A Norm Optimal Approach to Time-Varying ILC with Application to a Multi-Axis Robotic Testbed

Kira L. Barton, Member, IEEE and Andrew G. Alleyne, Senior Member, IEEE

Abstract

In this paper, we focus on improving performance and robustness in precision motion control (PMC) of multi-axis systems through the use of Iterative Learning Control (ILC). A Norm Optimal ILC framework is used to design optimal learning filters based on design objectives. The paper contains two key contributions. The first half of the paper presents the norm optimal framework, including the introduction of an additional degree of design flexibility via time-varying weighting matrices. This addition enables the controller to take trajectory, position-dependent dynamics, and time-varying stochastic disturbances into consideration when designing the optimal learning controller. Explicit guidelines and analysis requirements for weighting matrix design are provided. The second half of the paper seeks to demonstrate the use of these guidelines. Using the design details provided in the paper, norm optimal learning controllers using time-invariant and time-varying weighting matrices are designed for comparison through simulation on a model of a multi-axis robotic testbed.

Index Terms

Learning Control Systems, Time-varying Systems, Design Methodology, MIMO Systems

I. INTRODUCTION

In this paper we present a method for improving the precision motion control (PMC) of multi-input multi-output (MIMO) motion systems that execute the same task repetitively. The repetitive nature of these systems enables a controller to learn from previous iterations and modify the control input for improved tracking performance [1]. Iterative learning control (ILC) is an adaptive feedforward control method which minimizes tracking errors through iterative updates to the control signal [2], [3]. The ILC process is illustrated in Fig. 1.

Current applications of ILC have generally been for systems in which the unmodelled dynamics and external disturbances occur throughout the time period, yielding linear time-invariant (LTI) learning controllers [4]. However, in many systems there are position- or time-varying dynamics that affect the performance or robustness of the system at different times throughout a single iteration. For these types of systems, it is beneficial to consider a time-varying controller which enables one to focus on the different position or time dynamics independently. Focusing on individual dynamics at different times throughout the iteration may result in a final outcome that not only improves tracking control but is more robust to time-varying disturbances.

This paper focuses on the design of time-varying iterative learning controllers in the norm optimal framework. The objective of these controllers is to address time-varying dynamics and
Fig. 1. Iterative learning control process. As the number of iterations increases, the feedforward control signal is determined and the error signal is minimized.

disturbances that affect the performance and/or robustness of a system that repeats the same process iteration after iteration. Design and analysis for the time-varying learning controllers uses the lifted domain since that is a natural domain for iterative processes and is well suited for time-varying systems.

The norm optimal ILC framework in the lifted domain is designed to minimize a quadratic optimization problem through the selection of weighting matrices targeting performance and robustness criteria [5]. The general norm optimal ILC approach for PMC on repetitive systems has been previously used to implement time-invariant weighting matrices that are designed to satisfy various constraints [4], [6], [7]. For example, a norm optimal control design for minimum and non-minimum phase systems is developed in [4]. In this paper, the use of time-invariant diagonal weighting matrices enables the control algorithm in the lifted domain to correspond to a filtering operation in the frequency domain. In [6], time-invariant weighting matrices tailored for process control applications which exhibit time and/or position dependent dynamics are introduced. Although these time-invariant weighting matrices have been shown to improve performance and robustness for process control systems which exhibit time and/or position dependent behavior [6], limiting the performance of the learning controller to the worst-case dynamics may result in a more conservative controller and a decrease in system performance.

Previous work has been done to address the deficiencies of time-invariant weighting matrices through the design of time-varying weighting matrices for specific applications, [7]–[9]. In [7], critical and non-critical sections of a given trajectory are weighted separately using a time-varying weighting matrix design. [10] introduces the use of time-varying weighting matrices for enhanced contour tracking, while [9] implements a time-varying weighting matrix to address robustness issues in systems subjected to high frequency trajectory tracking. While some work has been done on improving the trajectory design using ILC [11], this work focuses on trajectories for which the critical design criterion is to follow the given trajectory. On the basis of previous work in the area of norm optimal learning controllers using time-invariant and time-varying weighting matrices, the authors have identified two key issues: 1) development of a generalized structure for time-varying weighting matrices, and 2) description of a design methodology for designing time-varying weighting matrices.
This paper seeks to address issues 1) and 2) by presenting the general format and design of time-varying weighting matrices using the norm optimal framework. The weighting matrices are then implemented on a model of a multi-axis robotic testbed. The outline of this paper is as follows. Section II provides the class of systems which are considered in this work. The norm optimal framework with regards to ILC will be discussed in Section III, including guidelines for tuning the weighting matrices and the formulation of a typical cost function. The general structure and design methodology for time-varying weighting matrices using the norm optimal framework are presented in Section IV. The design and simulation results of a time-varying weighting matrix for improved performance are given in Section V. Sections VI and VII demonstrate norm optimal designs and simulation results for time-varying weighting matrices for enhanced robustness and performance in the presence of model uncertainty and external stochastic disturbances, respectively. Concluding remarks are given in Section VIII.

II. SYSTEM SETUP

In this paper we consider linear, causal, discrete-time MIMO systems, \( P \), given as

\[
P ≜ \begin{cases} 
x_j(k + 1) = A(k)x_j(k) + B(k)u_j(k) \\
δy_j(k) = C(k)x_j(k) + D(k)u_j(k), \\
y_j(k) = δy_j(k) + y_o(k) + d_j(k) 
\end{cases}
\]  
(1)

where \( k = 0, 1, \ldots, N − 1 \) is the discrete time index, \( j = 0, 1, \ldots \) is the iteration index, \( u_j(k) \in \mathbb{R}^{q_i} \) is the control, \( y_j(k) \in \mathbb{R}^{q_o} \) is the output, \( y_o(k) \in \mathbb{R}^{q_o} \) is iteration-invariant, \( d_j(k) \in \mathbb{R}^{q_o} \) corresponds to stochastic (iteration-varying) external disturbances, \( x_j(k) \in \mathbb{R}^n \) are system states, and \( (A(k), B(k), C(k), D(k)) \) are appropriately sized iteration-invariant real-valued matrices. It is assumed that \( x_j(0) = x_o \) for all \( j \), and note that \( y_o(k) \) can be used to capture iteration-invariant initial-condition responses [1], feedback control [12], and external disturbances. As illustrated by the matrices, \( (A(k), B(k), C(k), D(k)) \), \( P \) is defined as time-varying over a single profile, but iteration-invariant from trial-to-trial. In the lifted-domain [13], [14], the discrete-time behavior of the system is represented by its convolution matrix \( P \) using impulse response data \( H_{i,j}(k) \),

\[
P = \begin{bmatrix} H_{0,0} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ H_{N-1,0} & \cdots & H_{N-1,N-1} \end{bmatrix}.
\]  
(3)

For MIMO LTV systems, \( H_{i,j}(k) \) contains the impulse response from each of the \( q_i \) inputs to each of the \( q_o \) outputs and can be derived using the matrices in (1),

\[
H_{i,j} : \begin{cases} 
D(i), & i = j \\
C(i)A(i-1)A(i-2) \ldots A(j+1)B(j), & i > j.
\end{cases}
\]  
(4)

Given \( H_{i,j}(k) \in \mathbb{R}^{q_o \times q_i} \), system \( P \in \mathbb{R}^{Nq_o \times Nq_i} \) is a lower triangular matrix with a block Toeplitz structure. While the results presented in this paper are for an LTV system, the same design process can be applied to LTI systems. In the case of LTI systems, \( H_{i,j} \) is of the form,

\[
H_{i,j} : \begin{cases} 
D, & i = j \\
CA^{-j}B, & i > j.
\end{cases}
\]  
(5)
Many system models contain some form of model uncertainty. To address this in the controller design, assume that the true system $P_t$ corresponds to the nominal model $P$ plus an uncertainty $\Delta_P: P_t = P(I + \Delta_P)$, with the multiplicative uncertainty $\Delta_P = W\Delta$ and $||\Delta||_2 \leq 1$.

During trial $j$, system $P_t$ maps the input signal $u_j$ to the measured output signal $y_j$, i.e., $y_j = P_t u_j + d_j$, with $u_j$, $y_j$, and $d_j$ defined in (6), (7), and (8) respectively.

\begin{align*}
u_j &= [u_j^T(0) \ u_j^T(1) \ \cdots \ u_j^T(N-1)]^T \quad (6) \\
y_j &= [y_j^T(0) \ y_j^T(1) \ \cdots \ y_j^T(N-1)]^T \quad (7) \\
d_j &= [d_j^T(0) \ d_j^T(1) \ \cdots \ d_j^T(N-1)]^T, \quad (8)
\end{align*}

with
\begin{align*}
u_j^T(k) &= [u_j^1(k) \ \cdots \ u_j^{q_1}(k)] \\
y_j^T(k) &= [y_j^1(k) \ \cdots \ y_j^{q_0}(k)] \\
\text{and} \quad d_j^T(k) &= [d_j^1(k) \ \cdots \ d_j^{q_0}(k)].
\end{align*}

Here we adopt a widely used norm optimal ILC update law [14], [15]

\begin{equation}
u_{j+1} = Lu_j + Le_j \quad (9)
\end{equation}

where $y_r$ is the reference signal and is assumed iteration invariant. In (9), $L_u$ and $L_e$ are solutions to a quadratic optimization problem detailed shortly in Section III. These lifted matrices are generally non-causal, time-invariant filters on the control and error signals, respectively. The non-causality leads to full lifted matrices rather than the lower triangular toplitz form of the system $P$. In this paper, time-varying designs for these filters are used in order to address particular challenges at specific time intervals.

For MIMO systems consisting of two or more uncoupled axes, an additional error component known as the contour error can be identified. Contour tracking errors for the class of MIMO systems containing multiple uncoupled axes can be defined with respect to individual axis errors, $e_1, e_2, \ldots, e_{q_o}$, and trajectory dependent gains known as coupling gains [16], [17], $c_1(\theta, k)$, $c_2(\theta, k)$, \ldots, $c_{q_o}(\theta, k)$, where $k$ is the time interval from $k = 0, 1, \ldots, N-1$, $\theta$ is the instantaneous angle of the reference trajectory with respect to the first axis, and 1, 2, \ldots,$q_o$ are the individual uncoupled axes. Mathematically, for two axes this can be shown as [16]

\begin{equation}
e(k) = c_1(\theta, k) \cdot e_1(k) + c_2(\theta, k) \cdot e_2(k), \quad (11)
\end{equation}

where the matrix representation of (11) is given as

\begin{equation}
\varepsilon(k) = C(\theta, k) \cdot e(k) \quad (12)
\end{equation}

Linearized coupling gains for this 2 degree-of-freedom (DOF) example have the following format

\begin{equation}
c_1(\theta, k) = -\sin \theta(k); c_2(\theta, k) = \cos \theta(k). \quad (13)
\end{equation}

Note that the use of trajectory-dependent coupling gains leads to a time-varying controller.

Previous work in [8] introduced a specific norm optimal ILC design which reformats the general norm optimal cost function to enable the controller to focus on minimizing contour tracking.
errors. The objective of the work presented in this paper is to develop general time-varying design strategies in an effort to improve trajectory tracking performance and robustness through the use of time-varying weighting matrices in the norm optimal framework. The generalized time-invariant structure for this framework, along with some basic guidelines for tuning the design of the controller, is given in the following section.

III. NORM OPTIMAL ILC

In the previous section, a norm optimal ILC control algorithm was presented in (9). This controller results from a quadratic optimization problem, [18]. In this problem, we want to minimize an objective $J$, with $J$ corresponding to the sum of weighted norms of the error $||e_{j+1}||_Q$, the command signal $||u_{j+1}||_S$, and the rate of change of the command signal $||u_{j+1} - u_j||_R$, as shown in Eq. (14).

$$J = e_{j+1}^TQe_{j+1} + u_{j+1}^TSu_{j+1} + (u_{j+1} - u_j)^TR(u_{j+1} - u_j).$$ (14)

$(Q, R, S)$ are symmetric positive definite matrices with a common form given as $(Q, R, S) \triangleq (qI, rI, sI)$. Note that in some cases $(Q, R, S)$ may be positive semi-definite matrices, as long as $P^TQP + S + R$ is positive definite.

Applying the substitution $e_{j+1} = e_j - P(u_{j+1} - u_j)$, differentiating $J$ with respect to $u_{j+1}$, setting this derivative equal to zero, and rearranging the solution, yields the norm optimal ILC controller from (9) with respect to the weighting matrices $(Q, R, S)$ and the plant $P$,

$$u_{j+1} = Lu_j + Le_y, \quad L_u = (P^TQP + S + R)^{-1}(P^TQP + R), \quad L_e = (P^TQP + S + R)^{-1}P^TQ.$$ (15)

Note that for $(L_u, L_e)$ to ensure convergence we require $P^TQP + S + R$ to be positive definite, as will be shown in Section III-A.

Although this ILC control strategy is relatively well known [15], there has been relatively little documentation on how to tune $(Q, R, S)$ [6], [19]. Therefore, the following sections derive tuning guidelines by studying the properties of the ILC controlled system with respect to nominal convergence, robust convergence, and performance.

A. Nominal Convergence

In the following subsection, nominal convergence is explored for the nominal plant model $P$. Given the ILC controller (15) and the system dynamics $y_j = Pu_j$ (with $y_o = 0, d_j = 0$), the trial domain dynamics can be given by

$$u_{j+1} = (L_u - L_eP)u_j + L_ey_r.$$ (16)

For this system to be asymptotically stable, the spectral radius $\max_i |\lambda_i(L_u - L_eP)| < 1$ for $i = 1, 2, \ldots, N_{q_o}$ where $\lambda(\cdot)$ is the eigenvalue of $(\cdot)$. For most practical systems, asymptotic stability is not a strong enough condition. ILC systems that are asymptotically stable may experience large transients in the iteration domain prior to convergence [1]. Monotonic convergence is a
stronger stability requirement than asymptotic stability and minimizes the possibility of transient growth.

Monotonic convergence [3] of the control input requires that $\eta = \|L_u - L_e P\|_2 < 1$, where $\|A\|_2 = \sigma(A)$ and $\sigma(A)$ is the largest singular value of $A$. Note that $\max_i |\lambda_i(A)| \leq \|A\|_2$. Furthermore, $\|u_\infty - u_{j+1}\|_2 \leq \eta\|u_\infty - u_j\|_2$ gives the rate of convergence, where $\lim_{j \to \infty} u_j \triangleq u_\infty$.

For the norm optimal ILC controller, we have $L_u - L_e P = (P^T Q P + S + R)^{-1} R$. As a result, convergence is guaranteed for any symmetric positive semi-definite $(Q, R, S)$ with $P^T Q P + S + R$ positive definite. Note that the convergence speed strongly depends on $R$. For $\|R\|_2 = 0$ deadbeat control is achieved and as $\|R\|_2 \to \infty$ the convergence speed approaches zero.

B. Robust convergence

In this subsection, we consider the true system $P_t$ to correspond to the nominal model $P$ plus an uncertainty $\Delta_P$: $P_t = P(I + \Delta_P)$, with the multiplicative uncertainty $\Delta_P = W\Delta$ and $\|\Delta\|_2 \leq 1$ as defined in Section II. As a result, the requirement for robust convergence is

$$\|L_u - L_e P_t\|_2 < 1$$

(17)

$$\Rightarrow \max_\Delta \|(P^T Q P + S + R)^{-1}(R - P^T Q P \Delta_P)\|_2 < 1.$$

**Lemma 1:** Consider (17) with $\|R\| = 0$. Then a sufficient condition for robust convergence is given by $\|(P^T Q P + S)^{-1}P^T Q P W\|_2 < 1$.

**Proof:** Follows directly from (17) and the inequality:

$$\|(P^T Q P + S)^{-1}P^T Q P W\|_2 \leq \|P^T Q P + S\|_2 \|P^T Q P W\|_2 \|\Delta\|_2.$$

**Lemma 2:** Consider (17) with $\|(P^T Q P + S)^{-1}P^T Q P W\|_2 < 1$, and assume $P^T Q P + S$ symmetric and positive definite. Then robust convergence is guaranteed for all $R = rI$, $r \in \mathbb{R}$.

**Proof:** With $P^T Q P + S$ a symmetric positive definite matrix, its singular value decomposition equals $P^T Q P + S = U\Sigma U^T$ with $U$ a unitary matrix and $\Sigma$ diagonal and of full rank with diagonal elements $\sigma_i$. Furthermore, with $Z \triangleq (U\Sigma U^T)^{-1}P^T Q P W$ and $\|(U\Sigma U^T)^{-1}P^T Q P W\|_2 \triangleq \alpha < 1$ we have $\|Z\|_2 = \alpha < 1$. Therefore:

$$\max_\Delta \|(P^T Q P + S + R)^{-1}(R - P^T Q P W \Delta)\|_2$$

$$= \max_\Delta \|(U\Sigma U^T + rI)^{-1}(rI + P^T Q P W \Delta)\|_2$$

$$= \max_\Delta \|(U\Sigma U^T + rI)^{-1}(rI + U\Sigma U^T Z \Delta)\|_2$$

$$= \max_\Delta \|(\Sigma + rI)^{-1}(rU^T + \Sigma U^T Z \Delta)\|_2$$

$$\leq \|(\Sigma + rI)^{-1}(rI + \alpha \Sigma)\|_2$$

$$= \max_i \frac{\alpha \sigma_i + r}{\sigma_i + r}$$

$$< 1, \forall r \in \mathbb{R} \geq 0.$$

■
From Lemma 2, we can conclude that the design parameter \( R = rI \) does not influence the robust convergence properties of the ILC controlled system. \( S \) on the other hand, should be designed such that the robust convergence condition in Lemma 1 holds. Similar statements and conclusions have been provided in [20].

C. Performance

To study performance, we study the steady state error \( e_{ss} \triangleq \lim_{j \to \infty} e_j \triangleq e_\infty \). The existence of \( e_\infty < \infty \) implies a convergent ILC controlled system (16). However, with \((L_u, L_e)\) requiring \( P^TQP + S + R \) to be positive definite, convergence is guaranteed, as given in subsection III-A.

The steady state error is derived from the steady state command signal \( u_{ss} \) (18). If the controller is asymptotically stable and \((I - L_u + L_eP)\) is nonsingular, then the steady state control can be found as

\[
    u_{ss} \triangleq \lim_{j \to \infty} u_j \triangleq u_\infty = (I - L_u + L_eP)^{-1}L_e y_r
\]

With (18), \( e_\infty = y_r - Pu_\infty \), and \( d_j(k) = 0 \) the steady state error is given by,

\[
    \lim_{j \to \infty} e_j \triangleq e_\infty = (I - P(P^TQP + S)^{-1}P^TQ)y_r.
\]

From (19), we can now conclude the following: the smallest possible error, considered optimal performance in the ILC literature, requires \( \|S\|_2 = 0 \) and hence \( P^TQP \) to be positive definite. Furthermore, \( e_\infty \) in (19) is not a function of \( R \), and hence performance is not a function of convergence speed in the absence of external stochastic disturbances.

In order to extend performance aspects of norm optimal ILC by including trial varying or stochastic disturbances \( d_j \), consider \( e_j = y_r - Pu_j - d_j \). Substituting \( e_j \) into (16) yields the iteration domain closed loop update law with stochastic disturbances,

\[
    u_{j+1} = (L_u - L_eP)u_j + L_e y_r - L_e d_j.
\]

If \( d_j \) is bounded, then the asymptotic stability condition becomes the bounded-input, bounded-output stability condition for stochastic disturbances. Note that \( d_j \) is filtered by \( L_e \) in (20), and thus one would expect the stochastic disturbance sensitivity to decrease when the gain \( L_e \) is reduced. From (15) minimizing \( \|L_e\|_2 \) requires changes to \( S, R \), or a combination of the two. Additionally, as was shown in [9], the influence of stochastic disturbances can be reduced by reducing the convergence speed. Given that convergence speed is highly dependent on \( R \) and yet \( R \) does not affect performance, it is the natural candidate for reducing stochastic disturbance sensitivity.

Note that the steady state solution for \( e_j \) is a function of \( d_j \) for any given learning filter.

\[
    e_\infty = (I - P(P^TQP + S)^{-1}P^TQ)y_r
\]

Hence, after convergence the error \( e_\infty \) will continue to fluctuate. In general, a larger \( \|R\|_2 \) will result in smaller fluctuations in \( e_\infty \). In the presence of stochastic disturbances, a compromise must be made between having a large \( \|R\|_2 \) to minimize disturbance effects, while also maintaining acceptable learning rates through a small \( \|R\|_2 \).
D. Tuning Guidelines for Time-Invariant Weighting Matrices

Based on the previous subsections, the following design tuning guidelines [21] for norm-optimal ILC control are given. The guidelines are design heuristics which can be used as a starting point for weighting matrix design. Similar to other design techniques (i.e. Ziegler-Nichols tuning rules [22]), the initial selection of the weighting matrices may not result in a controlled system which exhibits the desired design criteria. In this situation, the tuning may need to be refined over multiple designs.

Note that selection of $(S, R)$ should occur before a sequence of trials, not between trials. This minimizes the possibility of selecting a combination of $(S, R)$ weighting matrices which may result in an undesirable performance or even unstable system. This also allows one to accurately determine the effect of the variation in the $(S, R)$ weighting matrices on the system.

These guidelines are most easily implemented using common $qI$, $sI$, $rI$ diagonal type real-valued scalar gains. The guidelines are given as follows.

s1) Design $Q$: $Q$ corresponds to the desired weighting of the error. Generally let $Q = I$ for uniform weighting of the individual axis errors.

s2) Design $S$: The actual system dynamics will not usually be perfectly captured by the system model. Thus, $S$ must be designed such that the system is robustly monotonically convergent. Start with an $S$ yielding $||S||_{i2} \approx 0.01||P||_{i2}$. Note, the critical design parameter is the size of $||S||_{i2}$ relative to the size of $||P||_{i2}$, where the magnitude of $||P||_{i2}$ is related to system uncertainty through $P \Delta P$. Subsequently reduce $||S||_{i2}$ until the system diverges. Set $||S||_{i2} = 2 \cdot ||S||_{i2}^{min}$ to allow for a safety factor of 2.

s3) Design $R$: When stochastic disturbances are present, steady state error fluctuations will occur. Start with $||R||_{i2} = 0$ and increase $||R||_{i2}$ until the steady state error fluctuations are within desired bounds, or the root mean square (RMS) error does not decrease anymore.

s4) Iterate process: If ILC performance does not meet design specifications, one may need to go as far back as system identification to determine a smaller model uncertainty. Repeat design steps s1) - s3) until the ILC performance is within desired convergence and performance requirements.

These tuning guidelines provide a general approach for designing a norm optimal learning controller using time-invariant weighting matrices. The relationships between the $(Q, S, R)$ weighting matrices and nominal convergence, robust convergence, and performance criteria for a given system are maintained irrespective of the use of time-invariant or time-varying weighting matrices. However, when designing norm optimal learning controllers using time-varying weighting matrices a more structured design approach needs to be followed. The general format and design methodology for time-varying weighting matrices are provided in the following section.

IV. TIME-VARYING WEIGHTING MATRICES

The previous section presented the basic format and design methodology for norm optimal ILC using time-invariant weighting matrices. This section introduces a novel format for time-varying weighting matrices which enables one to weight individual and coupled signals independently, as well as provide an overall gain that weights the error, control, and change in control signals with respect to each other. A four step design methodology for building time-varying weighting matrices is presented in subsection IV-B.
A. Motivation

As the previous section stated, the weighting matrices are generally of the form $(Q, R, S) =: (qI, rI, sI)$. While this approach works well for time-invariant unmodelled dynamics and external disturbances, in many manufacturing systems the disturbances, dynamics, and tracking errors are time and position dependent. For these systems, time-varying weighting matrices of the form $(Q_{tv}, S_{tv}, R_{tv})$ are better able to address specific design requirements at specific time locations throughout the trajectory. Time-varying weighting matrices result in a modified optimization cost function,

$$J = e_{j+1}^T Q_{tv} e_{j+1} + u_{j+1}^T S_{tv} u_{j+1} + (u_{j+1} - u_j)^T R_{tv} (u_{j+1} - u_j),$$

where the generalized structures of $(Q_{tv}, S_{tv}, R_{tv})$ for multi-axis systems are given below,

$$Q_{tv} = \Sigma_Q \cdot [\Gamma_1 Q + \Gamma_2 Q \cdot C_Q^T C_Q]$$

$$S_{tv} = \Sigma_S \cdot [\Gamma_1 S + \Gamma_2 S \cdot C_S^T C_S]$$

$$R_{tv} = \Sigma_R \cdot [\Gamma_1 R + \Gamma_2 R \cdot C_R^T C_R].$$

In (23)–(25), the $C_{(\cdot)}$ matrices contain the terms used to define coupling between the individual signals of a MIMO system. The coupling can be determined with respect to the tracking profile, the physical system, or to user defined relationships between the signals. For example, $C_Q$ corresponds to the coupling matrix used to define contour error with respect to the individual axis errors, (12), as a function of the reference trajectory. The matrices $\Gamma_1(\cdot)$ and $\Gamma_2(\cdot)$ refer to the amount of weighting applied to the coupled or individual signals, respectively. These matrices are of the forms provided in (26) and (27), where the inner block diagonal matrices are shown for a 2 DOF system.

$$\Gamma_1(\cdot) = \begin{bmatrix} 
\gamma(1) & 0 & 0 \\
0 & \gamma(1) & 0 \\
0 & 0 & \gamma(N)
\end{bmatrix},$$

$$\Gamma_2(\cdot) = \begin{bmatrix} 
1 - \gamma(1) & 0 & 0 \\
0 & 1 - \gamma(1) & 0 \\
0 & 0 & 1 - \gamma(N)
\end{bmatrix}.$$

As can be seen from (26) and (27), the individual elements in $\Gamma_1(\cdot)$ and $\Gamma_2(\cdot)$ are directly related. The gain $\gamma(\cdot)(k)$ is used to determine what portion of the overall weighting is applied to the individual and coupled signals, respectively. From (26) and (27), $(\gamma(\cdot)(k) = 1)$ refers to all of the weighting being applied to the individual signals, while $(\gamma(\cdot)(k) = 0)$ results in only...
the coupled signals being weighted. The gain matrix $\Sigma_\cdot$ determines the overall weighting on the error signals, control signals, or change in control signals and is of the form shown in (28). Note that the inner diagonal matrix is illustrated for a 2 DOF system.

$$\Sigma_\cdot = \begin{bmatrix} \sigma(\cdot)(1) & 0 & \cdots & 0 \\ 0 & \sigma(\cdot)(1) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma(\cdot)(N) \end{bmatrix}$$

(28)

The gains $(\sigma_Q(k), \sigma_S(k), \sigma_R(k))$ are similar to the gains $(q, s, r)$ from the time-invariant form of the weighting matrices $(Q, R, S) =: (qI, rI, sI)$ in that they weight the different components in the cost function. Therefore, the tuning rules identified in Section III-D can be used to design the gains $(\sigma_Q(k), \sigma_S(k), \sigma_R(k))$, respectively. Note that while the gain matrix $\Sigma_\cdot$ could be absorbed into the gain matrices $\Gamma_1\cdot$ and $\Gamma_2\cdot$, it is kept separate in this paper to add an additional design component which enables more direct intra element weighting within the cost function.

Each of the design elements in (23)–(25) offers a means of weighting different aspects of the error signals, control signals, and change in control signals. A four step design methodology providing details on each of the individual terms in (23) is given in subsection IV-B. While this methodology is presented for the $Q^{tv}$ weighting matrix, the same process can be applied to the design of $S^{tv}$ and $R^{tv}$ weighting matrices.

**B. Design Methodology**

When it comes to designing time-varying weighting matrices for multi-axis systems it is important to maintain design flexibility to allow for a variety of system dynamics, applications, and external environments. The format presented in (23)–(25) provides the framework for considering: the coordination between the signals $C_\cdot$, the importance of the individual versus coupled signals $\Gamma_1\cdot$ and $\Gamma_2\cdot$, and the weighting on the signals as a whole $\Sigma_\cdot$. The design procedure given below and detailed in Fig. 2 presents a four step methodology for designing a time-varying $Q^{tv}$ weighting matrix.

s1) Design $C_Q$: $C_Q$ corresponds to the desired coupling or coordination between the individual error signals. Generally, for the weighting on the error signals, let $\varepsilon = C_Q \cdot e$ where $C_Q$ contains the coupling gains determined from the reference trajectory (13).

s2) Design $\Gamma_1Q$ and $\Gamma_2Q$: As mentioned previously, $\Gamma_1\cdot$ and $\Gamma_2\cdot$ refer to the weighting gain matrices applied to the coupled and individual error signals, respectively. Set $\gamma_Q(k) = 1$ in (26) and (27) at the discrete times where time and position synchronization are critical. Set $\gamma_Q(k) = 0$ when the synchronization between time and position is not critical and the emphasis is on contour tracking.

s3) Design $\Sigma_Q$: $\Sigma_Q$ provides an overall weighting on the error. The design of $\Sigma_Q$ should follow the tuning guidelines presented in Section III-D with respect to the correlation between Q and performance. Generally, set $\sigma_Q(k) = 1$ for baseline tracking and increase the gain in the locations where more emphasis on trajectory tracking is required, i.e. in the corners of rastered trajectories. Continue increasing the gain until the system performance begins to display transient behavior. Set $\sigma_Q(k) = \frac{1}{2} \cdot \sigma_Q^{max}(k)$ to allow for a safely factor of 2.
s4) Iterate process: Evaluate the performance of the controlled system. If it does not meet design specifications, repeat design steps s1) - s3) until the ILC performance is within desired convergence and performance requirements.

$$Q^{tv} = \Sigma Q \left[ \Gamma_1 + \Gamma_2 \cdot C_q^T C_q \right]$$

Step 1: Determine the signal coupling
\( C_q^T C_q \)

i.e. contour error: \( \tau = C_q e \)

Step 2: Determine the gain matrices weighting coupled and individual signals
Gain Matrices: \( \Gamma_1, \Gamma_2 \)

Step 3: Determine gain matrix for overall weighting of the error signal
Gain Matrix: \( \Sigma Q \)

Evaluate performance

Design requirements satisfied

Finished

Design requirements not satisfied

Iterate Process

Fig. 2. Time-varying Weighting Matrix Design Methodology.

As mentioned previously, this type of design approach can be applied when designing the \( S^{tv} \) and \( R^{tv} \) weighting matrices with respect to the control signals or change in control signals, respectively. Time or position dependent weighting matrix designs allow the controller to maximize the performance and robustness of the system without being overly conservative or becoming unstable. The next three Sections (V,VI,VII) demonstrate the design methodology presented in Fig. 2 for generating time-varying weighting matrices of the form given in (23)–(25).

In order to validate the performance and robustness improvements attributed to the design of \( Q^{tv} \), \( S^{tv} \), and \( R^{tv} \) weighting matrices, the next three sections also include results obtained from simulating time-varying norm optimal controllers (29) with models of a multi-axis robotic testbed.

$$u_{j+1} = L_u^{tv} u_j + L_e^{tv} e_j$$

$$L_u^{tv} = (P^T Q^{tv} P + S^{tv} + R^{tv})^{-1}(P^T Q^{tv} P + R^{tv})$$

$$L_e^{tv} = (P^T Q^{tv} P + S^{tv} + R^{tv})^{-1} P^T Q^{tv}.$$

V. DESIGN EXAMPLE 1: TIME-VARYING Q WEIGHTING MATRIX

The previous section presented the basic format and design methodology for developing norm optimal learning controllers using time-varying weighting matrices. This section focuses on improving trajectory tracking through a time-varying design of the Q weighting matrix. As shown in (14), Q applies weighting to the error signals, thereby influencing the tracking performance directly. The Q weighting matrix is time-varied based on the reference trajectory and the initial
error signals without learning, i.e. iteration $j = 0$. Additionally, the time-varying format of the weighting matrix (23) introduces a method for weighting the coordination of the error signals through the use of the $C_Q$ matrix. For $Q^{tv}$, $C_Q$ corresponds to the $C$ matrix given in the definition for contour error (12), as will be illustrated in the following subsection.

A. Design Step 1

Using the general time-varying weighting matrix form given in (23), there are three key steps to designing a time-varying $Q^{tv}$ weighting matrix. The first step requires the derivation of the coupling matrix, $C_Q$. For the error element of the cost function (22), the coupling of the individual error signals comes in the form of an additional error component known as the contour error introduced in Section II. Applying the lifted approach to the definition given in (12) results in the lifted form of contour error,

$$\varepsilon = C \cdot e,$$

where $C$ is a lifted matrix containing the trajectory dependent, time-varying coupling gains. The term $C_Q^T C_Q$ used in (23) can now be written as,

$$C_Q^T C_Q = \begin{bmatrix}
C_T(\theta, 0)C(\theta, 0) & 0 & \cdots \\
0 & \ddots & \\
0 & \cdots & C_T(\theta, N-1)C(\theta, N-1)
\end{bmatrix},$$

where $C_T(\theta, k)C(\theta, k)$ for a 2 DOF system is defined as,

$$C_T(\theta, k)C(\theta, k) = \begin{bmatrix}
c_1(\theta, k)c_1(\theta, k) & c_1(\theta, k)c_2(\theta, k) \\
c_2(\theta, k)c_1(\theta, k) & c_2(\theta, k)c_2(\theta, k)
\end{bmatrix}.$$  

Using linearized coupling gains for the 2 DOF system, $C_Q^T C_Q$ results in a matrix which is time-varying and block diagonal. A weighting matrix of this form only weights certain combinations of individual error signals rather than each signal equally. By only weighting certain combinations, the system is free to generate different combinations of individual axis errors which minimize the contour errors while potentially increasing the individual axis errors. This process can be described as effectively decoupling position and time for each individual axis tracking task in order to focus on minimizing the contour tracking errors. [8] illustrates the enhanced trajectory tracking capabilities that result from the use of $C_Q^T C_Q$.

With $C_Q^T C_Q$ defined in (31), the next step in the design process is to determine the gains $\gamma_Q(k)$ and $(1 - \gamma_Q(k))$ for all $k$ which refer to the weighting gains applied to the contour or individual axis tracking, respectively. While generally constant, these gains can be varied throughout the trajectory using shaping criteria based on the reference trajectory.

B. Design Step 2

In order to explore the performance benefits of a time-varying $Q$ weighting matrix, we consider a rastered reference trajectory shown in Fig. 3. This type of trajectory is commonly used in atomic force microscopy (AFM), as well as other manufacturing systems which require sharp transitions between signals. Sections $A$ and $C$ of Fig 3 correspond to locations where the learning controller focuses on minimizing contour tracking by relinquishing position and time synchronization for
each axis, while in section $B$ the learning controller is designed to improve individual axis tracking and reestablish position and time synchronization [10]. The high acceleration transition points, identified using circles on Fig. 3, correspond to locations within the trajectory where the demanding acceleration requirements result in increased trajectory tracking errors as illustrated in Fig. 4 [23]. These areas indicate potential opportunities for large $\sigma_Q(k)$ gains to provide improved tracking capabilities as compared to small $\sigma_Q(k)$ gains. This will be explored in the third design step.

![Fig. 3. Raster trajectory containing linear sections (B), contoured sections (A,C), high acceleration sections ($t_1 - t_6$), and low acceleration sections.](image)

![Fig. 4. Initial contour tracking errors without the use of learning. Individual axis errors show a similar trend in peak locations for initial tracking errors.](image)

Consider the trajectory in Fig. 3. The use of the matrix $C_Q^T C_Q$ to enable weighting on the
coupled error signals has been shown to result in the most improved contour or trajectory tracking performance for rasters, such as sections A and C [8]. Focusing on the individual axis errors by selecting the gain $\gamma_Q(k) = 1$ for the gain matrices $\Gamma_1 Q$ and $\Gamma_2 Q$ in (23) can be used to reestablish position and time synchronization by focusing on minimizing the individual axis errors during the linear sections. In order to maximize the tracking performance, while maintaining position and time synchronization at the start of each raster, the gains are selected as $\gamma_Q(k) = 0$ and $1 - \gamma_Q(k) = 1$ in sections A and C, while the gains $\gamma_Q(k) = 1$ and $1 - \gamma_Q(k) = 0$ are used in section B.

To facilitate smooth transitions between the sections, time-varying vectors containing $\gamma_Q(k)$ and $1 - \gamma_Q(k)$ for $k = 1, \ldots, N - 1$ are filtered using a lowpass Gaussian filter with a bandwidth of 15 Hz. Although any lowpass filter type could be used, in this work we use a Gaussian filter because it is a symmetric filter in which the filter coefficients can be defined with respect to the bandwidth. This results in gain vectors of the form shown in Fig. 5.

![Figure 5](image)

**Fig. 5.** Alternating gains $\gamma_Q(k)$ and $1 - \gamma_Q(k)$ to switch between weighting individual versus coupled error signals.

### C. Design Step 3

While the gains $\gamma_Q(k)$ and $1 - \gamma_Q(k)$ focus on weighting the coupled versus individual error signals for a given trajectory, the third step in the design process focuses on establishing weighting on the error signal $(e_{j+1}(k))$ as a whole. The overall weighting on the error depends on the initial error signal. The locations where the error signal is large correspond to locations in the trajectory that challenge the performance of the system, i.e. the high acceleration sections $(t_1 - t_6)$. For the trajectory given in Fig. 3, the high and low acceleration components of the reference trajectory can be addressed using the $\Sigma Q$ gain matrix.

In a time-invariant norm optimal control design, the scalar weighting $\sigma_Q$ represents constant performance weighting on the error throughout the entire iteration. Figure 4 clearly indicates the locations where increased emphasis on the error (larger $\sigma_Q$ gains) may result in better trajectory tracking. From this information, the locations where the $\sigma_Q(k)$ gain will be increased from 1 to 30 have been identified as $t_1, t_2, t_3, t_4, t_5$, and $t_6$. As with the previous time-varying weighting
gains, the transitions between high and low gains are smoothed out using a lowpass Gaussian filter with a 15 Hz bandwidth. The modified time-varying $\sigma_Q(k)$ profile is given in Fig. 6.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig6}
\caption{Profile for the diagonal elements in the weighting matrix $\Sigma_Q$. Notice that the gain is increased in the locations corresponding to the high acceleration sections $(t_1 - t_6)$.}
\end{figure}

Following general protocol, the baseline value for the overall weighting gain was set to $\sigma_Q(k) = 1$. The overall weighting gains $\sigma_S(k)$ and $\sigma_R(k)$ were designed with respect to this baseline value; therefore, increasing $\sigma_Q(k)$ disrupts the relationship between these gains and care must be taken to ensure the controller meets performance and robustness requirements. While the trajectory tracking performance generally improves with the use of time-varying weighting matrices, there may be a tradeoff between tracking in the low versus high acceleration sections. The value of the gain during the high acceleration sections was chosen by increasing the value until the simulation performance of the system began to display transient behavior, and then reducing the gain to provide for a safety factor on the real system. For this system we chose a safety factor of 2.

Combining the three design steps results in a time-varying weighting matrix that addresses individual versus coupled axis errors, as well as overall error performance requirements.

\subsection*{D. Simulation Results}

In this subsection we apply the time-varying weighting matrix derived from combining the three design steps described in subsections V-A, V-B, and V-C on a model of the multi-axis robotic testbed illustrated in Fig. 7. For simulation purposes, 1-kHz sampled dynamic models of the $x$ and $y$ axes, along with stabilizing feedback controllers, were developed in [24]. Numerical values identified for the plant models along with controller coefficients can be found in Appendix B.

\begin{equation}
P_i(z) = \frac{K(z + \alpha_{i1})(z^2 - \alpha_{i2}z + \alpha_{i3})(z^2 - \alpha_{i4}z + \alpha_{i5})}{(z - \beta_{i1})(z - 1)(z^2 - \beta_{i2}z + \beta_{i3})(z^2 - \beta_{i4}z + \beta_{i5})}, \quad i = x, y.
\end{equation}
feedback controller $\triangleq k_{pi}(z) = \frac{k(z - \alpha_{i1})(z - \alpha_{i2})(z - \alpha_{i3})}{(z - \beta_{i1})(z - \beta_{i2})(z - \beta_{i3})}$, $i = x, y$. (34)

The following results were obtained using the stabilized dynamic models from (33) and (34), the reference trajectory from Fig 3 ($N = 1300$), and the norm optimal controllers (15) with $Q^w$ replacing $Q$. Using the tuning guidelines from section III-D and designing for the multi-axis system of Fig. 7, the scalar gains for $S = sI$ and $R = rI$ were heuristically chosen as $(s = 1e^{-2}, r = 2e^{-2})$.

Figure 8 illustrates the effect of using time-varying weighting gains, $\gamma_Q(k)$, $1 - \gamma_Q(k)$, and $\sigma_Q(k)$ on the RMS contour error, as compared to basic feedback control and a norm optimal controller with time-invariant gains $(\gamma_Q = 0, 1 - \gamma_Q = 1, \sigma_Q = 1)$. Figure 9 shows the improvement in the contour tracking at the corners as a result of high $\sigma_Q(k)$ weighting gains at these particular locations. Figure 10 demonstrates the improvement in the individual y-axis tracking, and therefore enhanced position and time synchronization, at the locations where the controller switches from focusing on contour tracking to individual axis tracking by changing the weighting gains from $(\gamma_Q(k) = 0, 1 - \gamma_Q(k) = 1)$ to $(\gamma_Q(k) = 1, 1 - \gamma_Q(k) = 0)$.

Figures 8-10 clearly indicate the performance improvements obtained by implementing a norm optimal learning controller using time-varying weighting matrix gains, $\gamma_Q(k)$, $1 - \gamma_Q(k)$, and $\sigma_Q(k)$, in simulation.

VI. DESIGN EXAMPLE 2: TIME-VARYING S WEIGHTING MATRIX

The previous section presented a technique for designing a time-varying weighting matrix for performance benefits. An equally important aspect in control design is ensuring robustness of the controller. This section focuses on implementing a time-varying $S$ weighting matrix in order to provide robustness in the presence of position dependent dynamics.

A. Motivation

Using analysis provided in [8], [20] it can be shown that the $S$ weighting matrix should be designed to ensure robust monotonic convergence in the presence of model uncertainty. Assuming a weighting matrix of the form $S = sI$, the weighting gain $s$ provides constant
Fig. 8. Comparison of RMS contour errors for feedback, norm optimal control using time-invariant \((\gamma_Q, 1 - \gamma_Q, \sigma_Q)\) gains, and norm optimal control using time-varying \((\gamma_Q(k), 1 - \gamma_Q(k), \sigma_Q(k))\) gains [Simulation].

Fig. 9. Trajectory tracking comparison of the norm optimal controllers with time-invariant and time-varying weighting gains. Notice that the controller using time-varying gains produces tighter tolerances around the corners as a result of increasing the gain at specific locations [Simulation].

weighting for uniform model uncertainty. However, in some applications, the dynamics are position dependent [25]. For applications which extend into locations with different dynamics, a time-varying weighting matrix of the form provided in (24) enables the controller to adequately address the model uncertainty at each location. As with the design of \(Q^{tv}\), \(S^{tv}\) has the versatility to consider coordination of the control signals, as well as each individual signal separately.

In many manufacturing applications, the system contains position dependent dynamics [26]–[28]. Often times these differences in dynamics are greatest at the edge of the system workspace...
System identification generally occurs in the center of the workspace, resulting in dynamics models which are less certain at the outer limits of the workspace. For these systems, increasing the value of the $\sigma_S(k)$ gain at the locations with more model uncertainty provides more robustness against the position-varying dynamics.

Consider a multi-axis system with potential $x$-axis position dependent dynamics illustrated in Fig. 11 [23], [26]. Resonance shifting in any axis will have similar results. For this example, the system resonances are shifted depending on the position of the axis during the trajectory. For these types of systems, time-varying designs which enable the controller to compensate for model uncertainty due to position shifting dynamics at specific time locations in the input signal are a reasonable choice. The design of a time-varying $S^{tv}$ weighting matrix is described in the following subsection.

### B. Weighting Matrix Design

Without loss of generality, the weighting matrix design in this section is focused on individual control signals. Therefore, design step 1 is not necessary because design step 2 sets $\gamma_S(k) = 1$ and $1 - \gamma_S(k) = 0$ for all $k = 1, 2, ..., N - 1$.

Assume a continuous reference trajectory in which the system performs a task at one location, moves to a different location for an additional task, and then returns to the start location. The objective is to design a time-varying weighting matrix gain $\sigma_S(k)$ that increases in value at the locations where the position dynamics have shifted and therefore the model contains some additional uncertainty. Figure 12 gives an example of such a trajectory, while Fig. 13 illustrates the heuristically determined time-varying weighting matrix gain vector $\sigma_S(k)$ for $k = 1, ..., N-1$ associated with this trajectory and uncertain system dynamics. The transitions between the high and low values for the gain have been smoothed using a lowpass Gaussian filter with a bandwidth of 5 Hz. The weighting matrix gains $\sigma_S(k)$ chosen heuristically in this example should satisfy (17) for a given uncertainty, so that the system is robustly monotonically convergent in the presence of the unmodelled dynamics.
C. Simulation Results

This subsection implements the time-varying S weighting matrix design from Fig. 13 to address position dependent dynamics. The system was subjected to a multiplicative uncertainty which mimics position dependent dynamics with a high degree of uncertainty at the position corresponding to $x, y$ from $N = 800$ to $N = 1200$. The uncertainty function can be found in Appendix B. The $Q^{tv}$ and $R$ weighting matrices were set to $(\gamma_Q(k) = 0, 1 - \gamma_Q(k) = 1, \sigma_Q(k) = 1)$ for $k = 1, 2, ..., N - 1$ and $R = rI$ with a heuristically chosen time-invariant scalar gain $r = 2e^{-2}$, respectively. Combining the time-varying gains $\sigma_S(k)$ from Fig. 13 and the $Q^{tv}$ and $R$ matrices defined above, a time-varying norm optimal learning controller was
Fig. 13. Heuristically determined time-varying gains $\sigma_S(k)$. Notice that the gain is increased in the location at which the dynamics are uncertain. The uncertainty in the dynamics leads to an increase in model uncertainty.

designed. Using this time-varying learning controller, along with the reference trajectory from Fig. 12 ($N = 1200$), the following results were obtained.

Figure 14 presents the normalized RMS contour errors for norm optimal controllers designed using high time-invariant $\sigma_S$, low time-invariant $\sigma_S$, and time-varying $\sigma_S(k)$ gain values. The time-invariant gains ($\gamma_S = 1, 1 - \gamma_S = 0, \sigma_S = constant$) in $S^{tv}$ are equivalent to using the gain $s$ in the time-invariant format $S = sI$. As the figure illustrates, low time-invariant $\sigma_S$ gain values in the presence of position dependent model uncertainty result in an unstable system, while high time-invariant $\sigma_S$ gain values produce a stable system that converges to a larger RMS contour error than the time-varying $\sigma_S(k)$ design. These results indicate how the use of a time-varying

![Diagram](image_url)
$S^{tv}$ weighting matrix results in a more robust system which converges to lower RMS contour errors in the presence of position dependent dynamics.

The next section presents a design example for a time-varying $R$ weighting matrix.

VII. DESIGN EXAMPLE 3: TIME-VARYING $R$ WEIGHTING MATRIX

The previous section presented a technique for designing a time-varying weighting matrix for robustness to position dependent dynamics. This section focuses on implementing a time-varying weighting matrix in order to maintain robustness and performance in the presence of position and time dependent external stochastic disturbances or noise.

A. Motivation

In this subsection, we consider performance in the presence of external stochastic disturbances or noise. As is shown in [9], the influence of stochastic disturbances can be minimized by reducing the convergence speed. In [8] the dominating factor in convergence speed was shown to be the $R$ weighting matrix. While a constant weighting gain $r$ in $R = rI$ provides consistent influence on the effect of stochastic disturbances, many applications include external disturbances and noise that change depending on time or position. For these cases, designing a time-varying weighting matrix of the form illustrated in (25) results in a more robust controller that is capable of handling different types of disturbances without requiring overly long convergence times. The design of a time-varying $R^{tv}$ weighting matrix is demonstrated in the following subsection.

B. Weighting Matrix Design

Without loss of generality, the design example for $R^{tv}$ presented here only considers individual signals, rather than coordination between the signals. Therefore, design step 1 can be skipped as a result of design step 2 being simplified to setting $\gamma_R(k) = 1$ and $1 - \gamma_R(k) = 0$ for $k = 1, 2, ..., N - 1$.

Consider a MIMO system in which an unknown stochastic disturbance occurs at a specific location or time during a given trajectory. An example could be a spot welding or laser cutting application where external electromagnetic interferences occur at discrete locations due to the on/off modes for these processes [30]. In applications which require mass production, the on/off modes and subsequently the locations of the time and position varying external disturbances repeat each iteration. While the occurrence of these disturbances can be predicted, the stochastic nature of the signal does not allow the system to learn the disturbances from iteration to iteration. If a time-invariant controller is designed too aggressively, the presence of the external stochastic disturbances may cause the converged error signal to fluctuate drastically, thus reducing the performance of the system. However, if a more conservative time-invariant controller is used, the system may experience long convergence times. For these types of systems with discrete external disturbances, a time-varying $R^{tv}$ weighting matrix design enables the controller to handle the time-dependent disturbances at specific times without forcing the controller to be overly conservative or too aggressive throughout the trajectory.

Figure 15 shows a raster trajectory in which the system is subject to some process on/off mode, which introduces an external interference, four times during a single time period. Figure 16 presents the time-varying weighting matrix gain $\sigma_R(k)$ associated with the given trajectory.
Fig. 15. *Raster trajectory in which the process undergoes intermittent on/off modes. The on/off transitions lead to an external stochastic disturbance that is repeated at predictable intervals.*

Fig. 16. *Time-varying weighting matrix gains $\sigma_R(k)$ for the raster trajectory in Fig. 15. Notice how the gain increases at the locations corresponding to the on/off modes which have led to the introduction of external stochastic disturbances.*

The time-varying weighting matrix gains $\sigma_R(k)$ are designed heuristically to ensure robustness and performance in the presence of external stochastic disturbances, while maintaining a reasonable convergence rate $\eta$. The transition between high and low gains is filtered using a lowpass Gaussian filter with a 15 Hz bandwidth.

**C. Simulation Results**

To validate system robustness and performance in the presence of position dependent stochastic disturbances, the time-varying $R^{tv}$ weighting matrix designed in the previous subsection is implemented on a model of the robotic testbed in Fig. 7. For this example, a Gaussian white noise
disturbance was introduced to the simulation at the specific position intervals which corresponded with the on/off locations depicted in Fig. 15. In order to determine the performance and robustness benefits of a time-varying $R^{tv}$ weighting matrix, the other two weighting matrices were set to time-invariant gains ($\gamma_Q = 0, 1 - \gamma_Q = 1, \sigma_Q = 1$) for $Q^{tv}$ and the heuristically chosen time-invariant scalar gain $s = 5e^{-1}$ for $S = sI$. Using these $Q^{tv}$ and $S$ weighting matrices, along with the time-varying gains $\sigma_R(k)$ from Fig. 16 to design a time-varying norm optimal controller and the reference trajectory from Fig. 15 ($N = 1200$), the following results were obtained.

![Normalized RMS Contou Error](image)

**Fig. 17.** Normalized contour errors for systems with time or position dependent stochastic disturbances. Variances of the converged error signals have been included in the figure. Note that the presence of a large magnitude stochastic disturbance signal results in a reduction in the overall performance [Simulation].

Figure 17 presents the normalized RMS contour errors for norm optimal controllers designed using high time-invariant $\sigma_R$, low time-invariant $\sigma_R$, and time-varying $\sigma_R(k)$ gain values. Note that due to the presence of stochastic external disturbances a more conservative $S$ weighting matrix was designed to ensure convergence. The increase in the gain $s$ resulted in a reduction in overall performance. As the figure illustrates, low time-invariant $\sigma_R$ gain values in the presence of time or position dependent external disturbances result in a system which exhibits a highly fluctuating converged error signal, while high time-invariant $\sigma_R$ gain values produce a more conservative system with a very slow convergence rate. The time-varying $\sigma_R(k)$ gain value design results in a system that converges to an error signal with smaller fluctuations than low $\sigma_R$ at a faster convergence rate than high $\sigma_R$. These results indicate how the use of a time-varying $R^{tv}$ weighting matrix results in a more robust system with a faster convergence rate and a less oscillatory converged error signal in the presence of time or position dependent stochastic disturbances.

**VIII. Conclusion**

In this paper we present the design of time-varying norm optimal ILC controllers for multi-axis systems. Explicit design steps demonstrate that time-varying weighting matrices provide a means for improving both performance and robustness of a given system.
There are two main contributions of this paper. The first half of the paper introduces the norm optimal framework and presents explicit guidelines and analysis requirements for weighting matrix design. This includes the introduction of an additional degree of flexibility via the time-varying weighting matrix format illustrated in (23)–(25). The second half of this paper demonstrates the use of these guidelines. Using the four step tuning guidelines and the four step time-varying weighting matrix design approach detailed in the paper, norm optimal learning controllers using time-invariant and time-varying weighing gains \((\gamma(k), 1 - \gamma(k), \sigma(k))\) were designed for comparison in simulation on a multi-axis robotic testbed. Simulation results showed that a norm optimal controller with time-varying gains in the \(Q_{tv}\) weighting matrix improves the trajectory tracking performance of a MIMO system over a norm optimal design using time-invariant gains \(\gamma, 1 - \gamma, \sigma\). Simulation results for the \(S_{tv}\) weighting matrix, which focuses on robustness issues, illustrated that an \(S_{tv}\) weighting matrix with time-invariant \(\gamma = 1, 1 - \gamma = 0\) and time-varying \(\sigma(k)\) gains result in either larger converged errors or an unstable system. Finally, a time-varying \(R_{tv}\) weighting matrix with time-invariant \(\gamma = 1, 1 - \gamma = 0\) and time-varying \(\sigma(k)\) gains resulted in a system that converged to an error signal with less fluctuations and a faster convergence rate in the presence of position and time-varying external stochastic disturbances, as compared to time-invariant \(\sigma\) weighting gain designs.

Future work will explore various systems in which the weighting matrix design considers coordinated control and change in control signals, as opposed to focusing on individual control signals \((u_{j+1})\). This will include determining the potential performance and robustness improvements obtained from choosing weighting gains which indicate a focus on coupled control \((u_{j+1})\) and change in control \((u_{j+1} - u_j)\) signals, \((\gamma = 0, 1 - \gamma = 0)\) and \((1 - \gamma = 1, 1 - \gamma = 1)\). Implementation of time-varying norm optimal designs which incorporate coupled control signals for specific applications will also be addressed.

IX. ACKNOWLEDGEMENTS

The authors would like to thank the Control Systems Technology group at the Technical University of Eindhoven, Eindhoven, The Netherlands. In particular, we would like to thank Maarten Steinbuch, Okko Bosgra, and Jeroen van de Wijdeven, for their insights and guidance in the area of norm optimal ILC.

APPENDIX A

EXPERIMENTAL VALIDATION OF TIME-VARYING \(Q_{tv}\)

Sections V-VII presented simulation results for different cases of the general time-varying weighting matrix designs. Here we present experimental results for the particular time-varying case where the original norm optimal design is made time-varying by changing the cost function to include time-varying weighting on the error signal. The simulation results from Section V are validated by implementing norm optimal learning controllers using time-invariant and time-varying gains \((\gamma, 1 - \gamma, \sigma)\) on the experimental testbed from Fig. 7. For this particular system, the need for time-varying \(S_{tv}\) and \(R_{tv}\) weighting matrices is not present. Analogous to the simulation results, the norm optimal controller using time-varying gains results in the most improved tracking performance as illustrated in Fig. 18 and Fig. 19.

In Fig. 18, a norm optimal learning controller using time-varying \((\gamma(k), 1 - \gamma(k), \sigma(k))\) weighting gains produces the lowest normalized RMS contour tracking errors as compared to a
norm optimal controller with time-invariant gains and a Feedback controller with a 32% reduction from the norm optimal controller using time-invariant to time-varying gains. The trajectory tracking performance improvements resulting from this reduction in RMS contour error can be seen in Fig. 19. These results indicate how time-varying $\gamma_Q(k)$, $1 - \gamma_Q(k)$, and $\sigma_Q(k)$ weighting gains result in a controller with more precise tracking for this particular trajectory.
APPENDIX B

A. Coefficients for the Plant and Controller Models

The values for the multi-axis plant and controller are provided in (35) and (36).

\[
\begin{bmatrix}
\text{Symbol} & \text{Quantity} \\
\text{Num} & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \\
\text{P}_x & 0.759 & 1.706 & 0.9596 & 0.0324 & 0.8968 \\
\text{P}_y & 0.9963 & 1.768 & 0.9567 & 0.2238 & 0.7933 \\
\text{Den} & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 \\
\text{P}_x & 0.9972 & 1.676 & 0.9479 & 0.3736 & 0.4904 \\
\text{P}_y & 0.9972 & 1.764 & 0.9562 & 0.1784 & 0.7898 \\
\text{Gain} & K \\
\text{P}_x & 0.0172 \\
\text{P}_y & 0.0459
\end{bmatrix}
\]  

(35)

\[
\begin{bmatrix}
\text{Symbol} & \text{Quantity} \\
\text{Num} & \alpha_1 & \alpha_2 & \alpha_3 \\
k_{px} & 1.92 & 0.8881 & 0.8583 \\
k_{py} & 1.377 & 0.9147 & 0.776 \\
\text{Den} & \beta_1 & \beta_2 & \beta_3 \\
k_{px} & 1.001 & 0.5182 & 0.1691 \\
k_{py} & 1.001 & 0.5182 & 0.1691 \\
\text{Gain} & K \\
k_{px} & 3.5 \\
k_{py} & 1.5
\end{bmatrix}
\]  

(36)

B. Multiplicative Uncertainty for the time-varying $S^{tv}$ Simulations Results

The multiplicative uncertainty for the simulation results presented in Section VI is given below. Each axis was subjected to multiplicative uncertainty of the following form,

\[
W = \frac{0.084}{z - 0.99}.
\]  

(37)

The uncertainty was added to the simulations at the position corresponding to $x, y$ from $N = 800$ to $N = 1200$. A block diagram representation of multiplicative uncertainty is provided below.

Fig. 20. Block diagram representation of multiplicative model uncertainty.
REFERENCES