Technical report from Automatic Control at Linköpings universitet

Comparison of performance and robustness for two classical ILC algorithms applied to a flexible system

Johanna Wallén, Mikael Norrlöf, Svante Gunnarsson

Division of Automatic Control

E-mail: johanna@isy.liu.se, mino@isy.liu.se, svante@isy.liu.se

3rd November 2008

Report no.: LiTH-ISY-R-2868

Address:
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

WWW: http://www.control.isy.liu.se

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET

Technical reports from the Automatic Control group in Linköping are available from http://www.control.isy.liu.se/publications.

Abstract

When an ILC algorithm is applied to an industrial robot, the goal is to move the tool along a desired trajectory, while only the motor position can be measured. In this paper aspects of robustness and performance are discussed when an ILC algorithm is applied to a flexible two-mass system. It is shown that the stabilising controller of the two-mass system also directly affects the robustness properties of the ILC algorithm. A classical non-causal P-ILC algorithm and a model-based ILC design using optimisation are applied to the system, based on the error for the first mass. Performance and robustness of the algorithms are compared when model errors are introduced in the system, showing that the optimisation-based approach can handle larger model uncertainties. It is illustrated that the performance of the overall system, when considering position of the second mass, is the practical limit compared to the limiting factor of the robustness of the ILC algorithms.

Keywords: Iterative learning control; Robotics; Robust control

Comparison of performance and robustness for two classical ILC algorithms applied to a flexible system

Johanna Wallén, Mikael Norrlöf and Svante Gunnarsson

***Abstract*— When an ILC algorithm is applied to an industrial robot, the goal is to move the tool along a desired trajectory, while only the motor position can be measured. In this paper aspects of robustness and performance are discussed when an ILC algorithm is applied to a flexible two-mass system. It is shown that the stabilising controller of the two-mass system also directly affects the robustness properties of the ILC algorithm. A classical non-causal P-ILC algorithm and a model-based ILC design using optimisation are applied to the system, based on the error for the first mass. Performance and robustness of the algorithms are compared when model errors are introduced in the system, showing that the optimisation-based approach can handle larger model uncertainties. It is illustrated that the performance of the overall system, when considering position of the second mass, is the practical limit compared to the limiting factor of the robustness of the ILC algorithms.**

I. INTRODUCTION

Traditionally *Iterative Learning Control* (ILC) has been applied to systems where the controlled output also is the measured variable. In industrial robots this is typically not the case, since the goal is that the tool follows a desired trajectory while the motor positions can be measured. In this paper the robot problem is studied using an idealised model, a flexible two-mass system, where it is assumed that only the position of the first mass, referred to as the motor position, is measurable. The position of the second mass, referred to as the arm position, is only used as an evaluation variable that should follow the desired arm trajectory.

The ILC method was introduced in [2], [6] and [7], and robotic applications have been an important field of application for ILC ever since, see for example [14] and [22]. The reason for this is that in many robotic applications the robot performs the same trajectory repeatedly, starting from the same initial conditions. In [17] a detailed overview over the ILC research area is given together with a categorisation of much of the publications from 1984 until 1998. Publications between 1998 and 2004 are covered in [1], while a résumé of recent publications can be found in [3].

This paper discusses robustness and performance aspects when an ILC algorithm is applied to a flexible system, such as the system shown in Fig. 1. In [15] and [12] ILC is applied to flexible systems but it is assumed that the position of the arm is measurable, while here it is assumed that only the position of the motor is available. It is assumed that the system is stabilised by a feedback controller and it is shown how this controller will directly affect the robustness properties of the ILC algorithm. The next contribution is to compare a classical non-causal P-ILC algorithm to a model-based ILC algorithm designed by optimisation [10], when the ILC update equation only uses the motor position. The performance of the two design methods is compared and the robustness is evaluated when model errors are introduced in the system. The model errors; parameter variations in arm inertia and spring constant, are motivated by real model uncertainties found in industrial robot applications. Finally it is discussed how the robustness of the ILC algorithm can be related to the performance of the overall system when the arm position in the two-mass system is considered.

This work was supported by the Swedish Research Council (VR).

J. Wallén, S. Gunnarsson, and M. Norrlöf are with Division of Automatic Control, Department of Electrical Engineering, Linköping University, SE-58183 Linköping, Sweden, {johanna, mino, svante}@isy.liu.se

II. PROBLEM DESCRIPTION

A modern industrial robot cannot be described by traditional rigid body models that are used in classical robot control. The trend is towards more flexible robots and it is also a fact that the flexibilities cannot be assumed to be only in the joints, see [20] and [16]. A good model of an industrial robot therefore includes joint as well as arm flexibilities and requires up to 50 spring-mass elements [4].

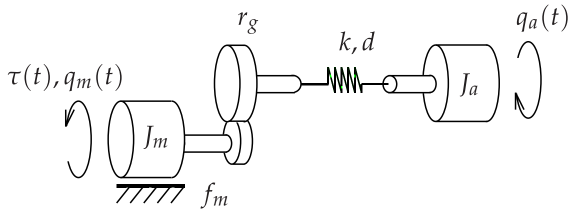

Fig. 1. A flexible two-mass model of the dynamics in a single robot joint, characterised by spring k, damper d, viscous friction f_m, gear ratio r_g, moments of inertia J_m, J_a, torque $\tau(t)$, motor angle $q_m(t)$ and arm angle $q_a(t)$.

The two-mass model in Fig. 1 is simple compared to an accurate model of an industrial robot, but it still captures some of the principle behaviours that can be confirmed from experiments performed with a robot, see for example [21], [25] and [24]. The system can be described by

$$\begin{aligned} J_m\ddot{q}_m(t) &= -f_m\dot{q}_m(t) - r_g k\big(r_g q_m(t) - q_a(t)\big) \\ &\quad - r_g d\big(r_g\dot{q}_m(t) - \dot{q}_a(t)\big) + k_\tau u(t), \\ J_a\ddot{q}_a(t) &= k\big(r_g q_m(t) - q_a(t)\big) + d\big(r_g\dot{q}_m(t) - \dot{q}_a(t)\big), \end{aligned} \tag{1}$$

where the parameter values used are presented in Table I. Introducing $x(t) = \begin{pmatrix} q_a(t) & \dot{q}_a(t) & q_m(t) & \dot{q}_m(t) \end{pmatrix}^T$ gives a linear state-space model of the system. The transfer functions G_m and G_a, describing the behaviour on the motor side and

TABLE I

PARAMETER VALUES USED IN THE FLEXIBLE TWO-MASS MODEL.

T_s	r_g	J_m	J_a	k	d	f_m	k_τ
0.01	0.2	0.0021	0.0991	5	0.0924	0.0713	0.122

from motor to arm side, respectively, are derived directly from the well-known relation $G(s) = C(sI - A)^{-1}B$.

The ILC algorithm is applied to the system and the structure is illustrated in Fig. 2. Since the desired servo performance is planned to be achieved by the ILC algorithm, a simple control structure can be used, for example a controller of PD type, PID type, or an LQG controller. Here, a PD controller is used including a lag part using a low-pass filter, giving the following transfer function

$$F(s) = K_p + \frac{K_d s}{1 + T_f s} = \frac{K_p + (K_d + K_p T_f)s}{1 + T_f s}. \tag{2}$$

The controller parameters are chosen to the following values

$$K_p = 6, \qquad K_d = 0.274, \qquad T_f = 0.075, \tag{3}$$

and the desired tracking of the reference signal on the motor side and arm side is achieved, as is further described in [24].

The motor-angle reference $r_m(t)$ is derived from the arm-angle reference $r_a(t)$ by using a pre-filter as in

$$r_m(t) = F_r(q) r_a(t). \tag{4}$$

To get an ideal correspondence between the arm and motor reference in the nominal case, the filter F_r is chosen as

$$F_r(q) = \frac{1}{q} \frac{1}{G_{a,d}(q)}, \tag{5}$$

where $G_{a,d}(q)$ is a sampled version of $G_a(s)$ for the nominal model with no model errors introduced in the model. The factor $1/q$ is introduced to make F_r proper. In the robustness analysis model errors are introduced in the system. The pre-filter is however always based upon the nominal model $G_{a,d}(q)$ and the effect on the performance in terms of tracking the reference trajectory on the arm side is further discussed in Section V.

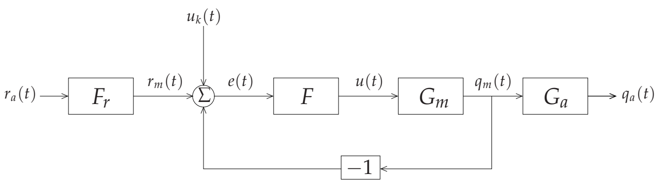

Fig. 2. The controlled system illustrated by G_m and G_a; transfer function from motor torque to motor angle $q_m(t)$ and motor angle to arm angle $q_a(t)$, respectively. F represents the feedback controller. The ILC input signal $u_k(t)$ is added to the reference $r_m(t)$. The reference $r_a(t)$ is filtered by F_r.

III. ILC ALGORITHMS

Two algorithms are presented in this section. The first is a standard non-causal P-type ILC algorithm, the design is here referred to as heuristic [18]. The second is an optimisation-based design and the algorithm uses explicitly a model of the system.

A. General system description

The general system description when an ILC algorithm is applied to a linear discrete-time SISO system can be formulated as

$$y_k(t) = T_r(q) r(t) + T_u(q) u_k(t), \tag{6}$$

where the ILC input signal and the output from the system are $u_k(t)$ and $y_k(t)$, respectively, k denotes iteration number and $r(t)$ is the reference input. The signals are defined on a finite time interval $t = 0, \ldots, N$ with N number of samples. Finally, $T_r(q)$ and $T_u(q)$ are stable discrete-time filters. System and measurement disturbances are not included here, but can easily be treated in this framework.

The update equation for a general first-order ILC algorithm with iteration-independent operators is given by

$$u_{k+1}(t) = Q(q)\big(u_k(t) + L(q) e_k(t)\big), \tag{7}$$

where q is the time-shift operator and the linear filters $Q(q)$ and $L(q)$ are possibly non-causal. The choice of filters are discussed below for two different design approaches. The error

$$e_k(t) = r_m(t) - y_k(t), \tag{8}$$

is the difference between motor-angle reference and measured motor angle at iteration k. The update equation (7) implies the standard frequency-domain convergence criterion, see for example [19],

$$|1 - L(e^{i\omega}) T_u(e^{i\omega})| < |Q^{-1}(e^{i\omega})|, \quad \forall \omega, \tag{9}$$

where T_u denotes the transfer function from the applied ILC input $u_k(t)$ to the measured output $y_k(t)$. The criterion shows that the filter Q can be used to improve the robustness of the ILC algorithm.

The system can be described in matrix form, also called lifted system description. Let

$$\boldsymbol{y}_k = \big(y_k(0) \quad \ldots \quad y_k(N-1)\big)^T, \tag{10}$$

and define $\boldsymbol{r}$ and $\boldsymbol{u}_k$ similarly. In the matrix formulation the system (6) can be formulated as

$$\boldsymbol{y}_k = \boldsymbol{T}_r \boldsymbol{r} + \boldsymbol{T}_u \boldsymbol{u}_k. \tag{11}$$

This system description is more general than the LTI-system representation presented above because $\boldsymbol{T}_r$ and $\boldsymbol{T}_u$ can be linear time-variant (LTV). When $\boldsymbol{T}_u$ represents a causal linear time-invariant system, the matrix $\boldsymbol{T}_u$ is formed by the impulse response coefficients of the transfer function $T_u(q)$ and is described by the Toeplitz matrix

$$\boldsymbol{T}_u = \begin{pmatrix} g_{T_r}(0) & 0 & \dots & 0 \\ g_{T_r}(1) & g_{T_r}(0) & & \vdots \\ \vdots & & \ddots & 0 \\ g_{T_r}(N-1) & g_{T_r}(N-2) & \dots & g_{T_r}(0) \end{pmatrix}. \tag{12}$$

$\boldsymbol{T}_r$ is defined analogously, see [10].

Using the matrix description, the update equation for the ILC algorithm (7) can be written

$$\boldsymbol{u}_{k+1} = \boldsymbol{Q}(\boldsymbol{u}_k + \boldsymbol{L}\boldsymbol{e}_k). \tag{13}$$

From an implementation point of view the drawback of the matrix formulation is that the size of the matrices involved increase with the number of samples N in the reference trajectory. In practice it is however often possible to use an equivalent filter formulation, as will be shown in the next sections.

B. Heuristic ILC design

The heuristic ILC design, described in Algorithm 1, uses a model of the system to ensure stability and monotone convergence from the criterion (9). The knowledge can also be reduced to only a known time delay and size of the first Markov parameter of the system, but then the criterion (9) cannot be guaranteed to be satisfied. The filter Q is applied to get a robust algorithm.

Algorithm 1 Heuristic design

1. Choose the Q filter as a low-pass filter with cutoff frequency such that the bandwidth of the learning algorithm is sufficient.
2. Let $L = \gamma q^{\delta}$. Choose γ and δ such that the stability criterion formulated in the frequency domain, $|1 - L(e^{i\omega})T_u(e^{i\omega})| < |Q^{-1}(e^{i\omega})|$, is satisfied.

Normally a learning gain γ close to 1 is preferred for fast convergence, but γ can be chosen in the range $0 < \gamma \leq 1$. Notice however that it is the static gain of T_u that decides the range for γ; here a static gain of 1 is assumed.

A high cutoff frequency of the robustifying low-pass filter Q means a filter near the ideal filter $Q = 1$, and thereby also a that the error is closer to zero after convergence, as can be seen in, for instance, [9].

C. Optimisation-based ILC design

The optimisation-based ILC design is based on [10], and similar work are presented in [8] and [13]. The matrix formulation (11) is used and the learning filters or matrices are derived by minimising the quadratic criterion

$$J_{k+1} = \boldsymbol{e}_{k+1}^T \boldsymbol{W}_e \boldsymbol{e}_{k+1} + \boldsymbol{u}_{k+1}^T \boldsymbol{W}_u \boldsymbol{u}_{k+1}, \tag{14}$$

subject to the constraint

$$(\boldsymbol{u}_{k+1} - \boldsymbol{u}_k)^T (\boldsymbol{u}_{k+1} - \boldsymbol{u}_k) \leq \delta. \tag{15}$$

The weighting matrices $\boldsymbol{W}_e$ and $\boldsymbol{W}_u$ determine the trade-off between performance and input energy. The optimisation is solved by introducing the Lagrange multiplier λ, which yields

$$\begin{aligned} \bar{J}_{k+1} = &\boldsymbol{e}_{k+1}^T \boldsymbol{W}_e \boldsymbol{e}_{k+1} + \boldsymbol{u}_{k+1}^T \boldsymbol{W}_u \boldsymbol{u}_{k+1} \\ &+ \lambda \big((\boldsymbol{u}_{k+1} - \boldsymbol{u}_k)^T (\boldsymbol{u}_{k+1} - \boldsymbol{u}_k) - \delta \big). \end{aligned} \tag{16}$$

Using the relation (11) implies that

$$\boldsymbol{e}_{k+1} = (I - \boldsymbol{T}_r)\boldsymbol{r} - \boldsymbol{T}_u \boldsymbol{u}_{k+1}. \tag{17}$$

Differentiating (16) and using (17) gives an optimum where the derivative is equal to zero. Rewriting the expression gives

$$\begin{aligned} \boldsymbol{u}_{k+1} = &(\boldsymbol{W}_u + \lambda \boldsymbol{I} + \boldsymbol{T}_u^T \boldsymbol{W}_e \boldsymbol{T}_u)^{-1} \\ &\big(\lambda \boldsymbol{u}_k + \boldsymbol{T}_u^T \boldsymbol{W}_e (\boldsymbol{I} - \boldsymbol{T}_r)\boldsymbol{r}\big). \end{aligned} \tag{18}$$

The relation $(\boldsymbol{I} - \boldsymbol{T}_r)\boldsymbol{r} = \boldsymbol{e}_k + \boldsymbol{T}_u \boldsymbol{u}_k$ then gives

$$\begin{aligned} \boldsymbol{u}_{k+1} = &(\boldsymbol{W}_u + \lambda \boldsymbol{I} + \boldsymbol{T}_u^T \boldsymbol{W}_e \boldsymbol{T}_u)^{-1} \\ &\big((\lambda \boldsymbol{I} + \boldsymbol{T}_u^T \boldsymbol{W}_e \boldsymbol{T}_u)\boldsymbol{u}_k + \boldsymbol{T}_u^T \boldsymbol{W}_e \boldsymbol{e}_k\big). \end{aligned} \tag{19}$$

Comparison to the structure given in (13), and noting that a model $\hat{\boldsymbol{T}}_u$ of the system has to be used in the final algorithm, it gives

$$\begin{aligned} \boldsymbol{Q} &= (\boldsymbol{W}_u + \lambda \boldsymbol{I} + \hat{\boldsymbol{T}}_u^T \boldsymbol{W}_e \hat{\boldsymbol{T}}_u)^{-1} (\lambda \boldsymbol{I} + \hat{\boldsymbol{T}}_u^T \boldsymbol{W}_e \hat{\boldsymbol{T}}_u), \\ \boldsymbol{L} &= (\lambda \boldsymbol{I} + \hat{\boldsymbol{T}}_u^T \boldsymbol{W}_e \hat{\boldsymbol{T}}_u)^{-1} \hat{\boldsymbol{T}}_u^T \boldsymbol{W}_e. \end{aligned} \tag{20}$$

The design procedure is summarised in Algorithm 2. In [10] a more thorough discussion of the optimisation-based design is given and more results are presented. The two algorithms presented here are also compared in the experimental study [18].

Algorithm 2 Model-based time-domain design using optimisation

1. Build a model of the relation between the ILC input and the resulting correction on the output, that is, find a model $\hat{\boldsymbol{T}}_u$ of $\boldsymbol{T}_u$.
2. Choose the weighting matrices $\boldsymbol{W}_e$ and $\boldsymbol{W}_u$ and calculate the matrices $\boldsymbol{Q}$ and $\boldsymbol{L}$ in (20).
3. Evaluate the design in an experiment or a simulation and change the weights if the result is not according to the specified performance and/or robustness.

Here the weights are chosen as $\boldsymbol{W}_e = \boldsymbol{I}$ and $\boldsymbol{W}_u = \rho \boldsymbol{I}$, as in [10]. This gives the matrices $\boldsymbol{Q}$ and $\boldsymbol{L}$ as

$$\begin{aligned} \boldsymbol{Q} &= (\rho \boldsymbol{I} + \lambda \boldsymbol{I} + \hat{\boldsymbol{T}}_u^T \hat{\boldsymbol{T}}_u)^{-1} (\lambda \boldsymbol{I} + \hat{\boldsymbol{T}}_u^T \hat{\boldsymbol{T}}_u), \\ \boldsymbol{L} &= (\lambda \boldsymbol{I} + \hat{\boldsymbol{T}}_u^T \hat{\boldsymbol{T}}_u)^{-1} \hat{\boldsymbol{T}}_u^T, \end{aligned} \tag{21}$$

which in the frequency domain corresponds to the filtering operation

$$\begin{aligned} U_{k+1}(q) &= Q(q)\big(U_k(q) + L(q)E_k(q)\big), \\ Q(q) &= \frac{\lambda + \hat{T}_u(q)\hat{T}_u(q^{-1})}{\rho + \lambda + \hat{T}_u(q)\hat{T}_u(q^{-1})}, \\ L(q) &= \frac{\hat{T}_u(q^{-1})}{\lambda + \hat{T}_u(q)\hat{T}_u(q^{-1})}, \end{aligned} \tag{22}$$

when $\hat{\boldsymbol{T}}_u$ is an LTI-system. From this expression [18], using the convergence criterion (9), it can be seen that the algorithm satisfies the convergence criterion for all choices of $\rho > 0$. Larger values of λ will give a slower convergence rate, while larger ρ gives a larger asymptotic error, as is shown in for example [10].

IV. Robustness Analysis

Assume that the two ILC design methods presented in Section III form the filters Q and L from the nominal system model $\hat{T}_u$. The ILC algorithms are actually applied to a system with model error,

$$T_u(q) = \big(1+\Delta(q)\big)\hat{T}_u(q), \tag{23}$$

where Δ is assumed to be a norm-bounded non-parametric uncertainty. The nominal case corresponds to no model error ($\Delta = 0$). The robustness can be derived as the maximum $|\Delta| < R(\omega)$ over the frequencies ω for the chosen L and Q while stability is still achieved. Robust performance is also of practical interest – how large model variations we can have and still get the desired performance. Finally, a performance evaluation of the error on the arm side is considered.

A. Unstructured model error

The model error model from (23) corresponds to the case where ILC is applied to a system T_u where no structural knowledge of the components of the error are known. A model of T_u can, for example, be found using system identification. In [26] different types of model errors are discussed and [5], [23] show examples applied to ILC.

Introducing the model (23), the frequency-domain convergence criterion (9) results in

$$|1-L(e^{i\omega})\big(1+\Delta(e^{i\omega})\big)\hat{T}_u(e^{i\omega})| < |Q^{-1}(e^{i\omega})|, \quad \forall\omega. \tag{24}$$

In this paper $\hat{T}_u$ is assumed to be a SISO system and therefore the transfer operators commute. Introducing the notation

$$\Delta(e^{i\omega}) = R(\omega)e^{i\varphi(\omega)} \tag{25}$$

gives the frequency-domain convergence criterion (24) as

$$|1-\big(1+R(\omega)e^{i\varphi(\omega)}\big)L(e^{i\omega})\hat{T}_u(e^{i\omega})| < |Q^{-1}(e^{i\omega})|, \quad \forall\omega. \tag{26}$$

The robustness margin can be derived from the optimisation problem for every frequency ω, formulated as

$$\begin{aligned}\Delta_{\text{rb}}(\omega) &= \min_{R,\varphi} R(\omega)\\ \text{s.t.}\quad &|1-\big(1+R(\omega)e^{i\varphi(\omega)}\big)L(e^{i\omega})\hat{T}_u(e^{i\omega})| = |Q^{-1}(e^{i\omega})|.\end{aligned} \tag{27}$$

This can be interpreted as the smallest R for which the convergence criterion (24) is fulfilled for all φ. From (27) it is clear the robustness margin Δ_{rb} is conservative.

B. Structured model error

In this section the model error is studied for the case when the ILC update is using the difference between the motor-angle reference and the measured motor angle. The ILC update is also assumed to be added to the motor-angle reference, as in Fig. 2.

When structural knowledge of T_u is available, such as the controller F and the system G_m in Fig. 2, the model error can be expressed explicitly in terms of model errors for the nominal system $\hat{G}_m$. The controlled system is then $G_m = (1+\Delta_G)\hat{G}_m$. The relation between Δ and Δ_G is given from

$$\begin{aligned}\big(1+\Delta(q)\big)\hat{T}_u(q) &= \frac{G_m(q)F(q)}{1+G_m(q)F(q)}\\ &= \frac{\big(1+\Delta_G(q)\big)\hat{G}_m(q)F(q)}{1+\big(1+\Delta_G(q)\big)\hat{G}_m(q)F(q)},\end{aligned} \tag{28}$$

which implies that

$$\Delta(q) = \frac{\Delta_G(q)}{1+\big(1+\Delta_G(q)\big)\hat{G}_m(q)F(q)}. \tag{29}$$

A robustness margin expressed in terms of uncertainty in G_m can be derived by solving the following optimisation problem for each frequency ω, using $\Delta_G(e^{i\omega}) = R(\omega)e^{i\varphi(\omega)}$,

$$\begin{aligned}\Delta_{G,\text{rb}}(\omega) &= \min_{R,\varphi} R(\omega)\\ \text{s.t.}\quad &\left|\frac{R(\omega)e^{i\varphi(\omega)}}{1+\big(1+R(\omega)e^{i\varphi(\omega)}\big)\hat{G}_m(e^{i\omega})F(e^{i\omega})}\right| = \Delta_{\text{rb}}.\end{aligned} \tag{30}$$

The robustness margin is also in this case conservative, since it results in the amplitude of the model error $\Delta_{G,\text{rb}}$ of the system G_m that satisfies both the equality (29) and the convergence criterion (24) for all φ. Without actual knowledge of the phase φ of the uncertainty, this is the best margin that can be achieved.

From (29) it can be seen that, in terms of the model error Δ, the controller F plays an important role. As small gain as possible for the relation is desired, to achieve a small Δ. With the sensitivity function for the true system G_m,

$$S_0(q) = \frac{1}{1+G_m(q)F(q)}, \tag{31}$$

the relation (29) can be rewritten to

$$\Delta(q) = S_0(q)\Delta_G(q). \tag{32}$$

The design of the controller F and the ILC algorithm thereby interact and tuning of the controller also influences the magnitude of the model errors. On example where the design of the feedback loop and the ILC algorithm is made in one step using a robust optimal design procedure, is [23].

C. Asymptotic error on the arm side

In this paper it is assumed that the controlled variable, represented by the position of the second mass in Fig. 1, cannot be measured. The performance evaluation of the resulting ILC algorithms has to include the asymptotic error on the arm side. Starting from the system (6), assuming that it is stable and that it is controlled by the ILC updating equation (7), the asymptotic ILC control signal converges to

$$\lim_{k\to\infty} u_k(t) = \frac{Q(q)L(q)(1-T_r(q))r(t)}{1-Q(q)\big(1-T_u(q)L(q)\big)}, \tag{33}$$

and the asymptotic output on the motor side can be computed from (6). The ILC algorithm is also assumed to satisfy the convergence criterion (9). The asymptotic output on the arm side is then achieved using $q_a(t) = G_a(q)q_m(t)$.

To evaluate the robustness of the complete system, model errors can now be introduced in the two-mass system. Stability is achieved by the same criterion as the feedback loop and the criterion for the ILC algorithm. Important is instead to analyse robust performance to, for instance, decide a maximum position error in the tracking of the mass on the arm side, and check how large variations it is possible to have in the model parameters and still meet the requirements in the tracking accuracy. The result should be compared to the robustness of the ILC algorithm.

V. SIMULATION RESULTS

A. Nominal performance

The two ILC design schemes presented in Section III are now used to design four ILC algorithms – a slow and fast nominal tuning approach for each ILC algorithm design. The idea is to compare the robustness for the slow and fast algorithms and also compare the two design methodologies, where the optimisation-based algorithm relies heavily on the model while the heuristic algorithm uses very few model knowledge. The Q filter in the heuristic design uses a low-pass second-order Butterworth filter, applied as a fourth-order zero-phase filter. The parameters used in the tuning of the heuristic design are

1) Slow convergence: $\gamma = 0.25$, $\delta = 5$ and $\omega_n = 10\,\text{Hz}$.
2) Fast convergence: $\gamma = 0.9$, $\delta = 5$ and $\omega_n = 17\,\text{Hz}$.

The parameters used for the optimisation-based ILC algorithm tuning are

1) Slow convergence: $\lambda = 1.4$ and $\rho = 0.0199$.
2) Fast convergence: $\lambda = 0.17$ and $\rho = 0.00376$.

The parameters of the heuristic ILC algorithm and the optimisation-based ILC algorithm are chosen so that the convergence rate and final error, measured in ∞-norm on the motor side, are as similar as possible for the two algorithms, as can be seen in Fig. 3. The system is driven by a reference which is a filtered step function, continuous until the third derivative with respect to time.

The convergence criterion (9) is illustrated in Fig. 4. For a fast convergence, $|1 - LT_u|$ should be small and the optimisation-based approach satisfies this requirement better than the heuristic design, and hence the optimisation-based design is better from a performance perspective. From a robustness perspective, the heuristic design has an extra degree of freedom since the robustifying Q filter can be chosen independently of the L filter. In practice it is easy to extend the optimisation-based approach to include a separate Q filter design and hence achieve the same robustness properties as the heuristic design, but this is not included here.

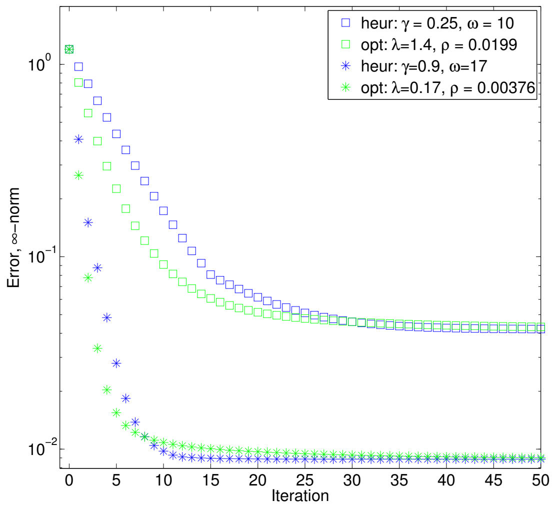

Fig. 3. The error $||e_k||_\infty$ on the motor side for the slow and fast tuning when using a heuristic and optimisation-based ILC design.

B. Robustness with respect to parameter variations

The parameter variations in moment of inertia on the arm side, J_a, and spring constant, k, corresponds to the cases where the mass on the arm side (including the load) is incorrect and the stiffness of the gearbox is uncertain. In Fig. 5 model variations $|\Delta(e^{i\omega})|$ are shown when J_a and k are changed $\pm 50\%$ from their nominal values, which means that the actual model error $\Delta(e^{i\omega})$ is illustrated. As a comparison, the robustness margin $\Delta_{\text{rb}}(e^{i\omega})$ from (27) is also illustrated in Fig. 5 for the ILC algorithm tuning cases described above, which shows how large model errors in the worst direction one can have for each frequency (a conservative measure) and still satisfy the convergence criterion (24).

It can be concluded that the optimisation-based design is much more robust within the bandwidth of the ILC algorithm, since a larger robustness margin $\Delta_{\text{rb}}(e^{i\omega})$ is achieved for each frequency in the case of the optimisation-based ILC algorithm design. The robustness margin Δ_{rb} is a conservative bound and the parameter variations can be higher if only the convergence criterion (24) is considered. As an example, the stiffness k can be increased by nearly a factor 2 without violating the convergence criterion (24) while the robustness gain Δ_{rb} gives an upper bound of approximately 15%.

C. Performance on the arm side

The performance evaluation of the error on the arm side is computed from the reference signal on the arm side minus the asymptotic output on the arm side. The error has to satisfy

$$|e_a(t)| = |r_a(t) - q_a(t)| \leq 0.02. \tag{34}$$

In Table II the maximum model errors for the parameters J_a and k that satisfy the error bound in (34) are given. These values should be compared to what is achieved in the

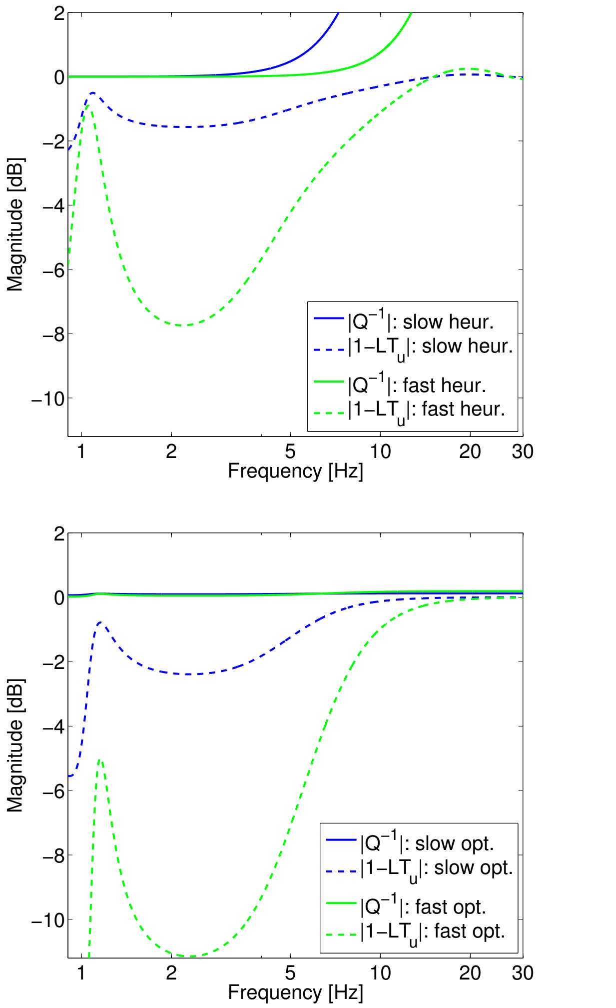

Fig. 4. Convergence criterion of the two design approaches – heuristic (upper) and optimisation-based (lower) design – shown for the two cases of slow and fast tuning.

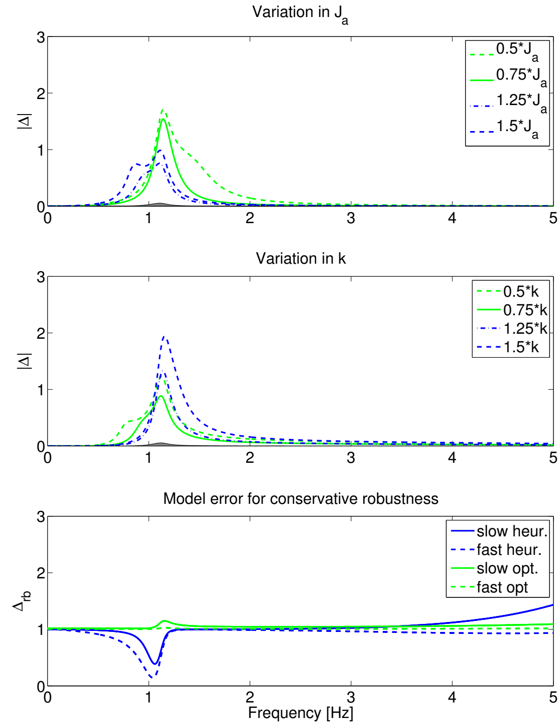

Fig. 5. Model errors, $|\Delta|$, computed from (29) when J_a (upper) and k (middle) are changed $\pm 50\%$ compared to the robust margin, Δ_{rb}, from (27) (lower).

TABLE II

MAXIMUM MODEL ERRORS FOR THE PARAMETERS J_a AND k, GIVING AN ARM-SIDE BEHAVIOUR SATISFYING $|e_a(t)| = |r_a(t) - q_a(t)| \leq 0.02$.

Algorithm	Deviation in J_a	Deviation in k
Slow heuristic	$+3.5\%$ -1.15%	$+1.3\%$ -3.42%
Fast heuristic	$+3.5\%$ -0.97%	$+1.08\%$ -3.42%
Slow optimisation-based	$+3.17\%$ -5.09%	$+3.72\%$ -3.09%
Fast optimisation-based	$+3.46\%$ -1.41%	$+1.56\%$ -3.35%

robustness analysis in Fig. 5. The robust performance of the complete system is clearly limited by the specification of the behaviour on the arm side. In practice it is therefore not the robustness of the ILC algorithm that limits the actual model errors that can be introduced. Instead it is the fact that only motor position is available and the design completely relies on a feed-forward filter based on the inverse of the nominal system model. A first step to improve the result could be to use the model of the system and to estimate the arm position from motor torque and position measurements. The estimate could then be used by the ILC algorithm as proposed in [11]. To include arm side measurements in the ILC algorithm is of course a natural next step, but in the industrial application this is not straightforward, mainly from a safety and a cost perspective.

VI. CONCLUSIONS

Iterative learning control is applied to a flexible system, modelled as a two-mass system, and a number of observations can be made from the results. First, the robustness of the ILC design is highly dependent on the feedback design if ILC is applied to a system in closed loop, and the model errors can be significantly higher compared to the case when ILC is directly applied to a system. Two ILC designs have been compared in a simulation study and one important result is that, independently of the choice of ILC design method, the resulting ILC algorithm is surprisingly robust when the system is subject to parameter variations. It is clear that the bandwidth of the ILC algorithm has an impact of the robustness, a slow algorithm is more robust than a fast algorithm, and the optimisation-based algorithm is better than the heuristic approach, but the overall result is that the ILC algorithms are very robust. A conservative robustness margin is derived and the robustness with respect to parameter variations is evaluated using this margin. The result is also compared to what is achieved by directly using the standard convergence criterion, which shows that even larger variations of the parameters can be introduced. The final conclusion is that the complete system is much more sensitive to model errors because of the fact that the controlled variable, the position of the second mass in the two-mass system, cannot be measured. The reference trajectory to the ILC algorithm is generated by a feed-forward filter from the reference trajectory of the second mass, based upon the nominal system model. From a practical point of view it is not the robustness of the ILC algorithms that limits the performance of the complete system, but that the controlled variable cannot be measured. Including direct or indirect measurements from the second mass is therefore a natural next step.

REFERENCES

[1] H.-S. Ahn, Y. Chen, and K. L. Moore. Iterative learning control: Brief survey and categorization. *IEEE Trans. Systems, Man, Cybernetics – Part C: Appl. Reviews*, 37(6):1099–1121, November 2007.

[2] S. Arimoto, S. Kawamura, and F. Miyazaki. Bettering operation of robots by learning. *Journal Robot. Syst.*, 1(2):123–140, 1984.

[3] D. A. Bristow, M. Tharayil, and A. G. Alleyne. A survey of iterative learning control. *IEEE Control Syst. Mag.*, pages 96–114, 2006.

[4] T. Brogårdh. Present and future robot control development – An industrial perspective. *Annual Reviews in Control*, 31(1):69–79, 2007.

[5] M. Butcher, A. Karimi, and R. Longchamp. A statistical analysis of certain iterative learning control algorithms. *Int. Journal Control*, 81(1):156–166, January 2008.

[6] G. Casalino and G. Bartolini. A learning procedure for the control of movements of robotic manipulators. In *IASTED Sym. Robot. Autom.*, pages 108–111, San Francisco, USA, May 1984.

[7] J. J. Craig. Adaptive control of manipulators through repeated trials. In *Proc. American Control Conf.*, pages 1566–1572, San Diego, CA, June 1984.

[8] D. de Roover. Synthesis of a robust iterative learning controller using an H_∞ approach. In *Proc. IEEE Conf. Decision Control*, pages 3044–3049, Kobe, Japan, December 1996.

[9] H. Elci, R. W. Longman, M. Q. Phan, J.-N. Juang, and R. Ugoletti. Simple learning control made practical by zero-phase filtering: Applications to robotics. *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, 49(6):753–767, June 2002.

[10] S. Gunnarsson and M. Norrlöf. On the design of ILC algorithms using optimization. *Automatica*, 37(12):2011–2016, December 2001.

[11] S. Gunnarsson, M. Norrlöf, E. Rahic, and M. Özbek. On the use of accelerometers in iterative learning control of a flexible robot arm. *Int. Journal Control*, 80(3):363–373, March 2007.

[12] K. Kinoshita, T. Sogo, and N. Adachi. Adjoint-type iterative learning control for a single-link flexible arm. In *Proc. IFAC World Congress*, Barcelona, Spain, July 2002.

[13] K. S. Lee and J. H. Lee. Design of quadratic criterion-based iterative learning control. In Z. Bien and J.-X. Xu, editors, *Iterative Learning Control: Analysis, Design, Integration and Applications*, pages 165–192. Kluwer Academic Publishers, 1998.

[14] R. W. Longman. Iterative learning control and repetitive control for engineering practice. *Int. Journal Control*, 73(10):930–954, July 2000.

[15] P. Lucibello, S. Panzieri, and G. Ulivi. Repositioning control of a two-link flexible arm by learning. *Automatica*, 33(4):579–590, April 1997.

[16] S. Moberg and S. Hanssen. A DAE approach to feedforward control of flexible manipulators. In *Proc. IEEE Int. Conf. Robotics Automation*, pages 3439–3444, Roma, Italy, April 2007.

[17] K. L. Moore. Iterative learning control - an expository overview. *App. Computational Controls, Signal Proc. Circuits*, 1:151–214, 1999.

[18] M. Norrlöf and S. Gunnarsson. Experimental comparison of some classical iterative learning control algorithms. *IEEE Trans. Robot. Autom.*, 18:636–641, 2002.

[19] M. Norrlöf and S. Gunnarsson. Time and frequency domain convergence properties in iterative learning control. *Int. Journal Control*, 75:1114–1126, 2002.

[20] J. Öhr, S. Moberg, E. Wernholt, S. Hanssen, J. Pettersson, S. Persson, and S. Sander-Tavallaey. Identification of flexibility parameters of 6-axis industrial manipulator models. In *Proc. ISMA2006 Int. Conf. Noise and Vibration Eng.*, pages 3305–3314, Leuven, Belgium, September 2006.

[21] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, Ltd, New York, 2006.

[22] A. Tayebi. Adaptive iterative learning control for robot manipulators. *Automatica*, 40(7):1195–1203, July 2004.

[23] A. Tayebi and M. B. Zaremba. Robust iterative learning control design is straightforward for uncertain LTI systems satisfying the robust performance condition. *IEEE Trans. Autom. Control*, 48(1):101–106, January 2003.

[24] J. Wallén. *On Kinematic Modelling and Iterative Learning Control of Industrial Robots*. Licentiate's thesis no 1343, Dept. Electr. Eng., Linköpings universitet, Sweden, January 2008. Available: `http://www.control.isy.liu.se/research/reports/LicentiateThesis/Lic1343.pdf`.

[25] J. Wallén, M. Norrlöf, and S. Gunnarsson. Arm-side evaluation of ILC applied to a six-degrees-of-freedom industrial robot. In *Proc. IFAC World Congress*, Seoul, Korea, July 2008. Accepted for publication.

[26] K. Zhou, J. C. Doyle, and K. Glover. *Robust and optimal control*. Prentice-Hall, Inc, Upper Saddle River, New Jersey, 1996.

Avdelning, Institution
Division, Department

Division of Automatic Control
Department of Electrical Engineering

Datum
Date

2008-11-03

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ ________________

Rapporttyp
Report category

☐ Licentiatavhandling
☐ Examensarbete
☐ C-uppsats
☐ D-uppsats
☒ Övrig rapport
☐ ________________

ISBN
—

ISRN
—

Serietitel och serienummer
Title of series, numbering

ISSN
1400-3902

LiTH-ISY-R-2868

URL för elektronisk version

http://www.control.isy.liu.se

Titel
Title

Comparison of performance and robustness for two classical ILC algorithms applied to a flexible system

Författare
Author

Johanna Wallén, Mikael Norrlöf, Svante Gunnarsson

Sammanfattning
Abstract

When an ILC algorithm is applied to an industrial robot, the goal is to move the tool along a desired trajectory, while only the motor position can be measured. In this paper aspects of robustness and performance are discussed when an ILC algorithm is applied to a flexible two-mass system. It is shown that the stabilising controller of the two-mass system also directly affects the robustness properties of the ILC algorithm. A classical non-causal P-ILC algorithm and a model-based ILC design using optimisation are applied to the system, based on the error for the first mass. Performance and robustness of the algorithms are compared when model errors are introduced in the system, showing that the optimisation-based approach can handle larger model uncertainties. It is illustrated that the performance of the overall system, when considering position of the second mass, is the practical limit compared to the limiting factor of the robustness of the ILC algorithms.

Nyckelord
Keywords

Iterative learning control; Robotics; Robust control