not sufficient, condition for asymptotic convergence of the input error, in the absence of disturbances, is:

$$|\lambda_i(\mathbf{I}-\gamma(k)\mathbf{LG})| < 1 \quad \forall k, \forall i \tag{6.17}$$

where $\lambda_i(\cdot)$ is the i^{th} eigenvalue. If $\mathbf{L}$ represents a causal operator and is therefore a real, lower triangular matrix, a link between this condition and those given by SA theory can be made, as detailed below. Since $\mathbf{I}-\gamma(k)\mathbf{LG}$ will be a real, lower triangular matrix, its eigenvalues will be real. (6.17) therefore implies:

$$\begin{aligned}\overline{\lambda}(\mathbf{I}-\gamma(k)\mathbf{LG}) < 1 &\iff 1-\gamma(k)\underline{\lambda}(\mathbf{LG}) < 1\\ &\iff \gamma(k)\underline{\lambda}(\mathbf{LG}) > 0\end{aligned} \tag{6.18}$$

and

$$\begin{aligned}\underline{\lambda}(\mathbf{I}-\gamma(k)\mathbf{LG}) > -1 &\iff 1-\overline{\lambda}(\gamma(k)\mathbf{LG}) > -1\\ &\iff \gamma(k)\overline{\lambda}(\mathbf{LG}) < 2,\end{aligned} \tag{6.19}$$

where $\underline{\lambda}(\cdot)$ and $\overline{\lambda}(\cdot)$ are the minimum and maximum eigenvalues, respectively. Moreover we can write:

$$\mathbf{LG}\mathbf{x}_i = \lambda_i \mathbf{x}_i, \tag{6.20}$$

where $\mathbf{x}_i$ is the real eigenvector corresponding to λ_i. Taking the transpose of (6.20) gives:

$$\mathbf{x}_i^T(\mathbf{LG})^T = \lambda_i \mathbf{x}_i^T. \tag{6.21}$$

Left multiplying (6.20) by $\mathbf{x}_i^T$, right multiplying (6.21) by $\mathbf{x}_i$ and adding gives:

$$\begin{aligned} &\mathbf{x}_i^T\mathbf{LG}\mathbf{x}_i + \mathbf{x}_i^T(\mathbf{LG})^T\mathbf{x}_i = 2\lambda_i\mathbf{x}_i^T\mathbf{x}_i \\ \iff &\mathbf{x}_i^T\left(\frac{\mathbf{LG}+(\mathbf{LG})^T}{2}\right)\mathbf{x}_i = \lambda_i\mathbf{x}_i^T\mathbf{x}_i. \end{aligned} \tag{6.22}$$

So, if $\mathbf{LG}$ is positive definite, (6.18) is satisfied. (6.19) can be satisfied by an appropriate choice of $\gamma(k)$.

6.3.2 Monotonic convergence

Whilst almost sure convergence of the input sequence to the solution $\mathbf{u}_0$ when $k \to \infty$ is, obviously, of utmost importance, practically it is not the only type of convergence of interest. The monotonic convergence, from one iteration to the next, of a norm of the controlled error is also of concern.

To proceed, we will need the following lemma:

Lemma 6.1 *If a real, square matrix* $\mathbf{M}$ *(not necessarily symmetric) is positive definite, there exists an* $\alpha > 0$ *such that:*

$$\overline{\sigma}(\mathbf{I} - \alpha\mathbf{M}) < 1, \tag{6.23}$$

where $\overline{\sigma}(\cdot)$ *is the maximum singular value.*

Proof: Condition (6.23) is true iff:

$$\begin{aligned} &\lambda_i\left(I - \alpha(\mathbf{M}^T + \mathbf{M}) + \alpha^2\mathbf{M}^T\mathbf{M}\right) < 1 \quad \forall i \\ \iff &1 - \lambda_i\left(\alpha(\mathbf{M}^T + \mathbf{M}) - \alpha^2\mathbf{M}^T\mathbf{M}\right) < 1 \quad \forall i \\ \iff &\lambda_i\left(\mathbf{M}^T + \mathbf{M} - \alpha\mathbf{M}^T\mathbf{M}\right) > 0 \quad \forall i. \end{aligned} \tag{6.24}$$

Furthermore the eigenvalues satisfy:

$$[\mathbf{M}^T + \mathbf{M} - \alpha\mathbf{M}^T\mathbf{M}]\mathbf{x}_i = \lambda_i\mathbf{x}_i. \tag{6.25}$$

Left multiplying (6.25) by $\mathbf{x}_i^T$ we get:

$$\mathbf{x}_i^T \left(\mathbf{M}^T + \mathbf{M}\right) \mathbf{x}_i - \alpha \mathbf{x}_i^T \mathbf{M}^T \mathbf{M} \mathbf{x}_i = \lambda_i \mathbf{x}_i^T \mathbf{x}_i. \tag{6.26}$$

So if $\mathbf{M} > 0$, (6.24), and thus Condition (6.23), are satisfied when:

$$0 < \alpha < \min_i \frac{\mathbf{x}_i^T \left(\mathbf{M}^T + \mathbf{M}\right) \mathbf{x}_i}{\mathbf{x}_i^T \mathbf{M}^T \mathbf{M} \mathbf{x}_i}. \tag{6.27}$$

■

Theorem 6.2 *If $\hat{\mathbf{G}}[\mathbf{I} + \mathbf{\Delta}]\mathbf{L} > 0$, there exists a sequence of positive step sizes $\gamma(k)$, satisfying Condition (6.12), such that monotonic convergence of the 2-norm of the mean controlled error is achieved.*

Proof: By combining equations (6.5), (6.6), (6.7) and (6.11) we can obtain the controlled error evolution equation as:

$$\begin{aligned} \boldsymbol{\epsilon}(k+1, \mathbf{u}(k+1)) = (\mathbf{I} - \gamma(k)\hat{\mathbf{G}}[\mathbf{I} + \mathbf{\Delta}]\mathbf{L})\boldsymbol{\epsilon}(k, \mathbf{u}(k)) + \mathbf{d}(k) \\ - \mathbf{d}(k+1) + \gamma(k)\hat{\mathbf{G}}[\mathbf{I} + \mathbf{\Delta}]\mathbf{L}\mathbf{n}(k). \end{aligned} \tag{6.28}$$

The mean value of equation (6.28) is:

$$E\{\boldsymbol{\epsilon}(k+1, \mathbf{u}(k+1))\} = (\mathbf{I} - \gamma(k)\hat{\mathbf{G}}[\mathbf{I} + \mathbf{\Delta}]\mathbf{L}) E\{\boldsymbol{\epsilon}(k, \mathbf{u}(k))\}. \tag{6.29}$$

Monotonic convergence of the 2-norm of the mean controlled error is obtained if the following condition is satisfied (see e.g. Theorem 2, [45]):

$$\overline{\sigma}(\mathbf{I} - \gamma(k)\hat{\mathbf{G}}[\mathbf{I} + \mathbf{\Delta}]\mathbf{L}) < 1 \quad \forall k. \tag{6.30}$$

If a given sequence $\gamma(k)$, satisfying Condition (6.12), does not satisfy (6.30), a new, scaled sequence $\gamma(k) \triangleq \kappa\gamma(k)$, $\kappa > 0$ can always be defined that does, as follows from Lemma 6.1. ■

Remarks:

1) Theorem 6.2's requirement that $\hat{\mathbf{G}}[\mathbf{I} + \mathbf{\Delta}]\mathbf{L}$ be positive definite is satisfied when $\mathbf{L}$ and $\hat{\mathbf{G}}[\mathbf{I} + \mathbf{\Delta}]$ commute, i.e. when $L(q^{-1})$ is causal, and condition (6.15) is satisfied.

2) As shown in the proof of Theorem 6.2, in order to achieve monotonic convergence of the 2-norm of the mean controlled error condition (6.30) should be satisfied. This can be achieved by reducing $\max_k \gamma(k)$.

6.3.3 Boundedness of the system's signals

Since the system $G(q^{-1})$ is assumed stable, its output and internal states will be bounded if its input is bounded. Combining equations (6.5), (6.6), (6.7) and (6.11) gives the input evolution equation as:

$$\mathbf{u}(k+1) = (\mathbf{I}-\gamma(k)\mathbf{L}\hat{\mathbf{G}}[\mathbf{I}+\boldsymbol{\Delta}])\mathbf{u}(k)+\gamma(k)\mathbf{L}(\mathbf{y_d}-\mathbf{d}(\mathbf{k})-\mathbf{n}(\mathbf{k})). \quad (6.31)$$

According to Theorem 5 of [45] the input will remain bounded from one iteration to the next if a) (6.31) is a uniformly exponentially stable iterative system, b) for a finite constant κ, $\|\gamma(k)\mathbf{L}\| < \kappa \ \forall k$, and c) $\mathbf{y_d}$, $\mathbf{d}(\mathbf{k})$ and $\mathbf{n}(\mathbf{k})$ are bounded. As stated in Corollary 1 of [45], (6.31) is a uniformly exponentially stable iterative system if $\overline{\sigma}(\mathbf{I} - \gamma(k)\mathbf{L}\hat{\mathbf{G}}[\mathbf{I} + \boldsymbol{\Delta}]) < 1 \quad \forall k$. This condition is considered in Lemma 6.1, implying that, when $\mathbf{L}\hat{\mathbf{G}}[\mathbf{I} + \boldsymbol{\Delta}] > 0$, a sequence $\gamma(k)$ exists that achieves uniform exponential stability. Furthermore, since $\|\gamma(k)\mathbf{L}\| = |\gamma(k)|\|\mathbf{L}\|$, there exists a sequence $\gamma(k)$ that satisfies the condition $\|\gamma(k)\mathbf{L}\| < \kappa \ \forall k$. So the boundedness of the system's signals requires the disturbances to be bounded, which can usually be assumed to be the case in practice.

It should be noted that the mean input $E\{\mathbf{u}(k)\}$ will be bounded if only the means of the disturbances are bounded, rather than the disturbances themselves.

6.3.4 Asymptotic distribution of the input error

The asymptotic distribution of the input error is given by the following theorem:

Theorem 6.3 *Assume that:*

i) Algorithm (6.11) converges almost surely to the solution $\mathbf{u}_0$ *as* $k \to \infty$.
ii) The sequence of step sizes is chosen as $\gamma(k) = \frac{\alpha}{k+1}$.
iii) All the eigenvalues of the matrix $\mathbf{D} = \mathbf{I}/2 + \alpha\mathbf{Q}(\mathbf{u}_0)$ *have negative real parts.*

Then the sequence $\sqrt{k}(\mathbf{u}(k) - \mathbf{u}_0) \in As\, \mathcal{N}(\mathbf{0}, \mathbf{V})$ *i.e it converges asymptotically in distribution to a zero-mean normal distribution with covariance:*

$$\mathbf{V} = \alpha^2 \int_0^\infty \exp(\mathbf{D}\mathbf{x})\mathbf{P}\exp(\mathbf{D}^T\mathbf{x})d\mathbf{x} \tag{6.32}$$

where $\mathbf{P}$ *is the covariance matrix of* $\mathbf{Le}(\mathbf{u}_0)$*:*

$$\mathbf{P} = E\{\mathbf{Le}(k, \mathbf{u}_0)(\mathbf{Le}(k, \mathbf{u}_0))^T\}. \tag{6.33}$$

Proof: The proof follows directly by applying Theorem 2.2. ■

Using Theorem 6.3 we have that:

$$\begin{aligned}\mathbf{P} &= E\{\mathbf{Le}(k, \mathbf{u}_0)(\mathbf{Le}(k, \mathbf{u}_0))^T\} \\ &= E\{(-\mathbf{L}(\mathbf{d}(k) + \mathbf{n}(k)))(-\mathbf{L}(\mathbf{d}(k) + \mathbf{n}(k)))^T\} \\ &= \mathbf{L}(\mathbf{R_d} + \mathbf{R_{dn}} + \mathbf{R_{nd}} + \mathbf{R_n})\mathbf{L}^T. \end{aligned} \tag{6.34}$$

Additionally, as $\mathbf{Q}(\mathbf{u}_0) = \frac{d}{d\mathbf{u}(k)} E\{\mathbf{Le}(k, \mathbf{u}(k))\}\big|_{\mathbf{u}(k)=\mathbf{u}_0} = -\mathbf{LG}$, we have that:

$$\mathbf{D} = (\mathbf{I}/2 - \alpha\mathbf{LG}). \tag{6.35}$$

The covariance matrix $\mathbf{V}$ is then the unique symmetric solution of the following Lyapunov equation:

$$\begin{aligned}2\alpha^2\mathbf{L}(\mathbf{R_d} + \mathbf{R_{dn}} + \mathbf{R_{nd}} + \mathbf{R_n})\mathbf{L}^T + (\mathbf{I} - 2\alpha\mathbf{LG})\mathbf{V} \\ + \mathbf{V}(\mathbf{I} - 2\alpha\mathbf{LG})^T = \mathbf{0}.\end{aligned} \tag{6.36}$$

It is shown in [6] (Proposition 4, p.112) that if, instead of using a scalar learning gain α, we use a non-singular learning matrix $\mathbf{K}$, then the optimal matrix $\mathbf{K}^*$ to mimimise the trace of $\mathbf{V}$ is given by:

$$\mathbf{K}^* = -\mathbf{Q}(\mathbf{u}_0)^{-1} = (\mathbf{LG})^{-1}. \tag{6.37}$$

Using this gain matrix results in the learning law:

$$\mathbf{u}(k+1) = \mathbf{u}(k) + \frac{\mathbf{G}^{-1}}{k+1}\mathbf{e}(k, \mathbf{u}(k)), \tag{6.38}$$

and the optimal asymptotic covariance matrix:

$$\mathbf{V}^* = \mathbf{G}^{-1}(\mathbf{R_d} + \mathbf{R_{dn}} + \mathbf{R_{nd}} + \mathbf{R_n})\mathbf{G}^{-T}, \tag{6.39}$$

which means that the sequence $\sqrt{k}(\mathbf{u}(k) - \mathbf{u}_0) \in \text{As}\,\mathcal{N}(\mathbf{0}, \mathbf{V}^*)$.

Moreover we have that $\boldsymbol{\epsilon}(k, \mathbf{u}(k)) = -\mathbf{G}(\mathbf{u}(k) - \mathbf{u}_0) - \mathbf{d}(k)$ so the covariance matrix of $\boldsymbol{\epsilon}(k, \mathbf{u}(k))$ is then given by:

$$\begin{aligned}\text{cov}(\boldsymbol{\epsilon}(k,\mathbf{u}(k))) &= E\{\boldsymbol{\epsilon}(k, \mathbf{u}(k))\boldsymbol{\epsilon}^T(k, \mathbf{u}(k))\} \\ &= \mathbf{G}E\{(\mathbf{u}(k) - \mathbf{u}_0)(\mathbf{u}(k) - \mathbf{u}_0)^T\}\mathbf{G}^T + \mathbf{R_d}. \end{aligned} \tag{6.40}$$

Using the optimal gain matrix $\mathbf{K}^*$ means that the sequence $\boldsymbol{\epsilon}(k, \mathbf{u}(k))$ will have a converged covariance matrix given by $\text{cov}(\boldsymbol{\epsilon}(k, \mathbf{u}(k))) = \frac{1}{k}(\mathbf{R_d} + \mathbf{R_{dn}} + \mathbf{R_{nd}} + \mathbf{R_n}) + \mathbf{R_d}$ and in the limit we have:

$$\lim_{k\to\infty} \text{cov}(\boldsymbol{\epsilon}(k, \mathbf{u}(k))) = \mathbf{R_d}. \tag{6.41}$$

$\mathbf{K}^*$ is, however, not implementable because exact knowledge of $\mathbf{G}$ is not achievable. Nonetheless it gives an ideal law to aim for in the design of a stochastic ILC algorithm.

Remark: Result (6.41) is similar to that derived in Chapter 5 for the algorithm with an iteration decreasing gain. This similitude clearly occurs because the algorithm there is the same as that considered here. It is, however, presented, and the result derived, under different frameworks in the two chapters.

6.4 Specific choices of the learning matrix

In this section specific choices of the learning matrix $\mathbf{L}$ will be considered.

6.4.1 Use of the uncertain system inverse

We consider here the choice of $\mathbf{L} = \hat{\mathbf{G}}^{-1}$ i.e. the inverse of the uncertain system model. This choice is motivated by the fact that $\mathbf{L} = \hat{\mathbf{G}}^{-1}$ is an approximation of the optimal learning gain used in (6.38).

Theorem 6.4 *Under Assumption A6.2 and when* $\mathbf{L} = \hat{\mathbf{G}}^{-1}$*, there exists a sequence of positive step sizes* $\gamma(k)$*, satisfying Condition (6.12), that ensures that the ILC algorithm (6.11) converges almost surely to* $\mathbf{u}_0$ *and that the 2-norm of the mean controlled error convergences monotonically.*

Proof: Condition (6.15) is automatically satisfied when $\mathbf{L} = \hat{\mathbf{G}}^{-1}$, under Assumption A6.2. Therefore, when the sequence of positive step sizes $\gamma(k)$ satisfies Condition (6.12), the ILC algorithm (6.11) converges almost surely to $\mathbf{u}_0$, as stated by Theorem 6.1. Moreover, because $\mathbf{I}+\boldsymbol{\Delta}$ is a lower triangular Toeplitz matrix, $\mathbf{I}+\boldsymbol{\Delta}$ commutes with $\hat{\mathbf{G}}$ and, under Assumption A6.2, $\hat{\mathbf{G}}[\mathbf{I} + \boldsymbol{\Delta}]\mathbf{L} > 0$. This result means Theorem 6.2 applies, implying the existence of a sequence, satisfying Condition (6.12), that ensures monotonic convergence. ■

6.4.2 Use of the uncertain system transpose

Another choice is $\mathbf{L} = \hat{\mathbf{G}}^T$. This choice is motivated by the fact that it can be used when $\hat{\mathbf{G}}$ is ill conditioned, as may be the case when $\hat{G}(q^{-1})$ has unstable zeros. The previously considered choice of $\mathbf{L}$, on the other hand, may not be usable because the input signal generated by the ILC algorithm can grow unacceptably large before converging to the ideal input.

Theorem 6.5 *Under Assumption A6.2 and when* $\mathbf{L} = \hat{\mathbf{G}}^T$*, there exists a sequence of positive step sizes* $\gamma(k)$*, satisfying Condition (6.12), that ensures that the ILC algorithm (6.11) converges almost surely to* $\mathbf{u}_0$ *and that the 2-norm of the mean controlled error convergences monotonically.*

Proof: Since $\mathbf{I}+\mathbf{\Delta}$ is a lower triangular Toeplitz matrix, $\mathbf{I}+\mathbf{\Delta}$ commutes with $\hat{\mathbf{G}}$ and condition (6.15) can be written as $\hat{\mathbf{G}}^T[\mathbf{I}+\mathbf{\Delta}]\hat{\mathbf{G}} > 0$, when $\mathbf{L} = \hat{\mathbf{G}}^T$. This condition is fulfilled when $\hat{\mathbf{G}}$ is non-singular and $\mathbf{I}+\mathbf{\Delta} > 0$. The former is true because N is finite and the latter is Assumption A6.2. Therefore, when the sequence of positive step sizes $\gamma(k)$ satisfies Condition (6.12), the ILC algorithm (6.11) converges almost surely to $\mathbf{u}_0$, as stated by Theorem 6.1. Moreover, Theorem 6.2 applies, implying the existence of a sequence, satisfying Condition (6.12), that ensures monotonic convergence. ■

6.4.3 Use of an experiment

So far the use of a model to give an $\mathbf{L}$ that can then be used in (6.11) to evaluate $\mathbf{Le}(k, \mathbf{u}(k))$ has been considered. For the specific choice of $\mathbf{L} = \mathbf{G}^T$, it is, however, possible to use an extra experiment per iteration to evaluate $\mathbf{Le}(k, \mathbf{u}(k))$. Condition (6.15) is automatically satisfied with this choice, and Theorem 6.2 also applies.

The fact that a special experiment can be used is seen by noting that $\mathbf{e2} = \mathbf{G}^T\mathbf{e}(k, \mathbf{u}(k))$ is equal to the following filtering operations:

$$e1(t) = G(q^{-1})e(N-t, k, \mathbf{u}(k)) \tag{6.42}$$

$$e2(t) = e1(N-t). \tag{6.43}$$

We see that, in the disturbance free case, $\mathbf{e2}$ can be found using an experiment on the true system, where the time reversed error signal is fed into the system as its input, the system output is measured and then time reversed itself. In reality the special experiment will have its own disturbances $d2(t)$ and $v2(t)$ associated with it. Nonetheless, an unbiased estimate of $\mathbf{e2}$ can still be found since:

$$\begin{aligned} E\{\mathbf{e2}\} &= E\{\mathbf{G}^T\mathbf{e}(k, \mathbf{u}(k)) + \mathbf{d2} + \mathbf{v2}\} \\ &= E\{\mathbf{G}^T\mathbf{e}(k, \mathbf{u}(k))\} + E\{\mathbf{d2}\} + E\{\mathbf{n2}\} \\ &= \mathbf{G}^T E\{\boldsymbol{\epsilon}(k, \mathbf{u}(k))\} + \mathbf{0} + \mathbf{0}. \end{aligned} \tag{6.44}$$

This method of evaluating **e2** is attractive as it avoids the problems of model uncertainty. It does, however, require an additional, non-standard, experiment at each iteration, which, depending on the application, may not always be possible. One case where it may be useful is when ILC is used to tune the input to improve the system's performance before the system is used in its intended application.

Remarks:

1) So far the motivation of the ILC algorithms considered has been to find the input that solves the root-finding type criterion (6.10), which aims to set the mean controlled error to zero. The model-free algorithm can be motivated differently. Instead of criterion (6.10), a logical alternative objective is the minimisation of the trace of the controlled error covariance matrix i.e.:

$$\min_{\mathbf{u}(k)} J_{SA}(k,\mathbf{u}(k)) = \min_{\mathbf{u}(k)} \frac{1}{2}\mathrm{tr}\left(E\{\boldsymbol{\epsilon}(k,\mathbf{u}(k))\boldsymbol{\epsilon}^T(k,\mathbf{u}(k))\}\right). \quad (6.45)$$

The minimum of this criterion occurs when:

$$\begin{aligned}\frac{dJ_{SA}(k,\mathbf{u}(k))}{d\mathbf{u}(k)}\Big|_{\mathbf{u}(k)=\mathbf{u}^*} &= E\left\{\left(\frac{\partial\boldsymbol{\epsilon}(k,\mathbf{u}(k))}{\partial\mathbf{u}(k)}\Big|_{\mathbf{u}(k)=\mathbf{u}^*}\right)^T\boldsymbol{\epsilon}(k,\mathbf{u}^*)\right\}\\ &= -\mathbf{G}^T E\{\boldsymbol{\epsilon}(k,\mathbf{u}^*)\} = \mathbf{0}. \quad (6.46)\end{aligned}$$

$E\{\boldsymbol{\epsilon}(k,\mathbf{u}(k))\}$ is not directly measurable. Nonetheless, because equation (6.46) can be written as:

$$\begin{aligned}\frac{dJ_{SA}(k,\mathbf{u}(k))}{d\mathbf{u}(k)}\Big|_{\mathbf{u}(k)=\mathbf{u}^*} &= -\mathbf{G}^T E\{\boldsymbol{\epsilon}(k,\mathbf{u}^*)\}\\ &= -\mathbf{G}^T E\{\mathbf{e}(k,\mathbf{u}^*)\} = \mathbf{0} \quad (6.47)\end{aligned}$$

it is possible to find the minimiser of the criterion, again, using the Robbins-Monro algorithm:

$$\mathbf{u}(k+1) = \mathbf{u}(k) + \gamma(k)\mathbf{G}^T\mathbf{e}(k,\mathbf{u}(k)), \quad (6.48)$$

i.e. (6.11) with $\mathbf{L} = \mathbf{G}^T$.

2) The model-free algorithm has similarities to that proposed in [65] where reversed time inputs are used to cancel the system phase and produce monotonic convergence. Stochastic aspects are not considered there, however.
It also has similarities to [21], which uses the steepest descent method, and calls $\mathbf{G}^T$ the adjoint of $\mathbf{G}$. It shows that by using this 'adjoint' with an iteration-varying gain, monotonic convergence occurs. The gain sequence is calculated via an optimisation, which does not consider stochastic disturbances. The gain at iteration k is given by:

$$\gamma(k) = \frac{\|\mathbf{G}^T\mathbf{e}(k-1)\|^2}{w_\gamma + \|\mathbf{G}\mathbf{G}^T\mathbf{e}(k-1)\|^2}, \tag{6.49}$$

where w_γ is a weight on $\gamma(k)$ in the cost function. Since the measured error signal is used to calculate the gain, it will be affected by stochastic disturbances. This means $\lim_{k\to\infty} \|\mathbf{G}^T\mathbf{e}(k-1)\|^2 \neq 0$ and so $\lim_{k\to\infty} \gamma(k) \neq 0$. This implies that the second series in (6.12) cannot be satisfied. Therefore, whilst the algorithm developed can lead to fast deterministic convergence to the optimal input, this cannot be proved when stochastic disturbances are present.

6.5 Simulation results

Simulations are carried out to illustrate the theoretical results. The specific choices of the learning matrix $\mathbf{L}$ discussed in Section 6.4 are tested. A continuous-time system, $G_c(s)$, and its identified model, $\hat{G}_c(s)$, are taken as:

$$G_c(s) = \frac{1}{s^2 + (2 \times 0.7 \times 1)s + 1} \tag{6.50}$$

and

$$\hat{G}_c(s) = \frac{1.2^2}{s^2 + (2 \times 0.8 \times 1.2)s + 1.2^2}. \tag{6.51}$$

$G_c(s)$ and $\hat{G}_c(s)$ are discretised using $h = 0.1$s and a zero-order hold to give the discrete-time systems $G(q^{-1})$ and $\hat{G}(q^{-1})$ respectively. $y_d(t)$ is defined by:

$$y_d(t) = \begin{cases} 1 - \cos(0.1\pi t) & 0 \leq t \leq 20s \\ 0 & 20 < t \leq 30s. \end{cases} \tag{6.52}$$

Using this $y_d(t)$ and the specified sampling period gives $N = 301$. The load disturbance, $d(t,k)$, is taken as a normally distributed, random sequence with $E\{d(t,k)\} = 0$ and $\sigma_d^2 = 0.04^2$. The measurement disturbance, $n(t,k)$, is also taken as a zero-mean, normally distributed, random sequence with variance equal to 0.04^2 but is then filtered with a 5th order Butterworth high-pass filter with a cut-off frequency of 2Hz, to simulate high frequency noise. It has a measured variance of 0.031^2.

20 iterations are carried out for each of the different choices of $\mathbf{L}$. $\mathbf{u}(0) = \mathbf{y}_d$ is used. Each simulation is repeated 200 times. Estimates of both $\frac{1}{N}E\{\boldsymbol{\epsilon}(k,\mathbf{u}(k))\}^T E\{\boldsymbol{\epsilon}(k,\mathbf{u}(k))\}$ and $\frac{1}{N}E\{\boldsymbol{\epsilon}^T(k,\mathbf{u}(k))\boldsymbol{\epsilon}(k,\mathbf{u}(k))\}$ are calculated for iterations $k = 0, 10$ and 20.

Although $\mathbf{d}(k)$ and $\mathbf{n}(k)$ are different for the 200 simulations, the same disturbance signals are used for the different choices, thus making possible a direct comparison of the performance of each $\mathbf{L}$ in the presence of the same disturbances.

The system $G(q^{-1})$ and model $\hat{G}(q^{-1})$ used in the simulation are such that Assumption A6.2 is satisfied, since $\underline{\lambda}(\mathbf{I} + \boldsymbol{\Delta} + (\mathbf{I} + \boldsymbol{\Delta})^T) = 1.0217$. Theorems 6.4 and 6.5 therefore apply. They state that, when $\mathbf{L} = \hat{\mathbf{G}}^{-1}$ or $\mathbf{L} = \hat{\mathbf{G}}^T$ is used, a sequence $\gamma(k)$, satisfying Condition (6.12), exists that ensures that the ILC algorithm (6.11) converges almost surely to $\mathbf{u}_0$ and that the 2-norm of the mean controlled error convergences monotonically.

A sequence satisfying Condition (6.12) is $\gamma(k) = \frac{\alpha}{k+1}$, $\alpha > 0$. As stated in the remarks after Theorem 6.2, monotonic convergence of

the 2-norm of the mean controlled error can be achieved by reducing $\max_k \gamma(k) = \alpha$ in order to satisfy the condition (6.30). For the system and model used here this condition is satisfied when $\alpha = 1$ for both choices of $\mathbf{L}$.

These choices are tested along with the model-free option. This latter option requires the special experiment at each iteration, involving the inputting of the reversed time error signal into the system. This experiment is carried out in simulation and the output is affected by a new realisation of the disturbance signals. The same $\gamma(k)$ is used as for the other two options.

In addition to testing the different options for the algorithm developed using SA theory, a fixed learning gain ILC algorithm using the same $\mathbf{L} = \hat{\mathbf{G}}^{-1}$, but with $\gamma(k) = 1$, is also tested in simulation. It should be noted that this choice of $\gamma(k)$ does not satisfy the conditions set by SA and therefore almost sure convergence to the optimal input is not guaranteed. This algorithm corresponds to the deterministic algorithm considered in Chapter 5.

Table 6.1 shows the values of $\frac{1}{N}\hat{E}\{\boldsymbol{\epsilon}(k,\mathbf{u}(k))\}^T\hat{E}\{\boldsymbol{\epsilon}(k,\mathbf{u}(k))\}$ found for the different options. We see that all options achieve prac-

k	$\mathbf{L} = \hat{\mathbf{G}}^{-1}$	$\mathbf{L} = \hat{\mathbf{G}}^T$	Model-free	Deterministic
0	0.0712	0.0712	0.0712	0.0712
10	0.000010	0.000018	0.000016	0.000015
20	0.000009	0.000014	0.000014	0.000015

Table 6.1. $\frac{1}{N}\hat{E}\{\boldsymbol{\epsilon}(k,\mathbf{u}(k))\}^T\hat{E}\{\boldsymbol{\epsilon}(k,\mathbf{u}(k))\}$ found in simulation

tically zero-mean converged controlled error compared to the initial mean error. This result is as expected, even for the deterministic, fixed-gain algorithm, which was shown in the last chapter to achieve zero-mean controlled error.

Table 6.2 shows the values of $\frac{1}{N}\hat{E}\{\boldsymbol{\epsilon}^T(k,\mathbf{u}(k))\boldsymbol{\epsilon}(k,\mathbf{u}(k))\}$ found for the different options. It can be seen that all three options devel-

oped using SA theory give mean square error values that are about the half of the deterministic, fixed-gain algorithm, illustrating the benefit of taking into account the stochastic affects.

k	$\mathbf{L} = \hat{\mathbf{G}}^{-1}$	$\mathbf{L} = \hat{\mathbf{G}}^{T}$	Model-free	Deterministic
0	0.0728	0.0728	0.0728	0.0728
10	0.0019	0.0016	0.0016	0.0031
20	0.0017	0.0016	0.0016	0.0031

Table 6.2. $\frac{1}{N}\hat{E}\{\boldsymbol{\epsilon}^T(k, \mathbf{u}(k))\boldsymbol{\epsilon}(k, \mathbf{u}(k))\}$ found in simulation

6.6 Experimental results

The model-free algorithm is applied to the tracking control of the LPMSM. As in the experiments of Chapter 5, the two-degree-of-freedom position controller is used to control the motor's position, operating at a sampling frequency of 2kHz. The input, $\mathbf{u}(k)$, computed by the ILC algorithm, is used as the position reference signal of the closed-loop system. The desired output position, $y_d(t)$, is also the same as that used in the experiments in the previous chapter i.e. a series of three Macromotions in the positive direction, followed by a similar series in the negative direction, see Figure 5.1.

If the model identified for the experiments in Chapter 5 is written in pole-zero form we have, from (5.53):

$$\hat{G}(q^{-1}) = \frac{0.00207(q - 0.456)(q - (-0.4567 \pm 3.1675j))}{(q - (0.8182 \pm 0.3214j))(q - (0.7336 \pm 0.1626j))}. \quad (6.53)$$

As can be seen, this model has 2 unstable zeros, due to the discretisation process. These unstable zeros make the matrix $\hat{\mathbf{G}}$ badly conditioned so if its inverse is used as $\mathbf{L}$ the signals grow very large before converging and are not practically applicable. Either the choice of

$\mathbf{L} = \hat{\mathbf{G}}^T$ or the model-free option are thus preferable as they do not require the inversion of $\hat{\mathbf{G}}$. Here the model-free choice is tested.

The sequence $\gamma(k) = \frac{\alpha}{k+1}$ is used with $\alpha = 0.85$, which is chosen to produce monotonic convergence. The initial input is taken as $\mathbf{u}(0) = \mathbf{y}_d$ and 100 iterations are carried out. Figures 6.1 and 6.2 show the convergence of the mean square error $\frac{1}{N}\mathbf{e}^T(k, \mathbf{u}(k))\mathbf{e}(k, \mathbf{u}(k))$ and the initial and final tracking achieved, respectively.

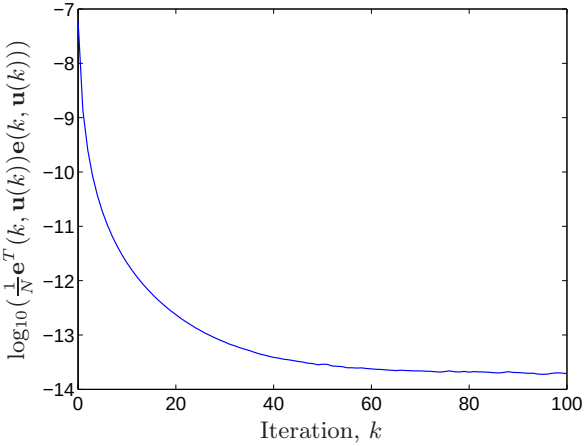

Fig. 6.1. Mean square error values obtained using the model-free method

As can be seen the algorithm considerably improves the tracking and near monotonic convergence is achieved.

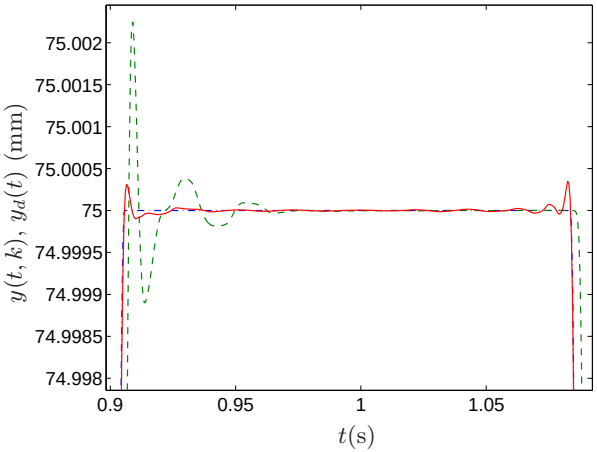

Fig. 6.2. LPMSM output at iteration $k = 0$ (dashed) and $k = 100$ (solid) using the model-free method, $y_d(t)$ (dot-dash)

6.7 Conclusions

The main contribution of this chapter is to show how stochastic approximation theory can be used to derive and analyse ILC algorithms for linear time-invariant systems that are less sensitive to non-repetitive disturbances. SA theory has provided general conditions that ensure almost sure convergence of the algorithm to the optimal input in the presence of stochastic disturbances. The decreasing gain algorithm, examined in Chapter 5, is a specific case of a family of algorithms that satisfy the conditions.

ILC for LTI systems has been considered here. Many of the results apply, however, to linear time-varying-in-the-iteration systems as well. In this case, however, the matrix $\mathbf{G}$ would not be lower triangular Toeplitz but a general lower triangular matrix instead. This implies that $\mathbf{L}$, $\hat{\mathbf{G}}$ and $\mathbf{I} + \boldsymbol{\Delta}$ will not, in general, commute.

It is noted that the conditions imposed by SA require the learning gain to tend to zero as the iterations tend to infinity. This requirement is essential for learning algorithms to converge to the ideal input. Practically it means that the learning ceases after a large number of iterations and if the desired output or repetitive disturbances change slowly the algorithm will not react and the tracking will deteriorate. It is thus necessary to have a surveillance program that restarts the learning when the errors rise above a certain threshold.

Appendix

6.A Input weighting

Using the minimisation philosophy discussed in Remark 1 of Section 6.4.3, it is possible to influence the amplitude of the converged input signal by modifying the cost function (6.45) to include a weighting on the input signal i.e.

$$J_{SA}(k,\mathbf{u}(k)) = \frac{1}{2}\mathrm{tr}\Big(E\{\boldsymbol{\epsilon}(k,\mathbf{u}(k))\mathbf{W}_\epsilon\boldsymbol{\epsilon}^T(k,\mathbf{u}(k))\} + \mathbf{u}(k)\mathbf{W}_u\mathbf{u}^T(k)\Big), \quad (6.54)$$

where $\mathbf{W}_\epsilon$ and $\mathbf{W}_u$ are $(N-m)\times(N-m)$ positive- and positive-semidefinite weighting matrices, respectively. The minimum of this criterion is found when:

$$\begin{aligned}\frac{dJ_{SA}(k,\mathbf{u}(k))}{d\mathbf{u}(k)}\Big|_{\mathbf{u}(k)=\mathbf{u}^*} &= E\left\{\left(\frac{\partial\boldsymbol{\epsilon}(k,\mathbf{u}(k))}{\partial\mathbf{u}(k)}\Big|_{\mathbf{u}(k)=\mathbf{u}^*}\right)^T\boldsymbol{\epsilon}(k,\mathbf{u}^*)\right\} \\ &\quad + \frac{1}{2}\frac{\partial}{\partial\mathbf{u}(k)}\mathbf{u}^T(k)\mathbf{W}_u\mathbf{u}(k)\Big|_{\mathbf{u}(k)=\mathbf{u}^*} \\ &= -\mathbf{G}^T\mathbf{W}_\epsilon E\{\boldsymbol{\epsilon}(k,\mathbf{u}^*)\} + \mathbf{W}_u\mathbf{u}^* = 0,\end{aligned} \quad (6.55)$$

whose solution is:

$$\mathbf{u}^* = (\mathbf{W}_u + \mathbf{G}^T\mathbf{W}_\epsilon\mathbf{G})^{-1}\mathbf{G}^T\mathbf{W}_\epsilon\mathbf{y_d}. \quad (6.56)$$

With this input the controlled error is:

$$\begin{aligned}\boldsymbol{\epsilon}(\mathbf{u}^*) &= \mathbf{y_d} - \mathbf{G}(\mathbf{W}_u + \mathbf{G}^T\mathbf{W}_\epsilon\mathbf{G})^{-1}\mathbf{G}^T\mathbf{W}_\epsilon\mathbf{y_d} - \mathbf{d}(k) \\ &= \mathbf{G}(\mathbf{W}_u + \mathbf{G}^T\mathbf{W}_\epsilon\mathbf{G})^{-1}\mathbf{W}_u\mathbf{G}^{-1}\mathbf{y_d} - \mathbf{d}(k).\end{aligned} \quad (6.57)$$

It can be seen that when $\mathbf{W}_u = 0$ the optimal solution gives zero-mean controlled error, as expected, but when $\mathbf{W}_u \neq 0$, perfect mean tracking is no longer achieved.

In order to calculate $\mathbf{u}^*$ from (6.56), exact knowledge of G is, however, required. As before, it is possible to avoid this requisite and calculate the minimiser using the Robbins-Monro algorithm:

$$\begin{aligned}\mathbf{u}(k+1) &= \mathbf{u}(k) + \gamma(k)(\mathbf{W}_\epsilon \mathbf{G}^T \mathbf{e}(k, \mathbf{u}(k)) - \mathbf{W}_u \mathbf{u}(k)) \\ &= (I - \gamma(k)\mathbf{W}_u)\mathbf{u}(k) + \gamma(k)\mathbf{W}_\epsilon \mathbf{G}^T \mathbf{e}(k, \mathbf{u}(k)). \end{aligned} \tag{6.58}$$

This algorithm converges almost surely to the optimal value $\mathbf{u}^*$, since for almost sure convergence of (6.58):

$$\frac{d^2 J_{SA}(k, \mathbf{u}(k))}{d\mathbf{u}(k)^2} = \mathbf{G}^T \mathbf{W}_\epsilon \mathbf{G} + \mathbf{W}_u > 0, \tag{6.59}$$

should hold, and this is the case as $\mathbf{W}_\epsilon$ and $\mathbf{W}_u$ are positive- and positive-semidefinite matrices, respectively, and $\mathbf{G}$ is non-singular.

7

ILC for LPV systems

7.1 Introduction

In the previous two chapters ILC algorithms that reduce the methodology's sensitivity to iteration-varying, stochastic disturbances have been studied. In practice, a system may also be affected by other types of iteration variation, such as deterministic changes in its dynamics and disturbances. In this case, instead of using the previous approaches aimed at making the algorithms less sensitive to these variations, it may be possible to use available information about the changes to adapt the algorithm.

To the author's knowledge, very little work has been done on this problem. In [40] the problem of deterministically iteration varying disturbances is considered. It is shown that, by using the internal-model principle in the iteration domain, the disturbances can be rejected as the iterations tend to infinity. The problem with this approach is that it is necessary to know the form of the disturbance variation in advance in order to include its model in the ILC controller.

In this chapter an ILC algorithm is developed that can lead to improved tracking for systems that can be represented by the LPV

class of systems, and therefore whose dynamics change as a function of a scheduling parameter. ILC for LPV systems has been considered in [31]. The variation due to the changing scheduling parameter is, however, assumed to take place during the iteration, rather than from one iteration to the next. The problem considered is, therefore, different to that studied here.

The method developed in this chapter is applied to the LPMSM. LPMSMs are affected by a periodic, position-dependent force ripple disturbance. When a movement starts from the same place this disturbance will be repetitive. ILC can thus adjust the system's input to compensate for it. However, if the movement starts from a different position, the disturbance will change and the learnt input will no longer produce optimal tracking. This problem has been investigated in [60]. The method proposed there is to learn the input that gives optimal tracking for a specific starting position. This input will compensate for errors due to the system's dynamics and the ripple force disturbance. It can thus be decomposed into these two components. When the movement is executed from a different starting position the component compensating the system's dynamics can be applied to the system, plus a phase-shifted version of the component that compensates the periodic, position dependent disturbance. The phase shift will be a function of the distance between starting positions. This method has the benefit that the input only needs to be learnt once, rather than for every starting position. Its disadvantage is that the learning process has to be done offline as a tuning procedure in order that the same starting position is used at each iteration.

In this chapter it will be shown how, for a certain class of movements whose amplitude is negligible compared to the period of the force ripple, the LPMSM can be modelled as an LPV system with the position-dependent force ripple as the scheduling parameter. The proposed LPV ILC algorithm is therefore applicable to the LPMSM and is shown to produce better performance than a standard LTI ILC algorithm.

The chapter is organised as follows. In Section 7.2 the problem is formulated. The proposed learning algorithm is presented in 7.3. Then its application to the LPMSM is detailed in Section 7.4. Finally some conclusions are made in Section 7.5.

7.2 Problem formulation

7.2.1 System description

The output at time t of the LPV SISO discrete-time system, resulting from linearising a real nonlinear system about the operating point $\boldsymbol{\sigma}(k)$, is given by:

$$\begin{aligned} &A(\boldsymbol{\sigma}(k), q^{-1}) y(t, k, \boldsymbol{\sigma}(k)) \\ &\quad = B(q^{-1}) u(t, k) + d(t, k, \boldsymbol{\sigma}(k)) + n(t, k, \boldsymbol{\sigma}(k)), \end{aligned} \tag{7.1}$$

where

$$A(\boldsymbol{\sigma}(k), q^{-1}) = \sum_{j=0}^{n_a} a_j(\boldsymbol{\sigma}(k)) q^{-j} \text{ and } B(q^{-1}) = \sum_{j=0}^{n_b} b_j q^{-j}.$$

$d(t, k, \boldsymbol{\sigma}(k))$ and $n(t, k, \boldsymbol{\sigma}(k))$ are a deterministic and a stochastic disturbance, respectively, both possibly dependent on $\boldsymbol{\sigma}$. The operating point $\boldsymbol{\sigma}(k)$ remains constant throughout repetition k. The dependence of the coefficients a_i and the deterministic disturbance on the scheduling parameter is assumed polytopic:

$$a_i(\boldsymbol{\sigma}(k)) = \sum_{j=0}^{J-1} \xi_j(\boldsymbol{\sigma}(k)) a_{i,j}$$

and

$$d(t, k, \boldsymbol{\sigma}(k)) = \sum_{j=0}^{J-1} \xi_j(\boldsymbol{\sigma}(k)) d_j(t),$$

$$0 \leq \xi_j(\boldsymbol{\sigma}(k)) \leq 1, \quad \sum_{j=0}^{J-1} \xi_j(\boldsymbol{\sigma}(k)) = 1, \tag{7.2}$$

where $\xi_j(\boldsymbol{\sigma}(k)) : \mathbb{R}^{n_\sigma} \to \mathbb{R}$. Additionally we designate the values of $\boldsymbol{\sigma}$ at the vertices of the polytopic space as $\boldsymbol{\sigma}_j$.

As in the previous chapter, the lifted-system representation can be used, giving the vectors:

$$\begin{aligned} \mathbf{u}(k) &= [u(0,k), u(1,k), \ldots, u(N-m-1,k)]^T \\ \text{and} \qquad \mathbf{y}(k,\boldsymbol{\sigma}(k)) &= [y(m,k,\boldsymbol{\sigma}(k)), y(m+1,k,\boldsymbol{\sigma}(k)), \\ &\qquad \ldots, y(N-1,k,\boldsymbol{\sigma}(k))]^T, \end{aligned} \tag{7.3}$$

with $\mathbf{y_d}$, $\mathbf{d_j}$ and $\mathbf{n}(k,\boldsymbol{\sigma}(k))$ defined similarly to $\mathbf{y}(k,\boldsymbol{\sigma}(k))$. This representation can then be used to write (7.1) as:

$$\begin{aligned} \mathbf{A}(\boldsymbol{\sigma}(k))\mathbf{y}(k,\boldsymbol{\sigma}(k)) &= \sum_{j=0}^{J-1} \xi_j(\boldsymbol{\sigma}(k))\mathbf{A}_j\mathbf{y}(k,\boldsymbol{\sigma}(k)) \\ &= \mathbf{B}\mathbf{u}(k) + \sum_{j=0}^{J-1} \xi_j(\boldsymbol{\sigma}(k))\mathbf{d}_j + \mathbf{n}(k,\boldsymbol{\sigma}(k)), \end{aligned} \tag{7.4}$$

where
$$\mathbf{A}_j = \begin{bmatrix} a_{0,j} & 0 & \ldots & 0 \\ a_{1,j} & a_{0,j} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-m-1,j} & a_{N-m-2,j} & \ldots & a_{0,j} \end{bmatrix}$$

and
$$\mathbf{B} = \begin{bmatrix} b_m & 0 & \ldots & 0 \\ b_{m+1} & b_m & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ b_{N-1} & b_{N-2} & \ldots & b_m \end{bmatrix}. \tag{7.5}$$

7.2.2 Ideal input

The measured error is defined as:

$$\mathbf{e}(k,\boldsymbol{\sigma}(k)) = \mathbf{y_d} - \mathbf{y}(k,\boldsymbol{\sigma}(k)). \tag{7.6}$$

The ideal input vector, defined as the one that achieves zero-mean error, i.e. $E\{\mathbf{e}(k,\boldsymbol{\sigma}(k))\} = \mathbf{0}\ \forall\boldsymbol{\sigma}$, is given by:

$$\mathbf{u}_0(k,\boldsymbol{\sigma}(k)) = \sum_{j=0}^{J-1} \xi_j(\boldsymbol{\sigma}(k))\mathbf{B}^{-1}\left(\mathbf{A}_j\mathbf{y_d} - \mathbf{d}_j\right). \tag{7.7}$$

At the values of $\boldsymbol{\sigma}$ that correspond to the vertices of the polytopic dependence we have:

$$\mathbf{u}_0(k,\boldsymbol{\sigma}_j) = \mathbf{u}_0(\boldsymbol{\sigma}_j) = \mathbf{B}^{-1}\left(\mathbf{A}_j\mathbf{y_d} - \mathbf{d}_j\right). \tag{7.8}$$

Using this (7.7) can be written as:

$$\mathbf{u}_0(k,\boldsymbol{\sigma}(k)) = \sum_{j=0}^{J-1} \xi_j(\boldsymbol{\sigma}(k))\mathbf{u}_0(\boldsymbol{\sigma}_j). \tag{7.9}$$

Remark: Expression (7.9) rationalises the class of LPV systems considered here i.e. LPV systems with dependence on the scheduling parameter solely in the denominator. Only for this system class does the ideal input depend linearly on the ideal inputs at the vertices. This linear dependence means the estimation of these inputs, as considered in the next section, can be done via linear least squares and a global minimiser will be found. Furthermore, as will be seen in the application section, real systems exist that belong to this system class.

7.2.3 Input parameterisation

We see from (7.9) that the ideal input is a function of the scheduling parameter $\boldsymbol{\sigma}(k)$. The ILC algorithm should therefore estimate an input that is also a function of $\boldsymbol{\sigma}(k)$. Motivated by the form of the ideal input, and under the assumption that the functions $\xi_j(\boldsymbol{\sigma}(k))$ are known, the input is parameterised as:

$$\mathbf{u}(k,\boldsymbol{\sigma}(k)) = \sum_{j=0}^{J-1} \xi_j(\boldsymbol{\sigma}(k))\mathbf{u}_j. \tag{7.10}$$

The system output can be written as:

$$\begin{aligned}\mathbf{y}(k,\boldsymbol{\sigma}(k)) &= \mathbf{A}^{-1}(\boldsymbol{\sigma}(k))\left[\mathbf{B}\mathbf{u}(k)+\mathbf{d}(k,\boldsymbol{\sigma}(k))+\mathbf{n}(k,\boldsymbol{\sigma}(k))\right]\\ &= \mathbf{G}(\boldsymbol{\sigma}(k))\mathbf{u}(k)+\mathbf{d}_A(k,\boldsymbol{\sigma}(k))+\mathbf{n}_A(k,\boldsymbol{\sigma}(k)),\end{aligned} \tag{7.11}$$

where $\mathbf{G}(\boldsymbol{\sigma}(k)) = \mathbf{A}^{-1}(\boldsymbol{\sigma}(k))\mathbf{B}$, $\mathbf{d}_A(k,\boldsymbol{\sigma}(k)) = \mathbf{A}^{-1}(\boldsymbol{\sigma}(k))\mathbf{d}(k,\boldsymbol{\sigma}(k))$ and $\mathbf{n}_A(k,\boldsymbol{\sigma}(k)) = \mathbf{A}^{-1}(\boldsymbol{\sigma}(k))\mathbf{n}(k,\boldsymbol{\sigma}(k))$. If the parameterised input (7.10) is applied to the system we have:

$$\begin{aligned}\mathbf{y}(k,\boldsymbol{\sigma}(k)) &= \mathbf{G}(\boldsymbol{\sigma}(k))\sum_{j=0}^{J-1}\xi_j(\boldsymbol{\sigma}(k))\mathbf{u}_j+\mathbf{d}_A(k,\boldsymbol{\sigma}(k))+\mathbf{n}_A(k,\boldsymbol{\sigma}(k))\\ &= \big[\xi_0(\boldsymbol{\sigma}(k))\mathbf{G}(\boldsymbol{\sigma}(k)),\xi_1(\boldsymbol{\sigma}(k))\mathbf{G}(\boldsymbol{\sigma}(k)),\ldots,\\ &\quad \xi_{J-1}(\boldsymbol{\sigma}(k))\mathbf{G}(\boldsymbol{\sigma}(k))\big]\left[\mathbf{u}_0^T,\mathbf{u}_1^T,\ldots,\mathbf{u}_{J-1}^T\right]^T\\ &\quad +\mathbf{d}_A(k,\boldsymbol{\sigma}(k))+\mathbf{n}_A(k,\boldsymbol{\sigma}(k))\\ &= \mathcal{G}(\boldsymbol{\sigma}(k))\bar{\mathbf{u}}+\mathbf{d}_A(k,\boldsymbol{\sigma}(k))+\mathbf{n}_A(k,\boldsymbol{\sigma}(k)),\end{aligned} \tag{7.12}$$

where $\mathcal{G}(\boldsymbol{\sigma}(k)) \in \mathbb{R}^{(N-m)\times J(N-m)}$ and $\bar{\mathbf{u}} \in \mathbb{R}^{J(N-m)}$.

The ideal input (7.9) is achieved when:

$$\bar{\mathbf{u}} = \bar{\mathbf{u}}_0 = \left[[\mathbf{u}_0(\boldsymbol{\sigma}_0)]^T,[\mathbf{u}_0(\boldsymbol{\sigma}_1)]^T,\ldots,[\mathbf{u}_0(\boldsymbol{\sigma}_{J-1})]^T\right]^T. \tag{7.13}$$

7.2.4 Assumptions

A7.1: The ideal input $\mathbf{u}_0(k,\boldsymbol{\sigma}(k))$ is realisable.

A7.2: The disturbance vector $\mathbf{n}(k,\boldsymbol{\sigma}(k))$ is a zero-mean, random vector with an unknown but bounded covariance matrix $\mathbf{R}_{\mathbf{n}}(k,\boldsymbol{\sigma}(k))$. Additionally, realisations of $\mathbf{n}(k,\boldsymbol{\sigma}(k))$ are independent.

7.3 Learning algorithm

The aim of the algorithm is to estimate $\bar{\mathbf{u}}_0$ over the iterations. The approach proposed here to find the estimate is to minimise a

quadratic cost function over all previous iterations i.e. to find the estimate that minimises:

$$J^K(\bar{\mathbf{u}}) = \frac{1}{2K}\sum_{k=0}^{K-1}\mathbf{e}^T(k,\boldsymbol{\sigma}(k),\bar{\mathbf{u}})\mathbf{e}(k,\boldsymbol{\sigma}(k),\bar{\mathbf{u}}), \tag{7.14}$$

where K is the number of completed iterations. Via some straightforward calculations, the estimate after K iterations, can be found as:

$$\bar{\mathbf{u}}(K) = \mathbf{P}_{\mathcal{G}}(K)\sum_{k=0}^{K-1}\mathcal{G}^T(\boldsymbol{\sigma}(k))[\mathbf{y_d} - \mathbf{d}_A(k,\boldsymbol{\sigma}(k)) - \mathbf{n}_A(k,\boldsymbol{\sigma}(k))], \tag{7.15}$$

where

$$\mathbf{P}_{\mathcal{G}}(K) = \left[\sum_{k=0}^{K-1}\mathcal{G}^T(\boldsymbol{\sigma}(k))\mathcal{G}(\boldsymbol{\sigma}(k))\right]^{-1}.$$

Alternatively, via some simple manipulations, (7.15) can be written in the recursive form as:

$$\bar{\mathbf{u}}(k+1) \;=\; \bar{\mathbf{u}}(k) + \mathbf{P}_{\mathcal{G}}(k+1)\mathcal{G}^T(\boldsymbol{\sigma}(k))\mathbf{e}(k,\boldsymbol{\sigma}(k),\bar{\mathbf{u}}(k)). \tag{7.16}$$

The error signal:

$$\mathbf{e}(k,\boldsymbol{\sigma}(k),\bar{\mathbf{u}}(k)) = \mathbf{y_d} - \mathcal{G}(\boldsymbol{\sigma}(k))\bar{\mathbf{u}}(k) - \mathbf{d}_A(k,\boldsymbol{\sigma}(k)) - \mathbf{n}_A(k,\boldsymbol{\sigma}(k))$$

can be evaluated experimentally by applying the input $\mathbf{u}(k,\boldsymbol{\sigma}(k))$ from (7.10) based on $\bar{\mathbf{u}}(k)$ to the real system. Therefore we see that the estimate $\bar{\mathbf{u}}$ can be evaluated recursively using data measured from the real system.

7.3.1 Consistency of estimates

Next a condition that assures consistent estimates is given.

Theorem 7.1 *Under the assumptions made in Subsection 7.2.4, the algorithm (7.16) is a consistent estimator, i.e.* $\bar{\mathbf{u}}(K)$ *converges almost surely to* $\bar{\mathbf{u}}_0$ *as* $K \to \infty$*, if:*

$$\lim_{K\to\infty} \frac{1}{K}\mathbf{P}_{\mathcal{G}}^{-1}(K) \tag{7.17}$$

is nonsingular.

Proof: The recursive algorithm (7.16) has the same asymptotic properties as the batch result (7.15) so the consistency of (7.15) can be considered. (7.15) can be rewritten as:

$$\begin{aligned}\bar{\mathbf{u}}(K) &= K\mathbf{P}_{\mathcal{G}}(K)\frac{1}{K}\sum_{k=0}^{K-1}\mathcal{G}^T(\boldsymbol{\sigma}(k))\left[\mathcal{G}(\boldsymbol{\sigma}(k))\bar{\mathbf{u}}_0 - \mathbf{n}_A(k,\boldsymbol{\sigma}(k))\right]\\ &= \bar{\mathbf{u}}_0 - K\mathbf{P}_{\mathcal{G}}(K)\frac{1}{K}\sum_{k=0}^{K-1}\mathcal{G}^T(\boldsymbol{\sigma}(k))\mathbf{n}_A(k,\boldsymbol{\sigma}(k)).\end{aligned}$$

In order for the estimates to be consistent it is necessary that:

$$\lim_{K\to\infty} \frac{1}{K}\mathbf{P}_{\mathcal{G}}^{-1}(K) \tag{7.18}$$

be nonsingular and

$$\mathbf{h}(K) = \frac{1}{K}\sum_{k=0}^{K-1}\mathcal{G}^T(\boldsymbol{\sigma}(k))\mathbf{n}_A(k,\boldsymbol{\sigma}(k)) \to \mathbf{0} \text{ a.s., as } K \to \infty. \tag{7.19}$$

(7.18) is the condition in the theorem that should be satisfied. To show that (7.19) is true we first write:

$$\begin{aligned}\mathcal{G}^T(\boldsymbol{\sigma}(k))\mathbf{n}_A(k,\boldsymbol{\sigma}(k)) &= \mathcal{G}^T(\boldsymbol{\sigma}(k))\mathbf{A}^{-1}(\boldsymbol{\sigma}(k))\mathbf{n}(k,\boldsymbol{\sigma}(k))\\ &= \mathbf{m}(k,\boldsymbol{\sigma}(k)).\end{aligned}$$

Since $\mathbf{n}(k,\boldsymbol{\sigma}(k))$ is assumed to be zero mean and independent between iterations, and $\mathbf{A}$ is nonsingular, this expression means that

$\mathbf{m}(k, \boldsymbol{\sigma}(k))$ will also be zero mean and independent between iterations. Additionally $\mathbf{m}(k, \boldsymbol{\sigma}(k))$'s covariance matrix is given by:

$$\mathbf{R_m}(k, \boldsymbol{\sigma}(k)) = \mathcal{G}^T(\boldsymbol{\sigma}(k))\mathbf{A}^{-1}(\boldsymbol{\sigma}(k))\mathbf{R_n}(k, \boldsymbol{\sigma}(k))\mathbf{A}^{-T}(\boldsymbol{\sigma}(k))\mathcal{G}(\boldsymbol{\sigma}(k)).$$

Since $\mathbf{R_n}(k, \boldsymbol{\sigma}(k))$ is assumed bounded and $\mathbf{A}$ is nonsingular, $\mathbf{R_m}(k, \boldsymbol{\sigma}(k))$ will be bounded.

The ith component of $\mathbf{h}(K)$ in (7.19) therefore represents the sample average of a sequence of zero-mean, independent random variables with finite, though possibly different, variances $\sigma_m^2(i, k, \boldsymbol{\sigma}(k))$. Lemma 2.3 implies that (7.19) is satisfied if:

$$\lim_{K\to\infty} \sum_{k=0}^{K-1} \frac{\sigma_m^2(i, k, \boldsymbol{\sigma}(k))}{(k+1)^2} < \infty. \tag{7.20}$$

Since

$$\sum_{k=0}^{K-1} \frac{\sigma_m^2(i, k, \boldsymbol{\sigma}(k))}{(k+1)^2} \leq \overline{\sigma}_m^2(i) \sum_{k=0}^{K-1} \frac{1}{(k+1)^2}$$

where $\sigma_m^2(i, k, \boldsymbol{\sigma}(k)) \leq \overline{\sigma}_m^2(i) < \infty \ \forall k$, and

$$\lim_{K\to\infty} \sum_{k=0}^{K-1} \frac{1}{(k+1)^2} = \frac{\pi}{6},$$

(7.20) is satisfied. The theorem is therefore proved. ∎

Remark: The condition (7.17) in Theorem 7.1 is a persistency of excitation condition that requires the scheduling parameter trajectory to be sufficiently rich.

7.4 Application results

As mentioned previously, the proposed method is applied to the LPMSM. The PID feedback controller operating at a sampling frequency of 2kHz is used to control the motor's position. The input,

$\mathbf{u}(k, \boldsymbol{\sigma}(k))$, computed by the ILC algorithm, is used as the closed-loop system's position reference signal.

The desired output motion is the Micromotion. In this chapter we consider the problem of improving the tracking performance of the LPMSM when it executes the movement repetitively but from a different starting position each repetition. This scenario is clearly possible in an industrial setting. If the system's dynamics and disturbances were position-independent, this problem could be solved with an LTI ILC algorithm. This, however, is not the case for the LPMSM, as can be seen from Figures 7.1 and 7.2. In these figures the mean system's response is shown when the movement is carried out at 10 different starting positions spaced 1.6 mm apart. The movements are repeated 10 times at each position in order to be able to differentiate between the effect of stochastic disturbances on the system's response and that due to changing dynamics. It can be

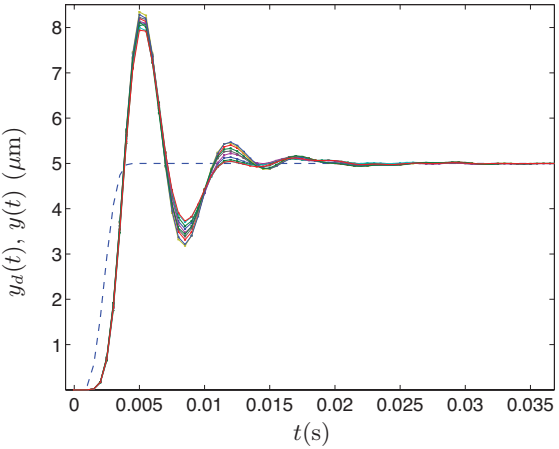

Fig. 7.1. Mean system response to desired output $y_d(t)$ (dashed) applied as closed-loop reference at 10 different positions

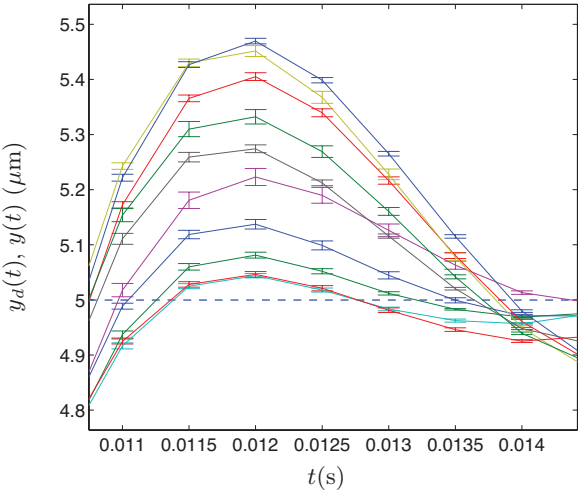

Fig. 7.2. Zoom of mean system response at different positions with error bars showing 1 standard deviation

clearly seen that significantly different responses are obtained which differ by an amount greater than could be expected from stochastic effects.

7.4.1 System modelling

Experimentally, a position dependence of the system's response has been shown. In order to understand where this comes from a model of the system will now be developed. The position $y(t,k)$ of the translator of the LPMSM, at time t and repetition k, obeys the following equation:

$$\frac{d^2y(t,k)}{dt^2} = f_m(t,k) - c_f\frac{dy(t,k)}{dt} - f_r(y(t,k)) \tag{7.21}$$

where $f_m(t,k)$ is the force applied to the motor, c_f is the viscous friction coefficient and $f_r(y(t,k))$ is the force ripple. Since the system has a current loop whose dynamics are much faster than the position dynamics, the force applied to the motor can be considered proportional to the applied current i.e. $f_m(t,k) = k_m i(t,k)$. The force ripple term contains both the cogging force and reluctance force, and is primarily a periodic function of the system's position $y(t,k)$ that can be modelled by:

$$f_r(y(t,k)) = \sum_{i=0}^{n_r-1} F_{r,i} \sin\left(\frac{2\pi y(t,k)}{T_{r,i}} + \phi_{r,i}\right), \tag{7.22}$$

where the periods $T_{r,i}$ are influenced by different factors such as the average pitch of the magnets and the size of the bearings. We see that this force has a nonlinear dependence on the system's position making the system, itself, nonlinear.

The developed ILC algorithm is for the LPV system class. The system represented by (7.21) does not have the form of an LPV system. Nonetheless for the specific class of movements considered here, whose amplitude is negligible compared to the force ripple's principal period, certain manipulations can be made in order to obtain an LPV model. Under this assumption on the movement amplitude, we can assume that the force ripple varies linearly in a small zone about a certain operating point corresponding to a specific starting position $\bar{y}(k)$. Using this idea we can obtain an LPV model, whose dynamics change as a function of the starting position. To show this we linearise the force ripple term at the operating point. This operation is done by first evaluating (7.21) at $y(t,k) = \bar{y}(k) + \delta y(t,k)$, where $\delta y(t,k)$ is a small deviation about $\bar{y}(k)$:

$$\begin{aligned}\frac{d^2(\bar{y}(k)+\delta y(t,k))}{dt^2} = \bar{f}_m(k) + \delta f_m(t,k) - \\ c_f \frac{d(\bar{y}(k)+\delta y(t,k))}{dt} - f_r(\bar{y}(k)+\delta y(t,k))),\end{aligned} \tag{7.23}$$

where $\bar{f}_m(k) + \delta f_m(t,k)$ is the force input required to maintain the system at this operating point. Using a Taylor series expansion, (7.23) can be written approximately as:

$$\begin{aligned}\frac{d^2(\bar{y}(k)+\delta y(t,k))}{dt^2} &\approx \bar{f}_m(k) + \delta f_m(t,k) \\ &\quad - c_f \frac{d(\bar{y}(k)+\delta y(t,k))}{dt} - f_r(\bar{y}(k)) \\ &\quad - \frac{df_r(y(t,k))}{dy(t,k)}\Big|_{y(t,k)=\bar{y}(k)} \delta y(t,k). \quad (7.24)\end{aligned}$$

Subtracting (7.21), evaluated at $y(t,k) = \bar{y}(k)$, from this gives:

$$\begin{aligned}\frac{d^2}{dt^2}\delta y(t,k) &\approx \delta f_m(t,k) - c_f \frac{d}{dt}\delta y(t,k) \\ &\quad - \frac{df_r(y(t,k))}{dy(t,k)}\Big|_{y(t,k)=\bar{y}(k)} \delta y(t,k) \\ &= \delta f_m(t,k) - c_f \frac{d}{dt}\delta y(t,k) - \boldsymbol{\sigma}(\bar{y}(k))\delta y(t,k), \quad (7.25)\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{\sigma}(\bar{y}(k)) &= \frac{df_r(y(t,k))}{dy(t,k)}\Big|_{y(t,k)=\bar{y}(k)} \\ &= \sum_{i=0}^{n_r-1} \left(\frac{2\pi F_{r,i}}{T_{r,i}}\right) \cos\left(\frac{2\pi \bar{y}(k)}{T_{r,i}} + \phi_{r,i}\right). \quad (7.26)\end{aligned}$$

This model can be seen to represent an LPV system with $\boldsymbol{\sigma}(\bar{y}(k))$ as the scheduling parameter. For simplicity of notation from now on $\delta y(t,k)$ and $\delta f_m(t,k)$ will be replaced by $y(t,k)$ and $f_m(t,k)$, respectively, though it should be remembered that they represent small deviations about the operating point values.

Taking the Laplace transform of (7.25), for a fixed $\boldsymbol{\sigma}(\bar{y}(k))$, gives:

$$\frac{Y(s,\boldsymbol{\sigma}(\bar{y}(k)))}{F_m(s,k)} = P_c(s,\boldsymbol{\sigma}(\bar{y}(k))) = \frac{1}{s^2 + c_f s + \boldsymbol{\sigma}(\bar{y}(k))}. \quad (7.27)$$

In practice, the ILC algorithm is applied to the reference signal of the system operating in closed loop with a PID feedback controller:

$$K_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right). \tag{7.28}$$

The closed-loop transfer function between the reference signal and the system output is:

$$\frac{Y(s, \boldsymbol{\sigma}(\bar{y}(k)))}{U(s)} = \frac{K_c(s)P_c(s, \boldsymbol{\sigma}(\bar{y}(k)))}{1 + K_c(s)P_c(s, \boldsymbol{\sigma}(\bar{y}(k)))}. \tag{7.29}$$

The ILC algorithm is applied in discrete-time. It is therefore necessary to discretise the transfer function (7.29). This discretisation is done using Euler's second method i.e. by substituting s by $(1 - z^{-1})/h$, where h is the sampling period. Additionally the discrete-time system has a two sampling period input delay. We therefore have the discrete-time closed-loop LPV system transfer function given by:

$$\begin{aligned} G(z^{-1}, \boldsymbol{\sigma}(\bar{y}(k))) &= \frac{B(z^{-1})}{A(\boldsymbol{\sigma}(\bar{y}(k)), z^{-1})} \\ &= \frac{z^{-2}\left(b_2 + b_3 z^{-1} + b_4 z^{-2}\right)}{a_0\left(\boldsymbol{\sigma}(\bar{y}(k))\right) + a_1\left(\boldsymbol{\sigma}(\bar{y}(k))\right) z^{-1} + a_2 z^{-2} + a_3 z^{-3}}, \end{aligned} \tag{7.30}$$

where

$$\begin{aligned} a_0\left(\boldsymbol{\sigma}(\bar{y}(k))\right) &= a_0^0 \xi_0(\boldsymbol{\sigma}(\bar{y}(k))) + a_0^1 \xi_1(\boldsymbol{\sigma}(\bar{y}(k))), \\ a_1\left(\boldsymbol{\sigma}(\bar{y}(k))\right) &= a_1^0 \xi_0(\boldsymbol{\sigma}(\bar{y}(k))) + a_1^1 \xi_1(\boldsymbol{\sigma}(\bar{y}(k))) \end{aligned}$$

and

$$\xi_0(\boldsymbol{\sigma}(\bar{y}(k)) = \frac{\overline{\boldsymbol{\sigma}} - \boldsymbol{\sigma}(\bar{y}(k))}{\overline{\boldsymbol{\sigma}} - \underline{\boldsymbol{\sigma}}}, \quad \xi_1(\boldsymbol{\sigma}(\bar{y}(k)) = \frac{\boldsymbol{\sigma}(\bar{y}(k)) - \underline{\boldsymbol{\sigma}}}{\overline{\boldsymbol{\sigma}} - \underline{\boldsymbol{\sigma}}}.$$

$\underline{\boldsymbol{\sigma}}$ and $\overline{\boldsymbol{\sigma}}$ are the minimum and maximum values of $\boldsymbol{\sigma}(\bar{y}(k))$, respectively.

It has, therefore, been shown that the LPMSM can be represented by an LPV model for the class of movements considered and thus the developed algorithm is applicable.

7.4.2 System identification

The developed ILC algorithm uses the matrix $\mathcal{G}(\boldsymbol{\sigma}(k))$, as seen in (7.16). This matrix is formed from the system matrix $\mathbf{G}(\boldsymbol{\sigma}(k))$ and the functions $\xi_i(\boldsymbol{\sigma}(\bar{y}(k)))$. A model of $G(z^{-1}, \boldsymbol{\sigma}(\bar{y}(k)))$ needs to be identified in order to construct the matrix $\mathbf{G}(\boldsymbol{\sigma}(k))$. This identification will be considered first, followed by the identification of $\xi_i(\boldsymbol{\sigma}(\bar{y}(k)))$.

The LPV model of $G(z^{-1}, \boldsymbol{\sigma}(\bar{y}(k)))$, $\hat{G}(z^{-1}, \boldsymbol{\sigma}(\bar{y}(k)))$, could be found by first identifying LTI models, each with the same structure (order and delay) as (7.30) at different positions. These models' parameters could then be used to find the parameters of the overall LPV model (7.30). This method is, however, very time consuming and calls into question the practical usefulness of the ILC method. A single LTI model $\hat{G}(q^{-1})$ is therefore identified to be used in the place of the LPV model in (7.16); the assumption being made that a certain amount of model uncertainty is acceptable.

In order to excite the system correctly for the identification of the LTI model, a PRBS signal is chosen as the system's reference signal. It uses a shift register of 10 bits and a divider of 7 giving a signal length of 7161 points. These values are chosen to sufficiently excite the system at low frequencies, and are calculated from an estimate of the system's settling time obtained via a step response test on the real system. The amplitude of the signal is selected large enough to give a good signal-to-noise ratio and reduce the effects of static friction, but small enough to remain in the zone of linearisation and avoid saturation. Two experiments are carried out at an arbitrarily chosen position, one set for parameter estimation and the other for validation. An output error model, with the same order and number of input delays as (7.30), is found to give a good fit to the validation data in simulation.

Next the identification of the functions $\xi_i(\boldsymbol{\sigma}(\bar{y}(k)))$ is undertaken. These functions are based on the scheduling parameter $\boldsymbol{\sigma}(\bar{y}(k))$. $\boldsymbol{\sigma}(\bar{y}(k))$ cannot be measured directly, only $\bar{y}(k)$. In order to calculate $\boldsymbol{\sigma}(\bar{y}(k))$ it is necessary to estimate the values of $F_{r,i}$, $T_{r,i}$ and

$\phi_{r,i}$ for $i = 0, \ldots, n_r - 1$. This can be done using measurements from the motor. The force applied by the feedback controller is measured whilst the motor moves at constant velocity. Since the acceleration is approximately equal to zero and the friction force is assumed constant, any variation in the applied force is to compensate the force ripple. By doing several experiments in the positive and negative directions, removing the offset due to the constant friction force and taking the average we can obtain an estimate of the force ripple, see Figure 7.3. The spectrum of the measured force ripple is calculated

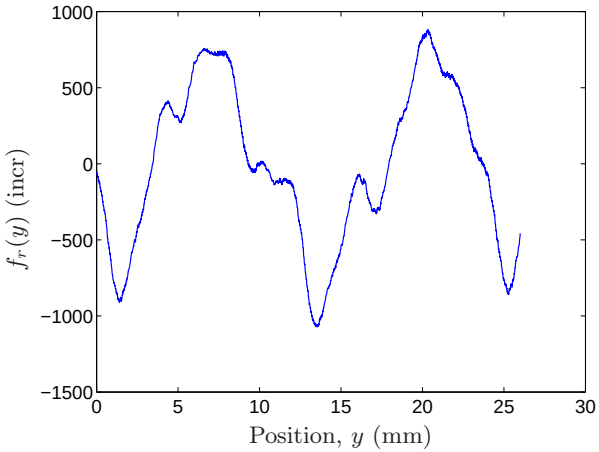

Fig. 7.3. Measured force ripple ('incr' are the LPMSM's unit of force)

in order to estimate the periods $T_{r,i}$. Significant peaks occur at spatial frequencies corresponding to periods of 2 mm, 3.7 mm, 8 mm, 13 mm and 21 mm. These values are thus used for $T_{r,i}$ for $i = 0, \ldots, 4$, respectively. With these values the parametric model of the force ripple is:

$$\hat{f}_r(y(t,k)) = \sum_{i=0}^{4} F_{r,i} \sin\left(\frac{2\pi y(t,k)}{T_{r,i}} + \phi_{r,i}\right)$$
$$= \begin{bmatrix} \sin\left(\frac{2\pi y(t,k)}{T_{r,0}}\right) \\ \cos\left(\frac{2\pi y(t,k)}{T_{r,0}}\right) \\ \sin\left(\frac{2\pi y(t,k)}{T_{r,1}}\right) \\ \vdots \\ \cos\left(\frac{2\pi y(t,k)}{T_{r,4}}\right) \end{bmatrix}^T \begin{bmatrix} f^1_{r,0} \\ f^2_{r,0} \\ f^1_{r,1} \\ \vdots \\ f^2_{r,4} \end{bmatrix}.$$

$f^1_{r,0}, f^2_{r,0}, \ldots$ are estimated by a least squares procedure using data from a 13 mm section of the total measured data. This length is chosen as it corresponds to the period of the largest component in the force ripple spectrum. The values of $F_{r,1}, \phi_{r,1}, \ldots$ are then calculated from these estimates. This procedure gives $F_{r,0} = 92.03$ incr, $\phi_{r,0} = 0.74$ rad, $F_{r,1} = 179.07$ incr, $\phi_{r,1} = 1.20$ rad, $F_{r,2} = -253.36$ incr, $\phi_{r,2} = 0.67$ rad, $F_{r,3} = -754.19$ incr, $\phi_{r,3} = 0.62$ rad, $F_{r,4} = 435.79$ incr and $\phi_{r,4} = 0.86$ rad. A comparison of the output of the model formed with these parameters and the force estimated from the measurements is shown in Figure 7.4. We see that the model fits the measurement estimates reasonably well.

Using this model, the scheduling parameter $\boldsymbol{\sigma}(\bar{y}(k))$ can be calculated from (7.26). The maximum and minimum values of $\boldsymbol{\sigma}(\bar{y}(k))$, $\overline{\boldsymbol{\sigma}}$ and $\underline{\boldsymbol{\sigma}}$ are also needed. As the dependence of the derivative of $\boldsymbol{\sigma}(\bar{y}(k))$ is nonlinear w.r.t. $\bar{y}(k)$, it is not possible to evaluate these values analytically. They are, therefore, estimated by gridding $\boldsymbol{\sigma}(\bar{y}(k))$ over the range considered.

7.4.3 Application of the proposed algorithm

With the required model identified, the proposed algorithm is applied to the LPMSM. The desired motion consists of a Micromotion in the positive direction followed by one in the negative direction. Its length

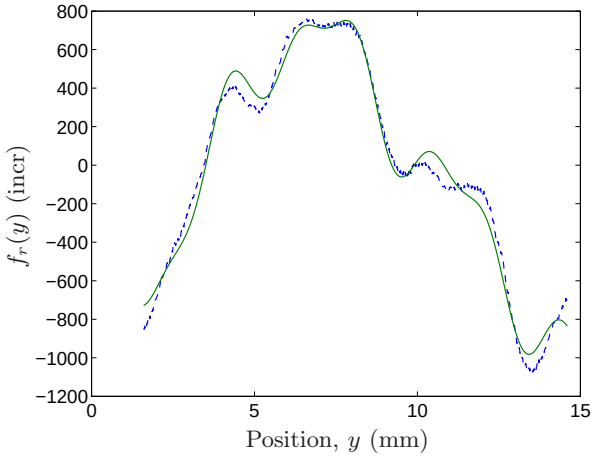

Fig. 7.4. Measured force ripple (dashed) and model estimate (solid)

is such that $N - m = 26$. In the implementation of the algorithm the matrix:

$$\mathbf{P}_{\mathcal{G}}(k+1) = \left[\sum_{i=0}^{k} \mathcal{G}^T(\boldsymbol{\sigma}(\bar{y}(i)))\mathcal{G}(\boldsymbol{\sigma}(\bar{y}(i)))\right]^{-1}$$

where

$$\mathcal{G}(\boldsymbol{\sigma}(\bar{y}(k))) \;=\; [\xi_0(\boldsymbol{\sigma}(\bar{y}(k)))\mathbf{G}(\boldsymbol{\sigma}(\bar{y}(k))), \xi_1(\boldsymbol{\sigma}(\bar{y}(k)))\mathbf{G}(\boldsymbol{\sigma}(\bar{y}(k)))]$$

is required at each iteration of the algorithm. In order for the inverse to exist, it is necessary that this matrix be full rank. For this to be the case a persistency of excitation condition on the scheduling parameter trajectory should be satisfied. A necessary, but not sufficient, condition for the matrix to be full rank is that 2 different values

of $\bar{y}(k)$ are visited.[1] In order to satisfy this the algorithm is not used to calculate the system input until 2 different values of $\bar{y}(k)$ have been visited, using $\mathbf{u}(k, \boldsymbol{\sigma}(k)) = \mathbf{y_d}$ for $k = 0, 1$. Nonetheless the matrix $\mathbf{P}_{\mathcal{G}}^{-1}(k+1)$ is still ill-conditioned. This is because the identified model $\hat{G}(q^{-1})$ used to produce the matrix $\mathbf{G}$ has an unstable zero at $q = -7$. The $\mathbf{G}$ used to generate $\mathbf{P}_{\mathcal{G}}^{-1}(k+1)$ thus uses $\hat{G}(q^{-1})$ with this zero stabilised by replacing it with a zero at $q = -1/7$. This stabilisation process is motivated by the fact that it gives a system with a frequency response that has the same magnitude as $\hat{G}(q^{-1})$, though the phase is different.

The movement's starting positions are chosen as a sawtooth waveform that periodically visits 13 equally spaced positions in the 13 mm range considered.

The results obtained using the proposed method are compared with those obtained using an ILC algorithm developed under the assumption of the system being LTI. The algorithm is given by:

$$\mathbf{u}(k+1) = \mathbf{u}(k) + \frac{1}{k+1}\mathbf{G}^{-1}\mathbf{e}(\mathbf{u}(k)). \tag{7.31}$$

This algorithm can be motivated either by the stochastic approximation theory presented in the last chapter or as the equivalent recursive version of the least squares solution when the 2-norm of the tracking error is minimised over all iterations up to iteration $k+1$. The latter is the same motivation as that used to develop the LPV ILC algorithm so makes the comparison fair. The algorithm is implemented on the system with the starting position being changed in the same way as for the experiments with the proposed algorithm.

The mean square values achieved using the LPV and LTI ILC algorithms are shown in Figures 7.5 and 7.6. It can be seen, especially from Figure 7.6, that the LPV algorithm gives better tracking than

[1] This condition arises because $\mathcal{G}(\boldsymbol{\sigma}(\bar{y}(k))) \in \mathbb{R}^{(N-m)\times 2(N-m)}$ is of maximum rank $N-m$, and thus so is $\mathcal{G}^T(\boldsymbol{\sigma}(\bar{y}(k)))\mathcal{G}(\boldsymbol{\sigma}(\bar{y}(k)))$. $\mathbf{P}_{\mathcal{G}}^{-1}(k+1)$ must be of rank $2(N-m)$. This is only possible after 2 different values of $\bar{y}(k)$ are visited.

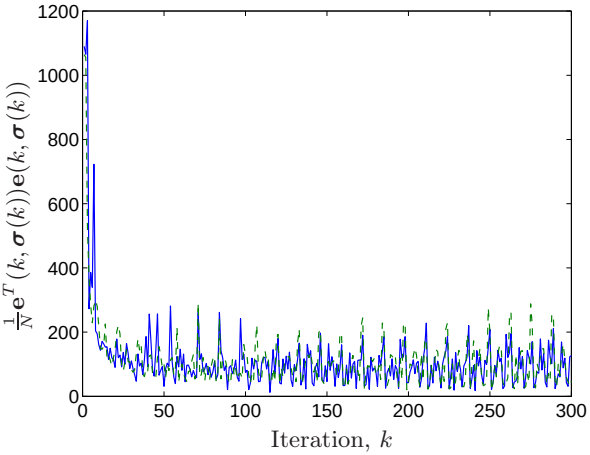

Fig. 7.5. Mean square values of $\mathbf{e}(k)$ using the proposed algorithm (solid) and the LTI algorithm (dashed)

that achieved with the LTI algorithm. More specifically the large peaks occurring with the LTI algorithm are reduced, meaning that the LPV ILC algorithm achieves more even tracking for all starting positions, as expected.

7.5 Conclusions

An ILC algorithm has been proposed for systems that can be represented by the discrete-time, LPV class of systems with polytopic dependence of the denominator polynomial's coefficients on the scheduling parameter. Consistency of the algorithm in the presence of nonstationary stochastic disturbances has been shown when the scheduling parameter trajectory is sufficiently exciting.

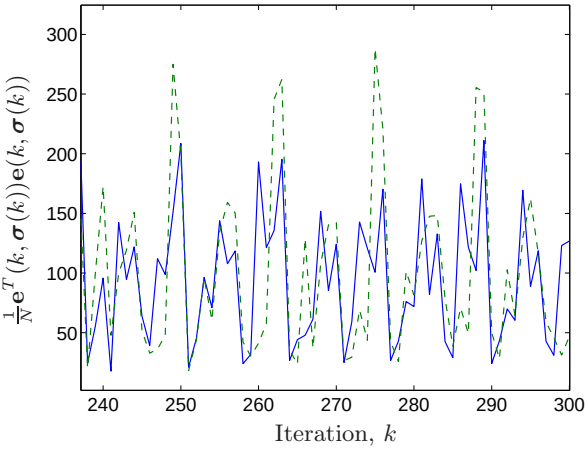

Fig. 7.6. Mean square values of $\mathbf{e}(k)$ using the proposed algorithm (solid) and the LTI algorithm (dashed)

The algorithm has been applied to the LPMSM, which has been shown to be an LPV system when its displacement is negligible compared to the period of the force ripple disturbance. Improved performance is achieved over that obtained using a standard LTI ILC algorithm.

Monotonic convergence of a norm of the error signal is of practical interest, and much attention has been given to this issue in LTI ILC, as seen in the previous chapters. Unfortunately, it seems unlikely that this property can be incorporated into LPV ILC algorithms as the system's dynamics change from one iteration to the next and so, depending on how they change, it is always possible that the error will increase slightly, though the overall trend should be to decrease.

Appendix

7.A Derivation of (7.16)

Equation (7.15) gives:

$$
\begin{aligned}
\bar{\mathbf{u}}(k{+}1) &= \mathbf{P}_{\mathcal{G}}(k+1)\sum_{i=0}^{k}\mathcal{G}^T(\boldsymbol{\sigma}(i))\big[\mathbf{y_d}-\mathbf{d}_A(i,\boldsymbol{\sigma}(i))-\mathbf{n}_A(i,\boldsymbol{\sigma}(i))\big]\\
&= \mathbf{P}_{\mathcal{G}}(k+1)\sum_{i=0}^{k-1}\mathcal{G}^T(\boldsymbol{\sigma}(i))\big[\mathbf{y_d}-\mathbf{d}_A(i,\boldsymbol{\sigma}(i))-\mathbf{n}_A(i,\boldsymbol{\sigma}(i))\big]\\
&\quad+\mathbf{P}_{\mathcal{G}}(k+1)\mathcal{G}^T(\boldsymbol{\sigma}(k))\big[\mathbf{y_d}-\mathbf{d}_A(k,\boldsymbol{\sigma}(k))-\mathbf{n}_A(k,\boldsymbol{\sigma}(k))\big]\\
&= \mathbf{P}_{\mathcal{G}}(k+1)\Big[\mathbf{P}_{\mathcal{G}}^{-1}(k)\bar{\mathbf{u}}(k)+\mathcal{G}^T(\boldsymbol{\sigma}(k))\big[\mathbf{y_d}-\mathbf{d}_A(k,\boldsymbol{\sigma}(k))\\
&\quad-\mathbf{n}_A(k,\boldsymbol{\sigma}(k))\big]\Big]\\
&= \mathbf{P}_{\mathcal{G}}(k+1)\Big[\big[\mathbf{P}_{\mathcal{G}}^{-1}(k+1)-\mathcal{G}^T(\boldsymbol{\sigma}(k))\mathcal{G}(\boldsymbol{\sigma}(k))\big]\,\bar{\mathbf{u}}(k)\\
&\quad+\mathcal{G}^T(\boldsymbol{\sigma}(k))\big[\mathbf{y_d}-\mathbf{d}_A(k,\boldsymbol{\sigma}(k))-\mathbf{n}_A(k,\boldsymbol{\sigma}(k))\big]\Big]\\
&= \bar{\mathbf{u}}(k)+\mathbf{P}_{\mathcal{G}}(k+1)\mathcal{G}^T(\boldsymbol{\sigma}(k))\big[\mathbf{y_d}-\mathcal{G}(\boldsymbol{\sigma}(k))\bar{\mathbf{u}}(k)\\
&\quad-\mathbf{d}_A(k,\boldsymbol{\sigma}(k))-\mathbf{n}_A(k,\boldsymbol{\sigma}(k))\big]\\
&= \bar{\mathbf{u}}(k)+\mathbf{P}_{\mathcal{G}}(k+1)\mathcal{G}^T(\boldsymbol{\sigma}(k))\mathbf{e}(k,\boldsymbol{\sigma}(k),\bar{\mathbf{u}}(k)),
\end{aligned}
$$

which is equal to expression (7.16).

8

Conclusions

8.1 Summary

In this thesis data-driven methods for tracking improvement have been proposed and analysed. For the general tracking problem, data-driven tuning methods for precompensators/prefilters are proposed. For the specific case of repetitive trajectory tracking, Iterative Learning Control has been studied. For both tracking problems algorithms have been investigated and developed for LTI systems and the broader system class of LPV systems.

In Chapter 3, a data-driven precompensator tuning method using the correlation approach is proposed. The method requires only one simple tuning experiment on the system. Moreover, the use of the correlation approach means that the 2-norm of the model-following criterion can be minimised and is asymptotically unaffected by the presence of stochastic disturbances on the measurements.

In Chapter 4, data-driven methods for precompensator tuning for LPV systems is considered. Since LPV systems do not commute in general, it is not possible to directly extend the method from Chapter 3. Nonetheless, in the specific case that the inverse of the system exists in the set of parameterised precompensators, the precompen-

sator and the system do commute, and parameters for the precompensator that represents the system inverse can be estimated using two straightforward experiments on the system. In the more, general case, when the precompensator and the system do not commute, a method is proposed that uses a number of experiments equal to twice the number of precompensator parameters. For both cases, consistency of the parameters is shown when the scheduling parameter is either noise free, or the noise affecting it is uncorrelated with that affecting the system output and the dependence on the scheduling parameters is affine.

In Chapter 5, a statistical analysis is carried out for a general ILC algorithm for LTI systems, as well as for a number of previously proposed algorithms that aim to be less sensitive to stochastic disturbances. Expressions are developed for the mean converged controlled error value and the controlled error variance. These expressions allow the different algorithms to be compared in a unified framework. It is shown that, when the disturbance spectra and spectrum of the desired output do not overlap, a filtered algorithm can give good performance. However, when this is not the case, an algorithm using an iteration decreasing gain can lead to a small mean and variance value for the error.

In Chapter 6, it is shown how ILC fits into the well-developed stochastic approximation framework. Using this framework, conditions are given that ensure almost sure convergence of the learnt input to the ideal input, which produces zero-mean controlled error.

In Chapter 7, an ILC algorithm for LPV systems, whose dynamics change from one iteration to the next, is developed. It is shown that the algorithm gives consistent estimates of the ideal input, despite the presence of nonstationary stochastic disturbances. Additionally it is shown that the LPMSM is a linear parameter varying system for the class of movements whose amplitude is negligible compared to the period of the LPMSM's force ripple disturbance.

From a practical point of view, the methods developed and analysed in this thesis can be easily applied to existing, stable open or closed-loop systems in order to improve their tracking performance.

Since they can be used to adjust the reference signal of a closed-loop system, they do not require the existing controller structure to be modified, which is clearly advantageous. Moreover, the methods do not require complex system information, such as probability distributions, which is sometimes the case for control algorithms for systems affected by stochastic disturbances.

The majority of the approaches developed in this thesis have been applied successfully in practice to the LPMSM, illustrating their industrial applicability.

8.2 Possible future work

There are a number of possible research directions that arise from the work done. A few are discussed below.

- It would be of interest to investigate data-driven precompensator tuning for multi-input multi-output systems. In general, as in the general SISO LPV case, the precompensator and system will not commute so the special tuning scheme of swapping their positions will not be feasible. A number of experiments will therefore be necessary to tune the precompensator parameters. It is possible, however, that using the correlation approach this number will be less than would be needed by a tuning method that minimises the error variance. This saving is because, as in the feedback controller tuning case [37], the system's inputs, and possibly scheduling parameters, could be simultaneously excited if they are uncorrelated.
- Experimentally the LPV ILC algorithm has been shown to be robust to a certain amount of model uncertainty. Nevertheless, a theoretical result quantifying the acceptable uncertainty would be of interest. Furthermore, the proposed method is restricted to LPV systems with a dependence on the scheduling parameter only in the denominator of the system's transfer function. It is clear that a more general algorithm that works for systems with

scheduling parameter dependence in the numerator as well would be of interest. A different approach to that presented in this thesis would be necessary, nonetheless, as the ideal input could not be represented as a linear combination of the ideal inputs at the polytopic vertices.

- A number of the consistency results have relied on the, practically impossible, use of an infinite number of data or iterations. It has been shown, via simulation and experiment, that the methods give good results even with a large, finite number of data/iterations. Nonetheless, it would be interesting to develop theoretical results quantifying the effect of using only this finite number of data.

A

Appendix

A.1 On the consistency of certain identification methods for LPV systems

A.1.1 Introduction

In this appendix the consistency of certain identification methods for LPV systems is addressed. In order to prove consistency it is necessary to apply ergodicity results to the time averages encountered in the analysis process. The ergodicity of the signals used in the identification of LPV systems is thus studied for different assumptions on the scheduling parameter. It is shown that the signals are ergodic if the scheduling parameter is noise-free or the measurement noise on the scheduling parameter is independent of that affecting the output. In these cases the use of instrumental variables leads to consistent parameter estimates. However, for the cases that the scheduling parameter's noise is correlated with that of the output signal or the LPV system does not have an affine dependency on the noisy scheduling parameter, the ergodicity of the signals cannot be proved and so consistency of the estimates cannot be shown.

The appendix is organised as follows. The identification problem is presented in Section A.1.2. The non-noisy scheduling parameter

case is considered in Section A.1.3. Then the noisy scheduling parameter case is analysed in Section A.1.4. Simulation results are presented in Section A.1.5. Finally, some concluding remarks are made in Section A.1.6.

A.1.2 Problem formulation

True system representation

Suppose that the measured output of the SISO LPV system $G_0(\boldsymbol{\sigma}(t), q^{-1})$ is given by (see Fig A.1):

$$y(t) = G_0(\boldsymbol{\sigma}(t), q^{-1})u(t) + H_0(\boldsymbol{\sigma}(t), q^{-1})\eta(t), \tag{A.1}$$

where $G_0(\boldsymbol{\sigma}(t), q^{-1})$ and $H_0(\boldsymbol{\sigma}(t), q^{-1})$ are the system and noise transfer operators, respectively. They are assumed to be LPV stable for all $\boldsymbol{\sigma}(t)$ in the operating zone.

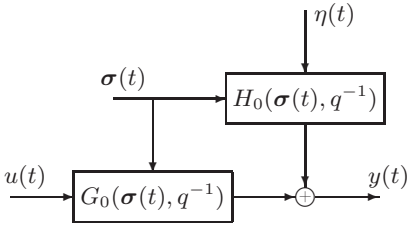

Fig. A.1. LPV system

$G_0(\boldsymbol{\sigma}(t), q^{-1})$ can be represented as:

$$G_0(\boldsymbol{\sigma}(t), q^{-1}) = \frac{B_0(\boldsymbol{\sigma}(t), q^{-1})q^{-m}}{A_0(\boldsymbol{\sigma}(t), q^{-1})} \tag{A.2}$$

where:

$$B_0(\boldsymbol{\sigma}(t), q^{-1}) = b_0^0(\boldsymbol{\sigma}(t)) + b_1^0(\boldsymbol{\sigma}(t))q^{-1} + \cdots + b_{n_b}^0(\boldsymbol{\sigma}(t))q^{-n_b} \quad \text{(A.3)}$$

and

$$A_0(\boldsymbol{\sigma}(t), q^{-1}) = 1 + a_1^0(\boldsymbol{\sigma}(t))q^{-1} + \cdots + a_{n_a}^0(\boldsymbol{\sigma}(t))q^{-n_a}. \quad \text{(A.4)}$$

n_b and n_a are the numerator and denominator orders, respectively. The coefficients of $B_0(\boldsymbol{\sigma}(t), q^{-1})$ are given by:

$$b_i^0(\boldsymbol{\sigma}(t)) = b_{i,0}^0\sigma_0(t) + b_{i,1}^0\sigma_1(t) + \cdots + b_{i,n_\sigma}^0\sigma_{n_\sigma}(t), \quad \text{(A.5)}$$

where $\sigma_j(t)$ represents the jth element of $\boldsymbol{\sigma}(t)$. As in the precompensator tuning case, this parameterisation allows a wide range of dependence on the scheduling parameter to be described. For example each $\sigma_j(t)$ could represent a function of a different scheduling parameter. Alternatively, as used in [5], the numerator and denominator coefficients could be polynomially dependent on a single scheduling parameter i.e.

$$\sigma_j(t) = \sigma^j(t), \quad \text{(A.6)}$$

where $\sigma(t)$ is the single measured scheduling parameter. A similar representation to (A.5) exists for $a_i^0(\boldsymbol{\sigma}(t))$.

Using (A.1) and (A.2) we can write:

$$\begin{aligned} A_0(\boldsymbol{\sigma}(t), q^{-1})y(t) &= B_0(\boldsymbol{\sigma}(t), q^{-1})q^{-m}u(t) \\ &\quad + A_0(\boldsymbol{\sigma}(t), q^{-1})H_0(\boldsymbol{\sigma}(t), q^{-1})\eta(t) \\ &= B_0(\boldsymbol{\sigma}(t), q^{-1})q^{-m}u(t) + v(t) \end{aligned} \quad \text{(A.7)}$$

This, in turn, can be written in linear regression form as:

$$y(t) = \boldsymbol{\phi}^T(t)\boldsymbol{\theta}_0 + v(t), \quad \text{(A.8)}$$

where the regressor vector is given by:

$$\boldsymbol{\phi}(t) = \boldsymbol{\varphi}(t) \otimes \boldsymbol{\sigma}(t) \quad \text{(A.9)}$$

where $\otimes$ is the Kronecker product and

$$\boldsymbol{\varphi}^T(t) = [-y(t-1), -y(t-2), \ldots, -y(t-n_a), \\ u(t-m), u(t-m-1), \ldots, u(t-m-n_b)] \quad \text{(A.10)}$$

and the true parameter vector is:

$$\boldsymbol{\theta}_0^T = [a^0_{1,0}, a^0_{1,1}, \ldots, a^0_{1,n_\sigma}, \ldots, a^0_{n_a,0}, a^0_{n_a,1}, \ldots, a^0_{n_a,n_\sigma}, \\ b^0_{0,0}, b^0_{0,1}, \ldots, b^0_{0,n_\sigma}, \ldots, b^0_{n_b,0}, b^0_{n_b,1}, \ldots, b^0_{n_b,n_\sigma}]. \quad \text{(A.11)}$$

Model representation

The model of the system is written as:

$$A(\boldsymbol{\sigma}(t), q^{-1})y(t) = B(\boldsymbol{\sigma}(t), q^{-1})q^{-m}u(t) \quad \text{(A.12)}$$

which allows the predictor:

$$\hat{y}(t|\boldsymbol{\theta}) = \boldsymbol{\phi}^T(t)\boldsymbol{\theta} \quad \text{(A.13)}$$

to be defined, along with the prediction error:

$$e_p(t, \boldsymbol{\theta}) = y(t) - \hat{y}(t|\boldsymbol{\theta}) = y(t) - \boldsymbol{\phi}^T(t)\boldsymbol{\theta}. \quad \text{(A.14)}$$

As consistency is being considered here, it will be assumed that the model order is always chosen equal to the true system order.

A.1.3 Non-noisy scheduling parameters

In this section the case of non-noisy scheduling parameters is considered. This assumption is normally made in the LPV identification literature, and is somewhat unrealistic unless the scheduling parameter is a function of the system input. It is considered here to illustrate some basic results of the analysis, before studying the more complete, but more complicated, case of noisy scheduling parameters in the next section. It also allows the polynomial scheduling parameter (A.6) to be analysed when the degree of the polynomial is

greater than 1, which is not possible in the next section. Moreover, in the case of a very low noise-to-signal ratio it might be possible to ignore the noise on the scheduling parameter.

One specific case that the following analysis does not apply to is that when the scheduling parameters is the system output $y(t)$. This case is excluded because the output is expressly considered noisy.

As in [5], we examine the case of minimising a quadratic criterion for the estimation of the system parameters i.e.

$$\hat{\boldsymbol{\theta}}^N = \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{t=0}^{N-1} e_p^2(t, \boldsymbol{\theta}). \tag{A.15}$$

This estimate can be evaluated using a number of well-known algorithms, including the recursive least squares algorithm as proposed in [5]. It is also possible to use the standard linear least squares algorithm for batch data, giving:

$$\hat{\boldsymbol{\theta}}_{LS}^N = \left[\frac{1}{N} \sum_{t=0}^{N-1} \boldsymbol{\phi}(t) \boldsymbol{\phi}^T(t) \right]^{-1} \frac{1}{N} \sum_{t=0}^{N-1} \boldsymbol{\phi}(t) y(t). \tag{A.16}$$

Replacing $y(t)$ in (A.16) with (A.8) gives:

$$\hat{\boldsymbol{\theta}}_{LS}^N = \boldsymbol{\theta}_0 + \left[\frac{1}{N} \sum_{t=0}^{N-1} \boldsymbol{\phi}(t) \boldsymbol{\phi}^T(t) \right]^{-1} \frac{1}{N} \sum_{t=0}^{N-1} \boldsymbol{\phi}(t) v(t). \tag{A.17}$$

For the estimate $\hat{\boldsymbol{\theta}}_{LS}^N$ to be consistent i.e. that $\hat{\boldsymbol{\theta}}_{LS}^N$ converges almost surely to $\boldsymbol{\theta}_0$, it is well-known that it is necessary that:

$$\text{i) } \lim_{N \to \infty} \frac{1}{N} \sum_{t=0}^{N-1} \boldsymbol{\phi}(t) \boldsymbol{\phi}^T(t) \text{ be nonsingular.} \tag{A.18}$$

$$\text{ii) } \lim_{N \to \infty} \frac{1}{N} \sum_{t=0}^{N-1} \boldsymbol{\phi}(t) v(t) = \mathbf{0} \quad \text{w.p. 1.} \tag{A.19}$$

Condition (A.18) is a persistency of excitation condition, which in the LPV case requires more attention, as discussed in [5] for the polynomial dependence case (A.6) and [64] for the more general case.

Condition (A.19) can be analysed using Corollary 2.1. We first note that $\boldsymbol{\phi}(t)v(t)$ contains terms such as $\sigma_j(t)u(t-p)v(t)$ and $\sigma_j(t)y(t-p)v(t)$. We then define:

$$\begin{aligned}
\mathbf{s}_1(t) &= \begin{bmatrix} \sigma_j(t)u(t-p) \\ \sigma_j(t)y(t-p) \\ v(t) \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ \sigma_j(t)H_0(\boldsymbol{\sigma}(t-p),q^{-1})\eta(t-p) \\ A_0(\boldsymbol{\sigma}(t),q^{-1})H_0(\boldsymbol{\sigma}(t),q^{-1})\eta(t) \end{bmatrix} \\
&\qquad + \begin{bmatrix} \sigma_j(t)u(t-p) \\ \sigma_j(t)G_0(\boldsymbol{\sigma}(t-p),q^{-1})u(t-p) \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} v_1^1(t) \\ v_2^1(t) \\ v_3^1(t) \end{bmatrix} + \begin{bmatrix} w_1^1(t) \\ w_2^1(t) \\ w_3^1(t) \end{bmatrix}
\end{aligned} \tag{A.20}$$

where $w_1^1(t)$ and $w_2^1(t)$ satisfy (2.39), due to the assumed LPV stability of $G_0(\boldsymbol{\sigma}(t),q^{-1})$ and the boundedness of $\sigma_j(t)$ and $u(t)$. Also $v_2^1(t)$ and $v_3^1(t)$ fit in with the desired form of (2.41) due to the assumed LPV stability of $H_0(\boldsymbol{\sigma}(t),q^{-1})$.

The components of $\mathbf{s}_1(t)\mathbf{s}_1^T(t)$ give, amongst others, $\sigma_j(t)u(t-p)v(t)$ and $\sigma_j(t)y(t-p)v(t)$. Then Corollary 2.1 means that it is possible to write:

$$\left\| \frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\phi}(t)v(t) - \frac{1}{N}\sum_{t=0}^{N-1} E\{\boldsymbol{\phi}(t)v(t)\} \right\|_F \to 0 \text{ w.p. } 1, \text{ as } N \to \infty. \tag{A.21}$$

Analysing the components of $E\{\boldsymbol{\phi}(t)v(t)\}$ we see that:

$$E\{\sigma_j(t)u(t-p)v(t)\} = \sigma_j(t)u(t-p)E\{v(t)\} = 0 \tag{A.22}$$

and

$$E\{\sigma_j(t)y(t-p)v(t)\} = \sigma_j(t)E\{[\boldsymbol{\phi}^T(t-p)\boldsymbol{\theta}_0 + v(t-p)]v(t)\} \quad \text{(A.23)}$$

which is only zero when either $v(t)$ is a zero-mean, white noise sequence, or $n_a = 0$. Neither of these cases occur often in practice, so the second sum in (A.21) will, in general, be non-zero, and thus also the first. The least squares method therefore typically does not give consistent parameter estimates.

In order to obtain consistent parameter estimates the IV technique can be used. The IV estimate is given by:

$$\hat{\boldsymbol{\theta}}_{IV}^N = \left[\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\zeta}(t)\boldsymbol{\phi}^T(t)\right]^{-1}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\zeta}(t)y(t) \quad \text{(A.24)}$$

where $\boldsymbol{\zeta}(t)$ is the IV vector, which must be correlated with the regressor $\boldsymbol{\phi}(t)$ but not with the noise $v(t)$ in order that the algorithm be consistent. It should be mentioned that, in order to keep the variance of the estimates low, the IV vector for LPV system identification should be a function of the scheduling parameters, as $\boldsymbol{\phi}(t)$ is. Good choices are either to use an auxiliary LPV model (identified by the LS method) to generate the IV vector, or use a second experiment.

A.1.4 Noisy scheduling parameters

In this section the more realistic case of the measured values of $\boldsymbol{\sigma}(t)$ being contaminated by noise will be examined. To do this, the noisy, measured scheduling parameter vector, $\boldsymbol{\sigma}_v(t)$, is expressed as the sum of a noise-free component and a noisy component i.e.

$$\boldsymbol{\sigma}_v(t) = \boldsymbol{\sigma}(t) + \mathbf{v}_\sigma(t). \quad \text{(A.25)}$$

The measured regressor vector is now given by:

$$\boldsymbol{\phi}_{\sigma_v}(t) = \boldsymbol{\phi}(t) + \boldsymbol{\phi}_{v_\sigma}(t), \quad \text{(A.26)}$$

where $\boldsymbol{\phi}(t)$ is as defined in (A.9) and

$$\boldsymbol{\phi}_{v_\sigma}(t) = \boldsymbol{\varphi}(t) \otimes \mathbf{v}_\sigma(t). \tag{A.27}$$

If the least squares algorithm is now used to estimate the parameters we have:

$$\hat{\boldsymbol{\theta}}_{LS}^N = \left[\frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\phi}_{\sigma_v}(t)\boldsymbol{\phi}_{\sigma_v}^T(t)\right]^{-1} \frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\phi}_{\sigma_v}(t)y(t) \tag{A.28}$$

where it can be seen that the measured version $\boldsymbol{\phi}_{\sigma_v}(t)$ is used instead of $\boldsymbol{\phi}(t)$. Substituting in (A.8) for $y(t)$ gives:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{LS}^N &= \left[\frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\phi}_{\sigma_v}(t)\boldsymbol{\phi}_{\sigma_v}^T(t)\right]^{-1} \frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\phi}_{\sigma_v}(t)(\boldsymbol{\phi}^T(t)\boldsymbol{\theta}_0 + v(t)) \\ &= \boldsymbol{\theta}_0 - \left[\frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\phi}_{\sigma_v}(t)\boldsymbol{\phi}_{\sigma_v}^T(t)\right]^{-1} \frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\phi}_{\sigma_v}(t)\left(\boldsymbol{\phi}_{v_\sigma}^T(t)\boldsymbol{\theta}_0 - v(t)\right).\end{aligned} \tag{A.29}$$

It is clear that this expression has an extra term over (A.17), which comes from the fact that not only the output $y(t)$ is noisy but also the scheduling parameter signal. This can be recognised as an errors-in-variables type problem.

For the estimate to be consistent we require, in a similar fashion to before, that:

$$\text{i)} \lim_{N\to\infty} \frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\phi}_{\sigma_v}(t)\boldsymbol{\phi}_{\sigma_v}^T(t) \text{ be nonsingular.} \tag{A.30}$$

$$\text{ii)} \lim_{N\to\infty} \frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\phi}_{\sigma_v}(t)\left(\boldsymbol{\phi}_{v_\sigma}^T(t)\boldsymbol{\theta}_0 - v(t)\right) = \mathbf{0} \quad \text{w.p. 1.} \tag{A.31}$$

As before, Condition (A.30) is a persistency of excitation condition.

Condition (A.31) can, again, be analysed using Corollary 2.1. We first consider the matrix $\boldsymbol{\phi}_{\sigma_v}(t)\boldsymbol{\phi}_{v_\sigma}^T(t)$. This contains the following terms:

$$(\sigma_j(t) + v_{\sigma,j}(t))y(t-p)v_{\sigma,i}(t)y(t-r), \tag{A.32}$$

$$(\sigma_j(t) + v_{\sigma,j}(t))y(t-p)v_{\sigma,i}(t)u(t-r) \tag{A.33}$$

and

$$(\sigma_j(t) + v_{\sigma,j}(t))u(t-p)v_{\sigma,i}(t)u(t-r). \tag{A.34}$$

Considering (A.32), it is clearly equal to:

$$\begin{aligned}
&\sigma_j(t)\left[G_0(\boldsymbol{\sigma}(t-p),q^{-1})u(t-p) + H_0(\boldsymbol{\sigma}(t-p),q^{-1})\eta(t-p)\right]\\
&\quad v_{\sigma,i}(t)\left[G_0(\boldsymbol{\sigma}(t-r),q^{-1})u(t-r) + H_0(\boldsymbol{\sigma}(t-r),q^{-1})\eta(t-r)\right]\\
&+ v_{\sigma,j}(t)\left[G_0(\boldsymbol{\sigma}(t-p),q^{-1})u(t-p) + H_0(\boldsymbol{\sigma}(t-p),q^{-1})\eta(t-p)\right]\\
&\quad v_{\sigma,i}(t)\left[G_0(\boldsymbol{\sigma}(t-r),q^{-1})u(t-r) + H_0(\boldsymbol{\sigma}(t-r),q^{-1})\eta(t-r)\right].
\end{aligned} \tag{A.35}$$

So, referring to Corollary 2.1, we can write:

$$\begin{aligned}
\mathbf{s}_2(t) = &\begin{bmatrix}
\sigma_j(t)H_0(\boldsymbol{\sigma}(t-p),q^{-1})\eta(t-p)\\
v_{\sigma,j}(t)G_0(\boldsymbol{\sigma}(t-p),q^{-1})u(t-p)\\
v_{\sigma,j}(t)H_0(\boldsymbol{\sigma}(t-p),q^{-1})\eta(t-p)\\
v_{\sigma,i}(t)G_0(\boldsymbol{\sigma}(t-r),q^{-1})u(t-r)\\
v_{\sigma,i}(t)H_0(\boldsymbol{\sigma}(t-r),q^{-1})\eta(t-r)\\
0
\end{bmatrix} +\\
&\begin{bmatrix}
0\\0\\0\\0\\0\\
\sigma_j(t)G_0(\boldsymbol{\sigma}(t-p),q^{-1})u(t-p)
\end{bmatrix} = \begin{bmatrix} v_1^2(t)\\ v_2^2(t)\\ v_3^2(t)\\ v_4^2(t)\\ v_5^2(t)\\ v_6^2(t)\end{bmatrix} + \begin{bmatrix} w_1^2(t)\\ w_2^2(t)\\ w_3^2(t)\\ w_4^2(t)\\ w_5^2(t)\\ w_6^2(t)\end{bmatrix}.
\end{aligned} \tag{A.36}$$

In order to write $v_3^2(t) = v_{\sigma,j}(t)H_0(\boldsymbol{\sigma}(t-p),q^{-1})\eta(t-p)$ and $v_5^2(t) = v_{\sigma,i}(t)H_0(\boldsymbol{\sigma}(t-r),q^{-1})\eta(t-r)$ it is necessary that $\mathbf{v}_\sigma(t)$ and $\eta(t)$ are uncorrelated. This condition is reasonable so long as the scheduling

parameter is not the system output $y(t)$. If they are correlated the expected value of their product is non-zero and does not satisfy the corollary's assumptions on the stochastic component. The ergodicity of the signals used in the identification method is, thus, not provable in this case.

Additionally it is not possible to establish the ergodicity of the signals when the scheduling parameter has the polynomial dependence discussed in (4.8) which is of a degree greater than 1. The reason is that we would have higher order moments of the noise term affecting $\sigma(t)$, which are non-zero mean. This, in turn, would imply that $v_{\sigma,j}(t)$ is non-zero mean and thus $v_2^2(t)$ and $v_4^2(t)$ are non-zero mean, violating the corollary's assumptions.

With these conditions in mind, we see that amongst the elements of $\mathbf{s}_2(t)\mathbf{s}_2^T(t)$ are all the cross-terms found in (A.35), and thus:

$$\Bigg\| \frac{1}{N} \sum_{t=0}^{N-1} \big[(\sigma_j(t) + v_{\sigma,j}(t)) y(t-p) v_{\sigma,i}(t) y(t-r) - E\{(\sigma_j(t) + v_{\sigma,j}(t)) y(t-p) v_{\sigma,i}(t) y(t-r)\} \big] \Bigg\|_F \to 0 \quad \text{w.p. 1, as } N \to \infty. \tag{A.37}$$

Similar results exist for (A.33) and (A.34), meaning that:

$$\left\| \frac{1}{N} \sum_{t=0}^{N-1} \boldsymbol{\phi}_{\sigma_v}(t) \boldsymbol{\phi}_{v_\sigma}^T(t) - \frac{1}{N} \sum_{t=0}^{N-1} E\{\boldsymbol{\phi}_{\sigma_v}(t) \boldsymbol{\phi}_{v_\sigma}^T(t)\} \right\|_F \to 0 \quad \text{w.p. 1, as } N \to \infty. \tag{A.38}$$

Considering, for example, $E\{(\sigma_j(t)+v_{\sigma,j}(t))y(t-p)v_{\sigma,i}(t)y(t-r)\}$ in (A.37) it contains terms like

$$E\big\{ \big(v_{\sigma,j}(t) G_0(\boldsymbol{\sigma}(t-p), q^{-1}) u(t-p)\big) \big(v_{\sigma,i}(t) G_0(\boldsymbol{\sigma}(t-r), q^{-1}) u(t-r)\big) \big\}$$

which are not necessarily equal to zero. This, with other terms, will mean that the second sum in (A.38) will not, in general, be zero. This implies that the first sum will not be zero either.

Now examining the second term in (A.31) we have that the vector $\boldsymbol{\phi}_{\sigma_v}(t)v(t)$ contains the terms

$$(\sigma_j(t) + v_{\sigma,j}(t))\, y(t-p)v(t) \quad \text{and} \quad (\sigma_j(t) + v_{\sigma,j}(t))\, u(t-p)v(t).$$

Again referring to Corollary 2.1 we can write:

$$\mathbf{s}_3(t) = \begin{bmatrix} \sigma_j(t)H_0(\boldsymbol{\sigma}(t-p), q^{-1})\eta(t-p) \\ v_{\sigma,j}(t)G_0(\boldsymbol{\sigma}(t-p), q^{-1})u(t-p) \\ v_{\sigma,j}(t)H_0(\boldsymbol{\sigma}(t-p), q^{-1})\eta(t-p) \\ A_0(\boldsymbol{\sigma}(t), q^{-1})H_0(\boldsymbol{\sigma}(t), q^{-1})\eta(t) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \sigma_j(t)G_0(\boldsymbol{\sigma}(t-p), q^{-1})u(t-p) \end{bmatrix} = \begin{bmatrix} v_1^3(t) \\ v_2^3(t) \\ v_3^3(t) \\ v_4^3(t) \\ v_5^3(t) \end{bmatrix} + \begin{bmatrix} w_1^3(t) \\ w_2^3(t) \\ w_3^3(t) \\ w_4^3(t) \\ w_5^3(t) \end{bmatrix}. \quad \text{(A.39)}$$

The components of $\mathbf{s}_3(t)\mathbf{s}_3^T(t)$ include the terms $(\sigma_j(t)+v_{\sigma,j}(t))y(t-p)v(t)$ and $(\sigma_j(t) + v_{\sigma,j}(t))u(t-p)v(t)$. So it is possible to write:

$$\left\| \frac{1}{N}\sum_{t=0}^{N-1} \boldsymbol{\phi}_{\sigma_v}(t)v(t) - \frac{1}{N}\sum_{t=0}^{N-1} E\{\boldsymbol{\phi}_{\sigma_v}(t)v(t)\} \right\|_F \to 0 \quad \text{w.p. 1, as } N \to \infty. \quad \text{(A.40)}$$

In (A.40) we have

$$E\{\boldsymbol{\phi}_{\sigma_v}(t)v(t)\} = E\{(\boldsymbol{\phi}(t) + \boldsymbol{\phi}_{v_\sigma}(t))v(t)\} \quad \text{(A.41)}$$

which contains terms similar to those found in (A.23), so, as in Section A.1.3, typically $E\{\boldsymbol{\phi}(t)v(t)\} \neq 0$. Thus the second sum, and so the first, in (A.40) will usually be non-zero. This, together with

the result above for the first matrix in (A.31), means that Condition (A.31) is not normally satisfied and again the estimates obtained using the least squares algorithm are usually not consistent.

The instrumental variables method can be used to find a consistent estimate of the parameters in this case also. The IV estimate is given by:

$$\hat{\boldsymbol{\theta}}_{IV}^{N} = \left[\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\zeta}(t)\boldsymbol{\phi}_{\sigma_v}^{T}(t)\right]^{-1}\frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\zeta}(t)y(t). \tag{A.42}$$

This time, however, not only should $\boldsymbol{\zeta}(t)$ be correlated with the regressor $\boldsymbol{\phi}(t)$ and not with the noise $v(t)$, it should also be uncorrelated with $\boldsymbol{\phi}_{v_\sigma}^{T}(t)$ in order that the algorithm be consistent.

A.1.5 Simulation results

Simulations are carried out to see how the identification techniques perform for a finite number of data.

Example 1

The case of non-noisy scheduling parameters is first tested. The true system is given by:

$$G_0(\boldsymbol{\sigma}(t), q^{-1}) = \frac{(b_0^0(\boldsymbol{\sigma}(t)) + b_1^0(\boldsymbol{\sigma}(t))q^{-1})q^{-1}}{1 + a_1^0(\boldsymbol{\sigma}(t))q^{-1} + a_2^0(\boldsymbol{\sigma}(t))q^{-2}}. \tag{A.43}$$

The coefficient dependence on the scheduling parameter is chosen as polynomial in a single parameter, as in (A.6), giving:

$$\begin{aligned}
a_1^0(\boldsymbol{\sigma}(t)) &= 1 - 0.5\sigma(t) + 0.2\sigma^2(t)\\
a_2^0(\boldsymbol{\sigma}(t)) &= 0.5 - 0.7\sigma(t) - 0.1\sigma^2(t)\\
b_0^0(\boldsymbol{\sigma}(t)) &= 0.5 - 0.4\sigma(t) + 0.01\sigma^2(t)\\
b_1^0(\boldsymbol{\sigma}(t)) &= 0.2 - 0.3\sigma(t) - 0.02\sigma^2(t).
\end{aligned}$$

The true vector of parameters is, thus:

$$\boldsymbol{\theta}_0^T = [1, -0.5, 0.2, 0.5, -0.7, -0.1, 0.5, -0.4, 0.01, 0.2, -0.3, -0.02]. \tag{A.44}$$

The signal $u(t)$ is chosen as a PRBS with a shift register of length 12, giving $N = 4095$ points, and an amplitude of 1. $\sigma(t)$ is selected as a sinusoid with a period equal to N. It oscillates in the interval (0,1). In order for the algorithm to be consistent it is necessary that the persistency of excitation conditions are met. With the choice of $u(t)$ and $\sigma(t)$ used, this is easily the case. The noise on the output is taken such that $\eta(t)$ is a zero-mean, normally distributed white noise with a variance of 0.005, and $H_0(\boldsymbol{\sigma}(t), q^{-1}) = 1$. The least squares and instrumental variables method are tested. The instrumental variable vector is taken as the regression vector using $y(t)$ calculated from the model formed using $\hat{\boldsymbol{\theta}}_{LS}^N$ i.e.

$$\boldsymbol{\zeta}(t) = \boldsymbol{\varphi}_{ls}(t) \otimes \boldsymbol{\sigma}(t) \tag{A.45}$$

where

$$\begin{aligned}\boldsymbol{\varphi}_{ls}^T(t) = [&-y_{ls}(t-1), -y_{ls}(t-2), \ldots, -y_{ls}(t-n_a),\\ &u(t-m), u(t-m-1), \ldots, u(t-m-n_b)]\end{aligned} \tag{A.46}$$

It therefore satisfies the conditions in Section A.1.3.

The simulations are carried out 200 times in order to estimate the bias and variance of the parameters. Table A.1 shows the results for the different methods. The expectation operators are estimated as averages over the simulations.

It can be seen from the table that the IV method can be considered to give unbiased parameter estimates compared to the LS method. The IV method does, however, have a larger variance, a known problem with these methods, which comes as a trade-off with their low computational complexity. This can often be improved using the Multistep Algorithm in [32].

A noise-free validation simulation is carried out for the $\boldsymbol{\theta}$ estimated, using different signals for $u(t)$ and $\sigma(t)$ to those used in identification. They are 2 uniformly distributed random sequences varying

	LS	IV
$\left(\boldsymbol{\theta}_0 - \hat{E}\left\{\hat{\boldsymbol{\theta}}^N\right\}\right)^T \left(\boldsymbol{\theta}_0 - \hat{E}\left\{\hat{\boldsymbol{\theta}}^N\right\}\right)$	1.3505	0.0164
$\hat{E}\left\{\left(\hat{\boldsymbol{\theta}}^N - \hat{E}\left\{\hat{\boldsymbol{\theta}}^N\right\}\right)^T \left(\hat{\boldsymbol{\theta}}^N - \hat{E}\left\{\hat{\boldsymbol{\theta}}^N\right\}\right)\right\}$	0.0112	0.0250
$\hat{E}\left\{\frac{1}{N}\sum_{t=0}^{N-1} e_p^2(t)\right\}$	0.0514	0.0099

Table A.1. Results for Example 1 with noise-free $\boldsymbol{\sigma}(t)$

in the intervals [-1,1] and [0,1] respectively, and of length 1024 points. The mean squared prediction error is calculated for each simulation and the average values over the 200 simulations for each method are shown in Table A.1. It can be seen that due to the fact that the parameters identified using the instrumental variables are consistent, the prediction errors achieved in the validation simulations are much smaller than those obtained with the LS method.

Example 2

A second simulation is done to examine the case where the scheduling parameter is contaminated by noise. To be in accordance with the conditions on the scheduling parameter in Section A.1.4 i.e. polynomial dependence no greater than affine, the system is given as:

$$G_0(\boldsymbol{\sigma}(t), q^{-1}) = \frac{(b_0^0(\boldsymbol{\sigma}(t)) + b_1^0(\boldsymbol{\sigma}(t))q^{-1})q^{-1}}{1 + a_1^0(\boldsymbol{\sigma}(t))q^{-1} + a_2^0(\boldsymbol{\sigma}(t))q^{-2}} \tag{A.47}$$

where

$$a_1^0(\boldsymbol{\sigma}(t)) = 1 - 0.5\sigma(t), \qquad a_2^0(\boldsymbol{\sigma}(t)) = 0.5 - 0.7\sigma(t),$$
$$b_0^0(\boldsymbol{\sigma}(t)) = 0.5 - 0.4\sigma(t), \qquad b_1^0(\boldsymbol{\sigma}(t)) = 0.2 - 0.3\sigma(t).$$

The true vector of parameters is this time, therefore, given by:

$$\boldsymbol{\theta}_0^T = [1, -0.5, 0.5, -0.7, 0.5, -0.4, 0.2, -0.3]. \tag{A.48}$$

The signals $u(t)$, $\sigma(t)$ and the noise on the output are chosen as before. The noise on $\sigma(t)$ is taken such that $\mathbf{v}_\sigma(t)$ is a zero-mean, normally distributed white noise with a variance of 0.0005. The least squares and instrumental variables method are tested. This time it is not possible to use the regression vector generated using the output of the model formed from $\hat{\boldsymbol{\theta}}_{LS}$ as an instrumental variable vector. This is because it would still use $\boldsymbol{\sigma}_v(t)$ from the first simulation. The instrumental variable vector is therefore formed by using the measured values of $y(t)$ and $\boldsymbol{\sigma}_v(t)$ from a second simulation, which has different realisations of $\eta(t)$ and $\mathbf{v}_\sigma(t)$, and are therefore not correlated with those in the first simulation.

The simulations are, again, carried out 200 times to estimate the statistical properties of the parameters. Table A.2 shows the bias and variance estimates for the parameters found using the different methods.

Again the results show that the IV method can be considered to give unbiased parameter estimates compared to the LS method, though once more the IV estimate's variance is larger.

Noise-free validation simulations are again done, using the same signals as in the validation in Example 1. The mean squared prediction errors are shown in Table A.2. Again we see that the prediction errors achieved using the instrumental variables method are, on average, much smaller than those obtained with the LS method.

A.1.6 Conclusions

In this appendix the consistency of certain LPV identification methods is investigated. It has been shown that ergodicity results, under

	LS	IV
$\left(\boldsymbol{\theta}_0 - \hat{E}\left\{\hat{\boldsymbol{\theta}}^N\right\}\right)^T \left(\boldsymbol{\theta}_0 - \hat{E}\left\{\hat{\boldsymbol{\theta}}^N\right\}\right)$	0.3161	0.0042
$\hat{E}\left\{\left(\hat{\boldsymbol{\theta}}^N - \hat{E}\left\{\hat{\boldsymbol{\theta}}^N\right\}\right)^T \left(\hat{\boldsymbol{\theta}}^N - \hat{E}\left\{\hat{\boldsymbol{\theta}}^N\right\}\right)\right\}$	0.000505	0.0024
$\hat{E}\left\{\frac{1}{N}\sum_{t=0}^{N-1} e_p^2(t)\right\}$	0.0376	0.0051

Table A.2. Results for Example 2 with noisy $\boldsymbol{\sigma}(t)$

certain conditions, can be applied to the signals in these LPV identification methods in order to carry out a consistency analysis, despite the fact that the stochastic disturbances are non-stationary.

When the scheduling parameter is noisy, it has been shown that an errors-in-variables type identification problem occurs and consistent estimates can be calculated using the IV method. In this case ergodicity was only demonstrable when the noise affecting the scheduling parameter and that affecting the output are uncorrelated. Furthermore, ergodicity is not provable when a polynomial dependence on the noisy scheduling parameter of degree greater than affine is present.

References

[1] H. S. Ahn, Y. Chen, and K.L. Moore. Iterative learning control: Brief survey and categorization. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1099–1121, Nov. 2007.

[2] H. S. Ahn, K. L. Moore, and Y. Chen. Kalman filter augmented iterative learning control on the iteration domain. In *IEEE American Control Conference*, pages 250–255, Minneapolis, U.S.A., 2006.

[3] S. Arimoto. Robustness of learning control for robotic manipulators. In *IEEE International Conference on Robotics and Automation*, pages 1528–1533, Cincinnati, Ohio USA, May 1990.

[4] S. Arimoto, S. Kawamura, and F. Miyazaki. Bettering operations of robots by learning. *Journal of Robotic Systems*, 1:123–140, 1984.

[5] B. Bamieh and L. Giarré. Identification of linear parameter varying models. *International Journal of Robust and Nonlinear Control*, 12:841–853, 2002.

[6] A. Benveniste, M. Metivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximation*. Springer-Verlag, Berlin and New York, 1990.

[7] J.R. Blum. Multidimensional Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 25(4):737 – 744, 1954.

[8] D.A. Bristow, M. Tharayil, and A.G. Alleyne. A survey of iterative learning control. *IEEE Control Systems Magazine*, 26(3):96–114, June 2006.

[9] M. Butcher, A. Karimi, and R. Longchamp. On the consistency of certain identification methods for linear parameter varying systems. In *17th IFAC World Congress*, Seoul, Korea, July 2008.

[10] M. C. Campi, A. Lecchini, and S. M. Savaresi. Virtual reference feedback tuning: A direct method for the design of feedback controllers. *Automatica*, 38:1337–1346, 2002.

[11] H. F. Chen. Almost sure convergence of iterative learning control for stochastic systems. *Sci. in China (Series F)*, 46(1):1–13, 2003.

[12] H. F. Chen and H. T. Fang. Output tracking for nonlinear stochastic systems by iterative learning control. *IEEE Transactions on Automatic Control*, 49(4):583–588, 2004.

[13] S. Devasia. Should model-based inverse inputs be used as feedforward under plant uncertainty? *IEEE Transactions on Automatic Control*, 47(11):1865–1871, 2002.

[14] B. G. Dijkstra. *Iterative Learning Control with applications to a wafer stage*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2003.

[15] A.T. Elfizy, G.M. Bone, and M.A. Elbestawi. Design and control of a dual-stage feed drive. *International Journal of Machine Tools and Manufacture*, 45(2):153 – 165, 2005.

[16] A. Faanes and S.Skogestad. Feedforward control under the presence of uncertainty. *European Journal of Control*, 10(1):30–46, 2004.

[17] G. Ferreres and C. Roos. Efficient convex design of robust feedforward controllers. In *44th IEEE Conference on Decision and Control*, pages 6460–6465, Seville, Spain, December 2005.

[18] S. Gunnarsson and M. Norrlöf. On the disturbance properties of high order iterative learning control algorithms. *Automatica*, 42(11):2031–2034, 2006.

[19] K. Hamamoto, T. Fukuda, and T. Sugie. Iterative feedback tuning of controllers for a two-mass-spring system with friction. *Control Engineering Practice*, 11:1061–1068, 2003.

[20] J. Hatonen. *Issues of algebra and optimality in Iterative Learning Control*. PhD thesis, University of Oulu, Finland, 2004.

[21] J. J. Hatonen, T. J. Harte, D. H. Owens, J. D. Radcliffe, P. L. Lewin, and E. Rogers. A new robust iterative learning control algorithm for application on a gantry robot. In *IEEE Conference on Emerging Technologies in Factory Automation*, pages 305–312, Lisbon, Portugal, 2003.

[22] G. Heinzinger, D. Fenwick, B. Paden, and F. Miyaziki. Robust learning control. In *28th IEEE Conference on Decision and Control*, pages 436–440, Tampa, Florida USA, December 1989.

[23] H. Hjalmarsson. Iterative feedback tuning - an overview. *International Journal of Adaptive Control and Signal Processing*, 16:373–395, 2002.

[24] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin. Iterative feedback tuning: Theory and application. *IEEE Control Systems Magazine*, pages 26–41, 1998.

[25] D. Hoover, R. Longchamp, and J. Rosenthal. Two-degree-of-freedom l_2-optimal tracking with preview. *Automatica*, 40(1):155–162, 2004.

[26] A. Karimi, M. Kunze, and R. Longchamp. Robust controller design by linear programming with application to a double-axis positioning system. *Control Engineering Practice*, 15(2):197–208, February 2007.

[27] A. Karimi, L. Mišković, and D. Bonvin. Iterative correlation-based controller tuning: Application to a magnetic suspension system. *Control Engineering Practice*, 11(9):1069–1087, 2003.

[28] A. Karimi, L. Mišković, and D. Bonvin. Iterative correlation-based controller tuning. *International Journal of Adaptive Control and Signal Processing*, 18(8):645–664, 2004.

[29] A. Lecchini, M. C. Campi, and S. M. Savaresi. Virtual reference feedback tuning for two degree of freedom controllers. *International Journal of Adaptive Control and Signal Processing*, 16:355–371, 2002.

[30] L.H. Lee and K. Poolla. Identification of linear parameter-varying systems using nonlinear programming. *Journal of Dynamic Systems, Measurement and Control*, 121:71–78, 1999.

[31] Z.G. Li, C.Y. Wen, Y.C. Soh, and Y.Q. Chen. Iterative learning control of linear parameterized varying uncertain systems. In *6th International Conference on Control, Automation, Robotics and Vision*, Singapore, December 2000.

[32] L. Ljung. *System Identification - Theory for the User*. Prentice Hall, NJ, USA, second edition, 1999.

[33] R.W. Longman. Iterative learning control and repetitive control for engineering practice. *International Journal of Control*, 73(10):930–954, 2000.

[34] M. Lovera, M. Verhaegen, and C.T. Chou. State space identification of MIMO linear parameter varying models. In *Proceedings of the International Conference on Mathematical Theory of Networks and Systems*, pages 839–842, Padova, Italy, 1998.

[35] R. Merry, R. van de Molengraft, and M. Steinbuch. The influence of disturbances in iterative learning control. In *IEEE Conference on Control Applications*, pages 974–979, Toronto, Canada, August 2005.

[36] L. Mišković, A. Karimi, and D. Bonvin. Correlation-based tuning of a restricted-complexity controller for an active suspension system. *European Journal of Control*, 9(1):77–83, January 2003.

[37] L. Mišković, A. Karimi, D. Bonvin, and M. Gevers. Correlation-based tuning of decoupling multivariable controllers. *Automatica*, 43(9):1481–1494, 2007.

[38] K. L. Moore. Iterative learning control: An expository overview. *Applied and Computational Controls, Signal Processing and Circuits*, 1(1):151–214, 1998.

[39] K. L. Moore. Multi-loop control approach to designing iterative learning controllers. In *37th IEEE Conference on Decision and Control*, pages 666–671, Tampa, Florida USA, December 1998.

[40] K.L. Moore and Y. Chen. A separative high-order framework for monotonic convergent iterative learning controller design. In *IEEE American Control Conference*, pages 3644–3649 vol.4, Denver, Colorado USA, 2003.

[41] M. Nemani, R. Ravikanth, and B.A. Bamieh. Identification of linear parametrically varying systems. In *34th IEEE Conference on Decision and Control*, pages 2990–2995, New Orleans, LA, USA, December 1995.

[42] M. B. Nevelson and R. Z. Hasminskii. *Stochastic Approximation and Recursive Estimation.* American Mathematical Society, Providence, Rhode Island, USA, 1973.

[43] M. Norrlöf. An adaptive iterative learning control algorithm with experiments on an industrial robot. *IEEE Transactions on Robotics and Automation*, 18(2):245–251, April 2002.

[44] M. Norrlöf and S. Gunnarsson. Disturbance aspects of iterative learning control. *Engineering Applications of Artificial Intelligence*, 14(1):87–94, 2001.

[45] M. Norrlöf and S. Gunnarsson. Time and frequency domain convergence properties in iterative learning control. *International Journal of Control*, 75(14):1114–1126, 2002.

[46] M. Norrlöf and S. Gunnarsson. Disturbance rejection using an ILC algorithm with iteration varying filter. *Asian Journal of Control*, 6(3):432–438, 2004.

[47] S. Panzieri and G. Ulivi. Disturbance rejection of iterative learning control applied to trajectory tracking for a flexible manipulator. In *European Control Conference*, pages 2374–2379, Rome, Italy, September 1995.

[48] E. Prempain and I. Postlethwaite. Feedforward control: a full information approach. *Automatica*, 37:17–28, 2001.

[49] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Stat.*, 22:400–407, 1951.

[50] S. S. Saab. Optimal selection of the forgetting matrix into an iterative learning control algorithm. *IEEE Transactions on Automatic Control*, 50(12):2039–2043, December 2005.

[51] S.S. Saab. A discrete-time stochastic learning control algorithm. *IEEE Transactions on Automatic Control*, 46(6):877–887, June 2001.

[52] S.S. Saab. On a discrete-time stochastic learning control algorithm. *IEEE Transactions on Automatic Control*, 46(8):1333–1336, August 2001.

[53] J. Shamma and M. Athans. Guaranteed properties of gain scheduled control for linear parameter-varying plants. *Automatica*, 27(3):559–564, 1991.

[54] T. Söderström. Errors-in-variables methods in system identification. In *14th IFAC Symposium on System Identification*, pages 1–19, Newcastle, Australia, 2006.

[55] H. Stark and J. W. Woods. *Probability, Random Processes and Estimation Theory for Engineers*. Prentice-Hall, New Jersey, U.S.A., 1986.

[56] M.K. Tao, R.L. Kosut, and A. Gurcan. Learning feedforward control. In *IEEE American Control Conference*, pages 2575–2579, Baltimore, Maryland USA, June 1994.

[57] R. Tóth. *Modeling and Identification of Linear Parameter-Varying System - an Orthonormal Basis Function Approach*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2008.

[58] K. Tsai, C. D. Schaper, and T. Kailath. Design of feedforward filters for improving tracking performances of existing feedback control systems. *IEEE Transactions on Control Systems Technology*, 12(5):742–749, September 2004.

[59] M. Uchiyama. Formulation of high-speed motion pattern of a mechanical arm by trial. *Transactions SICE (Society of Instrument and Control Engineers)*, 14(6):706–712, 1978.

[60] K. van Berkel, I. Rotariu, and M. Steinbuch. Cogging Compensating Piecewise Iterative Learning Control with application to

a motion system. In *IEEE American Control Conference*, pages 1275–1280, Denver, Colorado USA, July 2007.

[61] V. Verdult. *Nonlinear system identification: A state-space approach*. PhD thesis, University of Twente, Enschede, The Netherlands, 2002.

[62] V. Verdult and M. Verhaegen. Subspace identification of multivariable linear parameter-varying systems. *Automatica*, 38:805–814, 2002.

[63] V. Verdult and M. Verhaegen. Kernel methods for subspace identification of multivariable LPV and bilinear systems. *Automatica*, 41:1557–1565, 2005.

[64] X. Wei and L. Del Re. On persistent excitation for parameter estimation of quasi-LPV systems and its application in modeling of diesel engine torque. In *Proceedings of the 14th IFAC Symposium on System Identification*, pages 517–522, Newcastle, Australia, March 2006.

[65] Y. Ye and D. Wang. Zero phase learning control using reversed time input runs. *Journal of Dynamic Systems, Measurement and Control*, 127:133–139, March 2005.

Curriculum Vitae

Personal details

Name:	Mark Edward John BUTCHER
Date of birth:	23/02/1980
Nationality:	British
E-mail:	mejbutcher@yahoo.com

Education

07/2004 - 05/2009	Ph.D.; Automatic Control Laboratory, EPFL, Switzerland
10/1999 - 06/2003	M.Eng. in Engineering Science; University of Oxford, UK

Publications

Journal papers

1. M. Butcher, A. Karimi and R. Longchamp. A statistical analysis of certain iterative learning control algorithms. *International Journal of Control*, 81(1):156–166, January 2008.

2. A. Karimi, M. Butcher and R. Longchamp. Model-free precompensator tuning based on the correlation approach. *IEEE Transactions on Control Systems Technology*, 16(5):1013–1020, September 2008.

3. M. Butcher and A. Karimi. An analysis of iterative learning control algorithms using stochastic approximation theory. Submitted to *Automatica*

4. M. Butcher and A. Karimi. Data-driven tuning of linear parameter varying precompensators. Submitted to *International Journal of Adaptive Control and Signal Processing*

5. M. Butcher and A. Karimi. Iterative learning control for linear parameter varying systems with application to a linear motor system. Submitted to *IEEE/ASME Transactions on Mechatronics*

Conference papers

1. A. Karimi, M. Butcher and R. Longchamp. Model-free precompensator and feedforward tuning based on the correlation approach. In *44th IEEE Conference on Decision and Control and European Control Conference*, pages 4493–4498, Seville, Spain, December 2005.

2. M. Butcher, A. Karimi and R. Longchamp. A comparison of iterative learning control algorithms with application to a linear motor system. In *32nd Annual Conference of the IEEE Industrial Electronics Society*, Paris, France, November 2006.

3. M. Butcher, A. Karimi and R. Longchamp. On the consistency of certain identification methods for linear parameter varying systems. In *17th IFAC World Congress*, Seoul, Korea, July 2008.

4. M. Butcher, A. Karimi and R. Longchamp. Iterative learning control based on stochastic approximation. In *17th IFAC World Congress*, Seoul, Korea, July 2008.

5. M. Butcher, A. Karimi and R. Longchamp. Data-driven precompensator tuning for linear parameter varying systems. In *47th IEEE Conference on Decision and Control*, Cancun, Mexico, December 2008.

6. M. Butcher and A. Karimi. Iterative learning control for linear parameter varying systems. Submitted to *48th IEEE Conference on Decision and Control*, Shanghai, China, December 2009.