ITERATIVE LEARNING CONTROL WITH SENSOR FUSION FOR ROBOTS WITH MISMATCHED DYNAMICS AND MISMATCHED SENSING *

Wenjie Chen
Department of Mechanical Engineering
University of California
Berkeley, California 94720
Email: wjchen@me.berkeley.edu

Masayoshi Tomizuka
Department of Mechanical Engineering
University of California
Berkeley, California 94720
Email: tomizuka@me.berkeley.edu

ABSTRACT

The appropriate choice of sensing and how to obtain the desired state information from available sensing for feedback or learning process are essential for most control schemes, including iterative learning control (ILC), to achieve their performance objective. In the multi-joint robots with joint elasticity, the load side joint space measurements are usually not available, even though the load side (end-effector) performance is of ultimate interest. This is termed as mismatched sensing problem. Furthermore, the mismatched uncertainty and mismatched real-time feedback signals in the robots with joint elasticity set further difficulty in achieving high performance. In this paper, a hybrid two-stage model based iterative learning control (ILC) scheme is proposed to deal with the mismatched dynamics. Also, to tackle the mismatched sensing issue, a sensor fusion scheme is developed. An optimization based inverse differential kinematics algorithm and decoupled adaptive kinematic Kalman filter (KKF) are integrated to obtain load side joint space information from the insufficient end-effector measurements. The proposed ILC scheme together with the load side state estimation algorithm is validated through the experimental study on a 6-DOF industrial robot.

INTRODUCTION

In industrial applications, robot manipulators are often required to repeatedly perform a single task. If the robot repeatability is good, the trajectory tracking error will become repetitive from one run to another. In this case, a learning control scheme can be utilized to compensate for the repeatable trajectory tracking errors [1, 2]. While many literatures have suggested various kinds of iterative learning control (ILC) schemes, one of the essential factors for the algorithms to work effectively is to choose and obtain the appropriate error information for the learning process.

For the robots with joint compliance, the load side (end-effector) performance is of the ultimate interest, which can hardly be guaranteed with motor side information alone [3, 4]. Some early work [5, 6] on this topic only considered the single joint case or assumed availability of load side joint measurements. In industrial robots, however, the load side joint encoders are usually not available due to the cost and assembly issues. To overcome this mismatched sensing problem, a low-cost MEMS accelerometer can be easily adopted to measure the end-effector vibration [7]. This provides the possibility of designing control schemes to directly address the end-effector performance.

In decentralized multi-axis robot control, which is typically utilized in industrial robots, load side joint space information is preferred rather than the end-effector information. Thus, a further study of sensor fusion for joint information estimation is necessary. In [8, 9], this problem was investigated utilizing an accelerometer (and a gyroscope) for each joint without the use of motor encoders. The achieved accuracy was acceptable for service robots where positioning tolerance is at the order of millimeters. In [10, 11], the load side state estimation problem was handled with extended Kalman filter (EKF) or particle filter (PF) utilizing both the motor encoders and end-effector accelerometer. The computation load, however, was nontrivial due to the complex dynamic model and the EKF/PF algorithms. In [7], we tackled the same problem using an optimization based inverse differential kinematics algorithm and decoupled adaptive kinematic Kalman filter (KKF) for each joint. The method is computationally easy and will be utilized in this paper to provide

*THIS WORK WAS SUPPORTED BY FANUC LTD., JAPAN.

desired load side state estimates.

Other difficulties due to the mismatched behaviors (including dynamics and sensing) of the robots with joint elasticity include: the disturbance affects the output in a different way from the motor torque input, and the real-time feedback signals are normally from the motor side only instead of the load side. In [12], a model-based ILC approach was developed to learn the error component beyond the first resonant frequency. However, this approach requires an accurate piecewise-linear model to be identified and interpolated for each trajectory in advance, which limits its application. In [13], we proposed a hybrid two-stage model based ILC scheme, where the reference trajectory and the feedforward torque input were both iteratively updated to achieve high bandwidth performance while maintaining convergence property. This scheme will be applied in this paper to deal with the mismatched system dynamics in the robots with joint elasticity.

This paper is organized as follows. The robot modeling and control structure are described first, which is followed by the systematic design of the two-stage ILC scheme. A load side state estimation method is then introduced to retrieve the load side joint information from the measured Cartesian space acceleration for implementation of the ILC scheme in the joint space. Finally, the experimental study of the proposed scheme is presented.

SYSTEM OVERVIEW

Consider an n-joint robot with gear compliance. The robot is equipped with motor side encoders to provide direct measurements of motor side joint positions and velocities for real-time feedback. In addition, a three-dimensional accelerometer is mounted at the robot end-effector to measure the end-effector acceleration in Cartesian space. Note that, if the computing resource and the sensor configuration allow, the end-effector sensor can also be used online. This paper, however, will address the conservative case where the end-effector sensor is for off-line and training use only, which is usually preferred in industry due to the cost saving and the limited real-time computational power.

Robot Dynamic Model

Lagrangian Dynamics The dynamics of the n-joint robot with joint compliance can be formulated as [14]

$$\begin{aligned} &M_\ell(q_\ell)\ddot{q}_\ell + C(q_\ell,\dot{q}_\ell)\dot{q}_\ell + G(q_\ell) + D_\ell\dot{q}_\ell + F_{\ell c}\mathrm{sgn}(\dot{q}_\ell) \qquad (1)\\ &\quad + J^{\mathrm{T}}(q_\ell) f_{ext} = K_J\left(N^{-1}q_m - q_\ell\right) + D_J\left(N^{-1}\dot{q}_m - \dot{q}_\ell\right)\\ &M_m\ddot{q}_m + D_m\dot{q}_m + F_{mc}\mathrm{sgn}(\dot{q}_m) = \tau_m \qquad (2)\\ &\quad - N^{-1}\left[K_J\left(N^{-1}q_m - q_\ell\right) + D_J\left(N^{-1}\dot{q}_m - \dot{q}_\ell\right)\right] \end{aligned}$$

where $q_\ell, q_m \in \mathbb{R}^n$ are the load side and the motor side position vectors, respectively. $\tau_m \in \mathbb{R}^n$ is the motor torque vector. $M_\ell(q_\ell) \in \mathbb{R}^{n\times n}$ is the load side inertia matrix, $C(q_\ell, \dot{q}_\ell) \in \mathbb{R}^{n\times n}$ is the Coriolis and centrifugal force matrix, and $G(q_\ell) \in \mathbb{R}^n$ is the gravity vector. M_m, K_J, D_J, D_ℓ, D_m, $F_{\ell c}$, F_{mc}, and $N \in \mathbb{R}^{n\times n}$ are all diagonal matrices. The (i,i)-th elements of these matrices, M_{mi}, K_{Ji}, D_{Ji}, $D_{\ell i}$, D_{mi}, $F_{\ell ci}$, F_{mci}, and N_i, represent the motor side inertia, joint stiffness, joint damping, load side damping, motor side damping, load side Coulomb friction, motor side Coulomb friction, and gear ratio of the i-th joint, respectively. $f_{ext} \in \mathbb{R}^6$ denotes the external force acting on the robot due to contact with the environment. The matrix $J(q_\ell) \in \mathbb{R}^{6\times n}$ is the Jacobian matrix mapping from the load side joint space to the end-effector Cartesian space.

Decoupling Model Define the nominal load side inertia matrix as $M_n = \mathrm{diag}(M_{n1}, M_{n2}, \cdots, M_{nn}) \in \mathbb{R}^{n\times n}$, where $M_{ni} = M_{\ell,ii}(q_{\ell 0})$, and $M_{\ell,ii}(q_{\ell 0})$ is the (i,i)-th element of the inertia matrix $M_\ell(q_{\ell 0})$ at the home (or nominal) position $q_{\ell 0}$. M_n can be used to approximate the inertia matrix $M_\ell(q_\ell)$. The off-diagonal entries of $M_\ell(q_\ell)$ represent the coupling inertia between the joints. Then, the robot dynamic model can be reformulated as follows

$$\begin{aligned} &M_m\ddot{q}_m + D_m\dot{q}_m = \tau_m - F_{mc}\mathrm{sgn}(\dot{q}_m) \qquad (3a)\\ &\quad - N^{-1}\left[K_J\left(N^{-1}q_m - q_\ell\right) + D_J\left(N^{-1}\dot{q}_m - \dot{q}_\ell\right)\right]\\ &M_n\ddot{q}_\ell + D_\ell\dot{q}_\ell = d^\ell(q) \qquad (3b)\\ &\quad + K_J\left(N^{-1}q_m - q_\ell\right) + D_J\left(N^{-1}\dot{q}_m - \dot{q}_\ell\right) \end{aligned}$$

where all the coupling and nonlinear terms, such as Coriolis force, gravity, Coulomb frictions, and external forces, are grouped into a fictitious disturbance torque $d^\ell(q) \in \mathbb{R}^n$ as

$$\begin{aligned} d^\ell(q) = &\left[M_nM_\ell^{-1}(q_\ell) - I_n\right]\left[K_J\left(N^{-1}q_m - q_\ell\right) + D_J\left(N^{-1}\dot{q}_m - \dot{q}_\ell\right)\right.\\ &- D_\ell\dot{q}_\ell\big] - M_nM_\ell^{-1}(q_\ell)\big[C(q_\ell,\dot{q}_\ell)\dot{q}_\ell\\ &\left.+ G(q_\ell) + F_{\ell c}\mathrm{sgn}(\dot{q}_\ell) + J^{\mathrm{T}}(q_\ell) f_{ext}\right] \qquad (4) \end{aligned}$$

where $q = \left[q_m^{\mathrm{T}},\ q_\ell^{\mathrm{T}}\right]^{\mathrm{T}}$ and I_n is an $n \times n$ identity matrix.

Thus, the robot can be considered as a MIMO system with $2n$ inputs and $2n$ outputs as follows

$$\begin{aligned} q_m(j) &= P_{mu}(z)\tau_m(j) + P_{md}(z)d(j) \qquad (5a)\\ q_\ell(j) &= P_{\ell u}(z)\tau_m(j) + P_{\ell d}(z)d(j) \qquad (5b) \end{aligned}$$

where j is the time index, z is the one step time advance operator in the discrete time domain, and the fictitious disturbance input $d(j)$ is defined as

$$d(j) = d(q(j)) = \left[-[F_{mc}\mathrm{sgn}(\dot{q}_m(j))]^{\mathrm{T}},\ [d^\ell(q(j))]^{\mathrm{T}}\right]^{\mathrm{T}} \qquad (6)$$

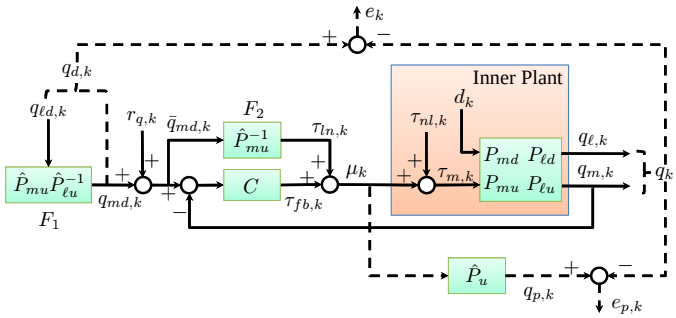

Figure 1. Robot Control Structure with Reference & Torque Update

The continuous time transfer functions from the inputs to the outputs for the i-th joint can be derived from (3) as follows

$$P_{mu}^i(s) = \frac{M_{ni}s^2 + (D_{Ji}+D_{\ell i})s + K_{Ji}}{M_{mi}M_{ni}s^4 + J_{di}s^3 + J_{ki}s^2 + K_{Ji}(D_{mi}+D_{\ell i}/N_i^2)s}$$

$$P_{\ell u}^i(s) = \frac{D_{Ji}s + K_{Ji}}{N_i\left[M_{mi}M_{ni}s^4 + J_{di}s^3 + J_{ki}s^2 + K_{Ji}(D_{mi}+D_{\ell i}/N_i^2)s\right]}$$

$$P_{md^\ell}^i(s) = \frac{D_{Ji}s + K_{Ji}}{N_i\left[M_{mi}M_{ni}s^4 + J_{di}s^3 + J_{ki}s^2 + K_{Ji}(D_{mi}+D_{\ell i}/N_i^2)s\right]}$$

$$P_{\ell d^\ell}^i(s) = \frac{M_{mi}s^2 + (D_{Ji}/N_i^2 + D_{mi})s + K_{Ji}/N_i^2}{M_{mi}M_{ni}s^4 + J_{di}s^3 + J_{ki}s^2 + K_{Ji}(D_{mi}+D_{\ell i}/N_i^2)s}$$

$$P_{md}^i(s) = \left[P_{mu}^i(s),\, P_{md^\ell}^i(s)\right], \quad P_{\ell d}^i(s) = \left[P_{\ell u}^i(s),\, P_{\ell d^\ell}^i(s)\right]$$

where the subscript/superscript i denotes the joint index, and

$$J_{di} = M_{mi}(D_{Ji}+D_{\ell i}) + M_{ni}\left(\frac{D_{Ji}}{N_i^2} + D_{mi}\right)$$

$$J_{ki} = M_{mi}K_{Ji} + \frac{M_{ni}K_{Ji}}{N_i^2} + (D_{Ji}+D_{\ell i})D_{mi} + \frac{D_{Ji}D_{\ell i}}{N_i^2}$$

Robot Controller Structure

It is seen that the robot dynamic model (3) is in a decoupled form since all the variables are expressed in the diagonal matrix form or vector form. Therefore, the robot controller can be implemented in a decentralized form for each individual joint. Note that the MIMO linear system representation (5) is obtained not through local linearization but by considering the fictitious disturbance d as an input which includes the model uncertainties and nonlinearities. The second component of the fictitious disturbance d in (6) influences the output in a different way from the motor torque input τ_m. Thus this robot system is regarded as a MIMO mismatched dynamic system. The compensation of this fictitious disturbance d should be considered in the controller design.

Figure 1 illustrates the control structure for this mismatched system, where the subscript k is the iteration index. It consists of two feedforward controllers, F_1 and F_2, and one feedback controller, C. Here, C can be any linear feedback controller such as a decoupled PID controller to stabilize the system. The feedforward controllers, F_1 and F_2, are designed using the nominal inverse model as

$$q_{md,k}(j) = \hat{P}_{mu}(z)\hat{P}_{\ell u}^{-1}(z)q_{\ell d,k}(j) \triangleq F_1(z)q_{\ell d,k}(j) \tag{7}$$

$$\tau_{ln,k}(j) = \hat{P}_{mu}^{-1}(z)\left[q_{md,k}(j) + r_{q,k}(j)\right] \triangleq F_2(z)\bar{q}_{md,k}(j) \tag{8}$$

where $\hat{\bullet}$ is the nominal model representation of $\bullet$, and $\bullet_d$ is the desired output of $\bullet$. $r_{q,k}$ and $\tau_{nl,k}$ are used as the additional reference and feedforward torque updates, which are generated iteratively by the two-stage ILC algorithm designed later, to compensate for the effect of the fictitious disturbance d. The initialization of these two updates for the first experiment iteration (i.e., initial run) is designed as

$$r_{q,0} = N\hat{K}_J^{-1}(\hat{\tau}_{\ell d} - \hat{M}_n\ddot{q}_{\ell d} - \hat{D}_\ell\dot{q}_{\ell d}) \tag{9}$$

$$\tau_{nl,0} = \tau_{ff,0} - \hat{\tau}_{ln,0} \tag{10}$$

where

$$\hat{\tau}_{\ell d} = \hat{M}_\ell(q_{\ell d})\ddot{q}_{\ell d} + \hat{C}(q_{\ell d},\dot{q}_{\ell d})\dot{q}_{\ell d} + \hat{G}(q_{\ell d}) + \hat{D}_\ell\dot{q}_{\ell d} + \hat{F}_{\ell c}\mathrm{sgn}(\dot{q}_{\ell d}) + J^{\mathrm{T}}(q_{\ell d})\,f_{ext,d} \tag{11}$$

$$\hat{\tau}_{md,0} = \hat{M}_m\ddot{\bar{q}}_{md,0} + \hat{D}_m\dot{\bar{q}}_{md,0} + \hat{F}_{mc}\mathrm{sgn}(\dot{\bar{q}}_{md,0}) \tag{12}$$

$$\tau_{ff,0} = \hat{\tau}_{md,0} + N^{-1}\hat{\tau}_{\ell d} \tag{13}$$

TWO-STAGE ILC SCHEME

To deal with the MIMO mismatched dynamic system described above, the hybrid two-stage ILC scheme proposed in [13] is adopted here with the specific consideration of the assumed sensor configuration.

ILC Basics

Consider the MIMO linear system with the error dynamics and the ILC law in the following form

$$\bar{e}_k(j) = -P_{eu}(z)\bar{u}_k(j) + \bar{r}(j) \tag{14}$$

$$\bar{u}_{k+1}(j) = Q(z)\left[\bar{u}_k(j) + L(z)\bar{e}_k(j)\right] \tag{15}$$

where $\bar{e}$ is the error that the ILC scheme aims to reduce, $\bar{r}$ is the lumped repetitive reference and/or disturbance input to the system, and $\bar{u}$ is the control input updated iteratively by the ILC scheme using the filters $L(z)$ and $Q(z)$. $-P_{eu}$ is the corresponding transfer function from the control input $\bar{u}$ to the error $\bar{e}$. Similar to [1, 15], the following convergence property holds:

Theorem 1. *The ILC system* (14)-(15) *is monotonically and exponentially convergent in the sense that* $\|\bar{u}_k - \bar{u}_\infty\|_p \to 0$ *and* $\|\bar{e}_k - \bar{e}_\infty\|_p \to 0$ *as* $k \to \infty$, *if*

$$\beta = \|Q(z)\left[I - L(z)P_{eu}(z)\right]\|_p < 1 \tag{16}$$

where β is the rate of convergence, I is the identity matrix with appropriate dimension, the p-norm $\|\bullet\|_p = (\sum_i |\bullet_i|^p)^{1/p}$, and

$$\bar{u}_\infty(j) = [I - Q(z) + Q(z)L(z)P_{eu}(z)]^{-1} Q(z)L(z)\bar{r}(j) \quad (17)$$

$$\bar{e}_\infty(j) = [I - Q(z) + Q(z)L(z)P_{eu}(z)]^{-1} [I - Q(z)]\bar{r}(j) \quad (18)$$

Proof. See [13] for details.

Given a fixed Q filter, the optimal learning filter to achieve the fastest convergence is thus obtained from (16) as

$$L^*(z) = \arg\min_{L(z)} \|Q(z)[I - L(z)P_{eu}(z)]\|_p \quad (19)$$

This leads to the plant inversion choice, i.e., $L^*(z) = P_{eu}^{-1}(z)$. Equation (18) shows that the steady state error $\bar{e}_\infty$ vanishes with complete learning (i.e., $Q(z) = I$), which means the effects on $\bar{e}_\infty$ from the repetitive input $\bar{r}$ will be fully compensated. In practice, a zero-phase low-pass filter $Q(z)$ with unity DC gain is typically employed to prevent the effects of high frequency uncertainties/noises from entering the learning process [1]. In order to achieve better performance, it is desired to push the bandwidth of $Q(z)$ as high as possible. Equation (16), however, indicates that the bandwidth of $Q(z)$ may have to be compromised to ensure monotonic convergence and to avoid poor transients in the learning process.

ILC with Reference Update

For robot applications, the ultimate objective is to reduce the load side (end-effector) tracking error. Here, to make the algorithm more general, define the tracking error as $e_k \triangleq q_{d,k} - q_k$. When load side learning is desired and feasible with available information, the tracking error is thus $e_k = e_{\ell,k} \triangleq q_{\ell d,k} - q_{\ell,k}$, with the corresponding transfer functions denoted as $P_u = P_{\ell u}$ and $P_d = P_{\ell d}$. In contrast, if the load side information is not available for learning, only motor side learning can be conducted with $e_k = e_{m,k} \triangleq q_{md,k} - q_{m,k}$. Similarly, the corresponding transfer functions would be $P_u = P_{mu}$ and $P_d = P_{md}$. The dynamics of the general tracking error can be derived from Fig. 1 as

$$\begin{aligned} e_k =& -\hat{P}_{mu}^{-1} P_u S_p \hat{S}_p^{-1} r_{q,k} + (I_n - P_u S_p \hat{P}_u^{-1} \hat{S}_p^{-1}) q_{d,k} \\ & - P_u S_p \tau_{nl,k} + (P_u S_p C P_{md} - P_d) d_k \\ \triangleq& -P_{eu,r} r_{q,k} + \bar{r}_{r,k} \end{aligned} \quad (20)$$

where $S_p = (I_n + CP_{mu})^{-1}$ is the sensitivity function of the closed loop system. The time index j and discrete-time operator z are omitted hereafter for simplicity. From Theorem 1, the tracking performance of the next iteration can be improved with the following reference update scheme using plant inversion learning filter

$$r_{q,k+1} = Q_r(r_{q,k} + L_r^* e_k) \quad (21)$$

$$L_r^* = \hat{P}_{eu,r}^{-1} = \hat{P}_u^{-1} \hat{P}_{mu} \quad (22)$$

If the desired trajectory $q_{d,k}$, the feedforward torque update $\tau_{nl,k}$, and the disturbance d_k are repetitive for each iteration, the monotonic convergence of this ILC scheme (21) can be guaranteed with the following condition

$$\beta_r^* = \left\| Q_r (I_n - \hat{P}_u^{-1} P_u S_p \hat{S}_p^{-1}) \right\|_\infty < 1 \quad (23)$$

This implies that, in order to achieve fast convergence rate without compromising the bandwidth of Q_r, it is desired to reduce the model uncertainties. This can be done by either obtaining a nominal model $\hat{P}_u$ more accurately representing the actual physical plant P_u, or in contrast, making the inner plant (blue shaded area in Fig. 1) behave as the chosen nominal model $\hat{P}_u$. In the next section, an ILC scheme with torque update is introduced to achieve the latter objective, i.e., making $q_k \to \hat{P}_u \mu_k$, where $\mu_k = \tau_{ln,k} + \tau_{fb,k}$ is the torque input to the inner plant.

ILC with Torque Update

Define $e_{p,k} \triangleq q_{p,k} - q_k$ as the model following error between the nominal plant output (i.e., $q_{p,k} \triangleq \hat{P}_u \mu_k$, and $\hat{P}_u = \hat{P}_{\ell u}$ or $\hat{P}_{mu}$) and the actual plant output q_k (i.e., $q_{\ell k}$ or q_{mk}). Then $e_{p,k}$ can be derived as

$$\begin{aligned} e_{p,k} =& -T_u S_p \tau_{nl,k} - \Delta P S_p (C + \hat{P}_{mu}^{-1})(q_{md,k} + r_{q,k}) \\ & + (\Delta P S_p C P_{md} - P_d) d_k \\ \triangleq& -P_{eu,u} \tau_{nl,k} + \bar{r}_{u,k} \end{aligned} \quad (24)$$

where $T_u = \hat{P}_u C P_{mu} + P_u$, and $\Delta P = P_u - \hat{P}_u$

Based on Theorem 1, the ILC scheme to reduce this model following error $e_{p,k}$ is formulated as

$$\tau_{nl,k+1} = Q_u \left(\tau_{nl,k} + L_u^* e_{p,k} \right) \quad (25)$$

$$L_u^* = \hat{P}_{eu,u}^{-1} = \hat{S}_p^{-1} \hat{T}_u^{-1} = \hat{P}_u^{-1} \quad (26)$$

Therefore, if the desired trajectory $q_{md,k}$, the reference update $r_{q,k}$, and the disturbance d_k remain the same for each iteration, the torque ILC scheme (25) will be monotonically converging as long as

$$\beta_u^* = \left\| Q_u \left(I_n - P_u \hat{P}_u^{-1} \right) S_p \right\|_\infty < 1 \quad (27)$$

Similarly, depending on the available error information for learning process, this ILC scheme can be either load side learning (i.e., $q_k = q_{\ell,k}$, $P_u = P_{\ell u}$, and $P_d = P_{\ell d}$), or motor side learning

(i.e., $q_k = q_{m,k}$, $P_u = P_{mu}$, and $P_d = P_{md}$). In practice, the plant inversion in (26) usually encounters numerical difficulty since the relative order of $P_{\ell u}(s)$ or $P_{mu}(s)$ is 3 or 2. Thus it is more favorable to choose $P_u(s) = P_{\ell u}(s)s^2$ or $P_u(s) = P_{mu}(s)s$, both of which have lower relative orders. The corresponding desired learning information for this new choice of P_u is the load side acceleration $\ddot{q}_{\ell,k}$ or the motor side velocity $\dot{q}_{m,k}$, which is available with the assumed sensor configuration (i.e., end-effector accelerometer and motor encoders). By similar derivation, the torque ILC scheme with these changes can still be obtained exactly the same as (25)-(26), and achieves the same objective, i.e., making the inner plant behave as the chosen nominal model. Thus, in the practical implementation such as the following experimental study, these changes will be applied to the torque ILC scheme to avoid numerical instability.

Hybrid Scheme for Two-Stage ILC

In general, for the closed loop system with a satisfactory feedback controller, the sensitivity function S_p will behave as a high-pass filter to mitigate the low frequency error. Therefore, in the convergence condition (27), the low frequency model uncertainty is greatly suppressed by S_p. This allows Q_u to have higher bandwidth without worrying about the low frequency uncertainty. With the effects of the torque ILC, the inner plant will behave like the nominal model (i.e., $q_k \to \hat{P}_u \mu_k$) up to the bandwidth of Q_u. Within this frequency range, the convergence condition of the reference ILC (23) is simplified to

$$\beta_r \approx \left\| Q_r \left(I_n - S_p \hat{S}_p^{-1} \right) \right\|_\infty < 1 \tag{28}$$

which allows to push Q_r to a higher bandwidth for better tracking performance.

Note that the repetitive assumption has been used in the derivation of the aforementioned two ILC schemes. Here, the disturbance d_k depends on the actual robot state q, and is thus not repetitive. This can be relaxed in practice as follows: since the robot basic performance should be already close to satisfactory, the tiny changes of q around desired state q_d in each iteration normally will not result in drastic change in d_k. However, when these two ILC schemes are activated simultaneously, the repetitive assumption will be no longer valid (i.e., $r_{q,k}$ and $\tau_{nl,k}$ are not repetitive from one iteration to another). Therefore, an ad hoc hybrid scheme is designed to reduce the adverse interference of the two ILC stages. Specifically, an iteration-varying gain is applied to each ILC stage as follows

$$\tau_{nl,k+1} = Q_u \left(\tau_{nl,k} + \gamma_{u,k} L_u^* e_{p,k} \right) \tag{29}$$

$$r_{q,k+1} = Q_r (r_{q,k} + \gamma_{r,k} L_r^* e_k) \tag{30}$$

where $\gamma_{u,k} = \mathrm{diag}(\gamma_{u,k}^1, \cdots, \gamma_{u,k}^n)$ and $\gamma_{r,k} = \mathrm{diag}(\gamma_{r,k}^1, \cdots, \gamma_{r,k}^n)$. The two gains $\gamma_{u,k}^i$ and $\gamma_{r,k}^i$ for the i-th joint can be tuned by trial and error, e.g.

$$\gamma_{u,k}^i = \max \left(0.2, \min \left(\frac{6}{\gamma_{u,k-1}^i} \left| \frac{\|e_{p,k}^i\|_2}{\|e_{p,k-1}^i\|_2} - 1 \right|, 1 \right) \right) \tag{31a}$$

$$\gamma_{r,k}^i = \left(1 - 0.8\gamma_{u,k}^i \right) \cdot \min \left(2 \cdot \frac{\|e_k^i\|_\infty}{\|e_0^i\|_\infty}, 1 \right) \tag{31b}$$

with the initialization as $\gamma_{u,0}^i = 1, \gamma_{r,0}^i = 0.2$. The basic idea behind is that once the model following error is becoming stable (i.e., $\frac{\|e_{p,k}^i\|_2}{\|e_{p,k-1}^i\|_2} \approx 1$, which means either the inner plant has behaved as the nominal model or the torque ILC cannot make further improvement), the torque ILC becomes less important and the reference ILC can be further activated with a decreased $\gamma_{u,k}^i$ and an increased $\gamma_{r,k}^i$. In contrast, the torque ILC can take more effects and make further improvement whenever the model following error is still drastically changing from the previous iteration. In order for the torque ILC to perform better, the effects of the reference ILC is accordingly attenuated with a decreased $\gamma_{r,k}^i$. Furthermore, if the maximum tracking error is sufficiently small (i.e., $\frac{\|e_k^i\|_\infty}{\|e_0^i\|_\infty} \approx 0$), the reference ILC becomes less necessary. Thus, the gain $\gamma_{r,k}^i$ is accordingly decreased. However, to maintain the basic convergence rate, the gain $\gamma_{u,k}^i$ for the torque ILC is constrained to be within $[0.2, 1]$ as indicated in (31).

Also it is understood that the nominal models used in two ILC stages should match with each other for the hybrid scheme to perform well. This means these two stages need to be both load side learning or both motor side learning, but not learning on the two sides together, since the nominal behaviors of load side and motor side cannot be achieved simultaneously due to the mismatched dynamics.

As discussed above, the proposed hybrid scheme is aimed to deal with this mismatched dynamics by improving the performance bandwidth of the ILC without compromising the stability. A simple one-joint robot example to illustrate the advantages of this hybrid two-stage ILC scheme over other schemes is demonstrated in [13].

ROBOT LOAD SIDE STATE ESTIMATION

Note that, in the above ILC scheme with load side learning, the required load side joint information (i.e., $q_{\ell,k}$ and $\ddot{q}_{\ell,k}$) cannot be measured directly. Therefore, it is desired to retrieve this information from the available sensing, i.e., by fusing the measured end-effector acceleration with the motor encoder measurements. This estimation problem is addressed in this section by utilizing the scheme developed in [7].

Robot Inverse Kinematics

Basic Differential Kinematics Let $v_e = \left[\dot{p}_e^{\mathrm{T}}, \omega_e^{\mathrm{T}} \right]^{\mathrm{T}} \in \mathbb{R}^6$ denotes the end-effector Cartesian velocity vector composing

of the translational velocity $\dot{p}_e$ and the angular velocity ω_e at the accelerometer mounting point. The kinematic relation between the joint space and the Cartesian space can be described as

$$v_e = J(q_\ell)\dot{q}_\ell \tag{32}$$

Take the time derivative of both sides of (32), which gives

$$\dot{v}_e = J(q_\ell)\ddot{q}_\ell + \dot{J}(q_\ell,\dot{q}_\ell)\dot{q}_\ell \tag{33}$$

Note that the acceleration measured by the end-effector accelerometer is only three-dimensional translational acceleration. Let $\bar{J}(q_\ell) \in \mathbb{R}^{3\times n}$ and $\bar{\dot{J}}(q_\ell,\dot{q}_\ell) \in \mathbb{R}^{3\times n}$ denote the first three rows of the Jacobian matrix, $J(q_\ell)$, and its time derivative, $\dot{J}(q_\ell,\dot{q}_\ell)$, respectively. Then (33) can be rewritten as

$$\bar{J}(q_\ell)\ddot{q}_\ell = \ddot{p}_e - \bar{\dot{J}}(q_\ell,\dot{q}_\ell)\dot{q}_\ell \;\Rightarrow\; \bar{A}\hat{\ddot{q}}_\ell = \bar{b} \tag{34}$$

which becomes a constraint for the successful load side acceleration estimate $\hat{\ddot{q}}_\ell$.

Optimization Based Inverse Kinematics With the robot dynamic model (2), the load side joint position q_ℓ can be roughly estimated as

$$\begin{aligned}\hat{q}^o_\ell = (\hat{D}_J s + \hat{K}_J)^{-1}\Big[&\hat{K}_J N^{-1} q_m + \hat{D}_J N^{-1}\dot{q}_m \\ &- N\left(\tau_m - \hat{M}_m\hat{\ddot{q}}_m - \hat{D}_m\dot{q}_m - \hat{F}_{mc}\mathrm{sgn}(\dot{q}_m)\right)\Big]\end{aligned} \tag{35}$$

where q_m and $\dot{q}_m$ are obtained from motor encoder measurements, and τ_m can be either motor torque command or measured by motor current. The desired trajectory $\ddot{q}_{md}$ is used instead of $\hat{\ddot{q}}_m$ in (35) as approximation. Furthermore, with Euler differentiation of $\hat{q}^o_\ell$, the rough estimate of the load side joint velocity, $\hat{\dot{q}}^o_\ell$, is obtained.

With $\hat{\dot{q}}^o_\ell \triangleq \int \hat{\ddot{q}}^o_\ell dt$, an optimization problem to estimate the load side joint acceleration can be formulated as

$$\min_{\hat{\ddot{q}}_\ell} f(\hat{\ddot{q}}_\ell) = \frac{1}{2}\left\| \int \hat{\ddot{q}}_\ell dt - \int \hat{\ddot{q}}^o_\ell dt \right\|_2^2 \;\text{ s.t. } \bar{A}\hat{\ddot{q}}_\ell = \bar{b} \tag{36}$$

This optimization problem over the whole time series would be impractical to solve. Instead, a point-wise optimization can be performed for each time step to generate the sub-optimal solution. For each time step t_j, let

$$\hat{\dot{q}}^o_\ell(j) = \int_0^{t_j} \hat{\ddot{q}}^o_\ell(t)dt, \quad \hat{\dot{q}}_\ell(j) = \sum_{i=0}^{j} \hat{\ddot{q}}_\ell(i)\Delta t \tag{37}$$

where Δt is the sampling time. Then (36) can be relaxed into a convex optimization problem for each time step t_j as

$$\begin{aligned}&\min_{\hat{\ddot{q}}_\ell(j)} f(\hat{\ddot{q}}_\ell(j)) = \frac{1}{2}\left\|\hat{\ddot{q}}_\ell(j) - \Delta\hat{\ddot{q}}_\ell(j)\right\|_2^2 \\ &\text{s.t. } A_j\hat{\ddot{q}}_\ell(j) = b_j\end{aligned} \tag{38}$$

where $\Delta\hat{\ddot{q}}_\ell(j) = \frac{\hat{\dot{q}}^o_\ell(j) - \hat{\dot{q}}_\ell(j-1)}{\Delta t}$ also includes the accumulated acceleration estimation error not compensated from previous steps. The optimal load side joint acceleration estimate is thus obtained as

$$\hat{\ddot{q}}_\ell(j) = A_j^{\mathrm{T}}\left(A_jA_j^{\mathrm{T}}\right)^{-1}b_j + \left[I - A_j^{\mathrm{T}}\left(A_jA_j^{\mathrm{T}}\right)^{-1}A_j\right]\Delta\hat{\ddot{q}}_\ell(j) \tag{39}$$

where

$$A_j = \bar{J}(q_\ell(j)), \quad b_j = \ddot{p}_e(j) - \bar{\dot{J}}(q_\ell(j),\dot{q}_\ell(j))\dot{q}_\ell(j) \tag{40}$$

Practical Implementation Issues In practice, the acceleration measurement f_a provided by the end-effector accelerometer is the translational acceleration with additional gravity effect expressed in the accelerometer coordinate frame. Thus, the end-effector translational acceleration $\ddot{p}_e$ in the world coordinates can be obtained as

$$\ddot{p}_e = R_a(q_\ell)\, f_a + g \tag{41}$$

where $R_a(q_\ell)$ is the rotation matrix of the accelerometer coordinate frame with respect to the world coordinate frame and $g = \begin{bmatrix}0 & 0 & -9.8\end{bmatrix}^{\mathrm{T}}\mathrm{m/sec}^2$ is the gravity vector expressed in the world coordinate frame.

Furthermore, since the measurements of q_ℓ and $\dot{q}_\ell$ are not available, the rough estimates $\hat{q}^o_\ell$ and $\hat{\dot{q}}^o_\ell$ are used instead in (40) and (41) to calculate $\bar{J}(q_\ell)$, $\bar{\dot{J}}(q_\ell,\dot{q}_\ell)\dot{q}_\ell$, and $R_a(q_\ell)$. These adjustments for the calculation are reasonable under the fact that the tiny discrepancies between the actual motion and the rough estimates do not make much differences in the Jacobian matrices and the orientation matrix.

Adaptive Kinematic Kalman Filter

With the load side rough approximations obtained as $\hat{q}^o_\ell$ in (35) and $\hat{\ddot{q}}_\ell$ in (39) for each joint, the estimation problem for the whole robot can be decoupled into n kinematic Kalman filters (KKF) running in parallel to better estimate the load side joint position and velocity. The discrete time kinematic model for the

Kalman filter is written as

$$\underbrace{\begin{bmatrix} q_\ell(j+1) \\ \dot{q}_\ell(j+1) \end{bmatrix}}_{x(j+1)} = \underbrace{\begin{bmatrix} I & \Delta t I \\ \mathbf{0} & I \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} q_\ell(j) \\ \dot{q}_\ell(j) \end{bmatrix}}_{x(j)} + \underbrace{\begin{bmatrix} \frac{1}{2}\Delta t^2 I \\ \Delta t I \end{bmatrix}}_{B} \underbrace{\hat{\tilde{q}}_\ell(j)}_{u(j)} + w(j) \tag{42a}$$

$$\underbrace{\hat{q}^o_\ell(j)}_{y(j)} = \underbrace{\begin{bmatrix} I & \mathbf{0} \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} q_\ell(j) \\ \dot{q}_\ell(j) \end{bmatrix}}_{x(j)} + v(j) \tag{42b}$$

$$\Rightarrow \quad x(j+1) = Ax(j) + Bu(j) + w(j) \tag{42c}$$

$$y(j) = Cx(j) + v(j) \tag{42d}$$

with the assumption that $1 \le j \le T$ where T is the number of total time steps of the executing trajectory, $x(1) \sim X_1 = \mathcal{N}(\hat{x}_1, P_1)$, $w(j) \sim W_j = \mathcal{N}(\mathbf{0}, Q)$, and $v(j) \sim V_j = \mathcal{N}(\mathbf{0}, R)$. Note that $\hat{q}^o_\ell(j)$ and $\hat{\tilde{q}}_\ell(j)$ are only approximations instead of direct measurements. Thus, to implement this KKF, it is critical to determine the appropriate covariances (i.e., Q and R) for the fictitious noises w and v.

In the ILC application, where off-line processing is available, expectation maximization (EM) algorithm [7, 16] based on maximum likelihood principle can be utilized to estimate the unknowns $\hat{x}_1, P_1, Q$, and R as follows (see [7, 16] for more details):

1. E-step: run Kalman smoother with current best estimates of $\hat{x}_1, P_1, Q$, and R.
2. M-step: update $\hat{x}_1, P_1, Q$, and R as in (43) using the acausal estimations from Kalman smoother.

$$\begin{aligned} \hat{x}_1 &= \hat{x}_{1|T} \qquad\qquad \hat{P}_1 = P_{1|T} \\ \hat{Q} &= \frac{1}{T-1} \sum_{j=2}^{T} \Big[\left(\hat{x}_{j|T} - A\hat{x}_{j-1|T} - Bu_{j-1}\right) \\ &\quad \cdot \left(\hat{x}_{j|T} - A\hat{x}_{j-1|T} - Bu_{j-1}\right)^{\mathrm{T}} \\ &\quad + P_{j|T} - AP^{\mathrm{T}}_{j,j-1|T} - P_{j,j-1|T}A^{\mathrm{T}} + AP_{j-1|T}A^{\mathrm{T}} \Big] \\ \hat{R} &= \frac{1}{T} \sum_{j=1}^{T} \left[\left(y(j) - C\hat{x}_{j|T}\right) \left(y(j) - C\hat{x}_{j|T}\right)^{\mathrm{T}} + CP_{j|T}C^{\mathrm{T}} \right] \end{aligned} \tag{43}$$

   where $\hat{\bullet}_{j|t}$ represents the conditional expectation of $\bullet(j)$ given the information up to the t-th time step.
3. Iterate from E-step until the increment of the expected likelihood is within chosen threshold.
4. Use $\hat{\tilde{q}}_\ell$ in (39) and $\hat{q}_{\ell,j|T}$ from the last Kalman smoother iteration as the required load side state information in the ILC scheme for load side learning.

EXPERIMENTAL STUDY

Test Setup

The proposed method is implemented on a 6-joint industrial robot, FANUC M-16*i*B/20, in Fig. 2. The robot is equipped with built-in motor encoders for each joint. An inertia sensor (Analog Devices, ADIS16400) consisting of a 3-axial accelerometer and a 3-axial gyroscope is attached to the end-effector. The three-dimensional position measurement system, CompuGauge 3D (repeatability of 0.02mm, accuracy of 0.15mm, resolution of 0.01mm), is utilized to measure the end-effector tool center point (TCP) position as a ground truth for performance validation. The sampling rates of all the sensor signals as well as the real-time controller implemented through MATLAB xPC Target are set to 1kHz. System identifications are conducted for each individual joint at several different postures to obtain the nominal dynamic parameters in the dynamic model (3).

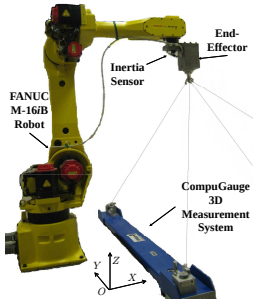

Figure 2. FANUC M-16*i*B Robot System

Algorithm Setup

The zero-phase acausal low-pass filters Q_r and Q_u are obtained as $Q_r(z) = Q_u(z) = Q_1(z^{-1})Q_1(z)$, where $Q_1(z)$ is a diagonal matrix of low-pass filters with cut-off frequencies beyond or around the identified first elastic anti-resonant frequency of the corresponding joint in order to deal with the joint elasticity. With this selection of $Q_r(z)$ and $Q_u(z)$, the frequency responses of β_r in (23) and β_u in (27) using nominal values and load side inertia variations among the workspace are checked to verify the monotonic stability conditions.

To see the superiority of proposed methods (i.e., hybrid two-stage scheme versus single stage scheme, load side learning versus motor side learning), the tracking performances in the experiments will be compared in the following four controller settings implemented for 10 iterations each

1. *RefILC(L)*: Reference ILC only using load side learning, i.e., $\hat{P}_u(s) = \hat{P}_{\ell u}(s)$, $\gamma_{r,k} \equiv 1$, and $\gamma_{u,k} \equiv 0$.
2. *TrqILC(L)*: Torque ILC only using load side learning, i.e., $\hat{P}_u(s) = \hat{P}_{\ell u}(s)s^2$, $\gamma_{r,k} \equiv 0$, and $\gamma_{u,k} \equiv 1$.
3. *RefILC(L)+TrqILC(L)*: Reference ILC plus torque ILC using load side learning, i.e., $\hat{P}_u(s) = \hat{P}_{\ell u}(s)$ for reference ILC and $\hat{P}_u(s) = \hat{P}_{\ell u}(s)s^2$ for torque ILC. $\gamma_{r,k}$ and $\gamma_{u,k}$ are updated as in (31).
4. *RefILC(M)+TrqILC(M)*: Reference ILC plus torque ILC using motor side learning, i.e., $\hat{P}_u(s) = \hat{P}_{mu}(s)$ for reference ILC and $\hat{P}_u(s) = \hat{P}_{mu}(s)s$ for torque ILC. $\gamma_{r,k}$ and $\gamma_{u,k}$ are updated as in (31).

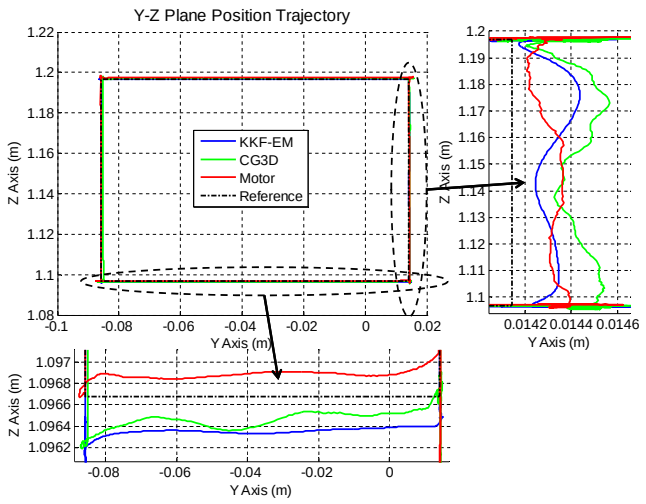

Figure 3. Y-Z Plane TCP Position Estimation (Experiment)

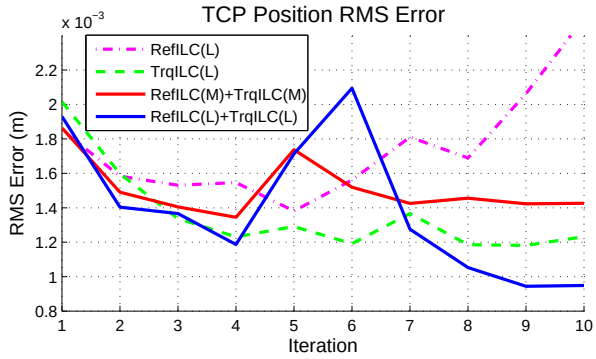

Figure 4. TCP Position RMS Error Comparisons in Iteration Domain

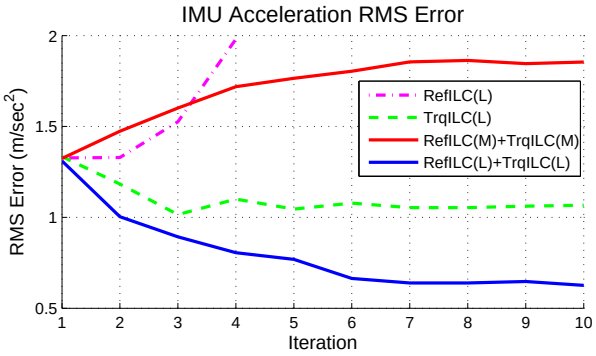

Figure 5. IMU Acceleration RMS Error Comparisons in Iteration Domain

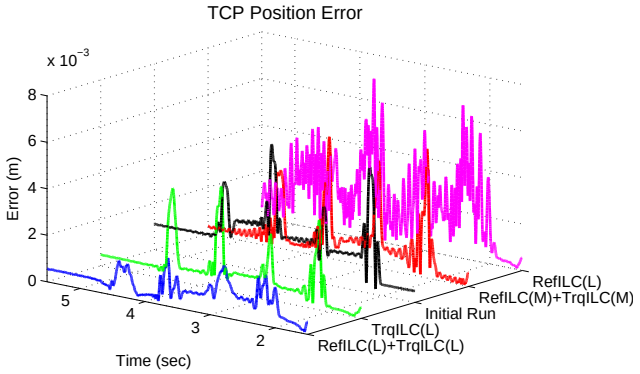

Figure 6. TCP Position Error Comparisons in Time Domain (Initial Run vs. the 10th Iteration of Four Controller Settings)

Experimental Results

The testing TCP trajectory (Fig. 3) is a 10cm $\times$ 10cm square path on the Y-Z plane with fixed orientation, maximum velocity of 1m/sec, and maximum acceleration of 12.5m/sec^2. Besides the reference trajectory (*Reference*), Fig. 3 also shows the actual trajectory (*CG3D*) measured by CompuGauge and the estimated trajectory by proposed method (*KKF-EM*) or by motor encoder measurements directly (*Motor*). It is seen that the *KKF-EM* estimation performs much better than the *Motor* setting by capturing closer transient motion on the Y Axis and with much less offset on the Z Axis.

Figure 4 and Fig. 5 show the iteration domain root-mean-square (RMS) tracking error [1] convergence profiles, while Fig. 6 and Fig. 7 show the time domain error profiles of the initial run and the last runs of these controller settings.

It can be seen that the *RefILC(L)+TrqILC(L)* setting achieves the overall best performance in position tracking and vibration reduction, even though there is non-monotonic transient around the 5-th and the 6-th iterations in the position error convergence due to the interference between the two ILC stages. The *RefILC(L)* setting turns out to be unstable in the iteration domain without the help of torque ILC to reduce mismatched model uncertainty. This implies that the *Q* filter bandwidth (i.e., learning ability) for the *RefILC(L)* setting needs to be further compromised. The *TrqILC(L)* setting looks monotonically convergent but with quite limited improvement in error reductions (especially for moving periods), since the torque ILC aims at modeling matching for the inner plant rather than directly addresses the load side tracking error. The *RefILC(M)+TrqILC(M)* setting does not perform well either, since motor side model can only be used for motor side learning, while the load side (end-effector) performance is not guaranteed and may be even degraded due to the mismatched dynamics.

[1]The Cartesian space error here is defined as the Euclidean distance between the estimated position/acceleration and the actual measured position/acceleration.

CONCLUSIONS

This paper proposed an iterative learning control scheme with sensor fusion for the robots with mismatched dynamics and

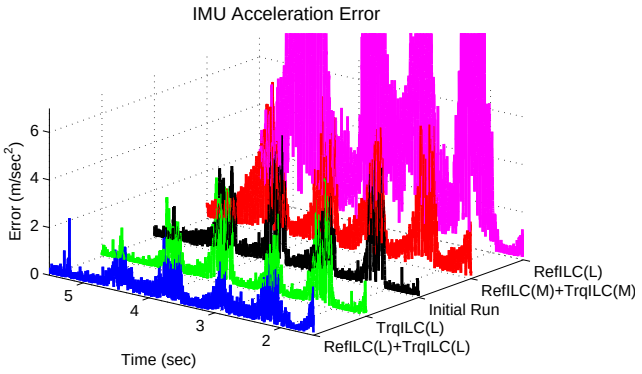

Figure 7. IMU Acceleration Error Comparisons in Time Domain (Initial Run vs. the 10th Iteration of Four Controller Settings)

mismatched sensing. A hybrid model based ILC scheme was applied to the robots with joint compliance for end-effector performance enhancement. The scheme iteratively updated the reference trajectory and the feedforward torque input in a two-stage manner in order to achieve high bandwidth performance while maintaining the robust convergence property. In order to implement the decentralized ILC scheme in joint space, a sensor fusion based load side state estimation method was developed to retrieve the joint space information from the end-effector measurements. The resulting ILC scheme and the load side state estimation method were validated through the experimental study on a 6-DOF industrial robot. It has been shown that for the interested end-effector position tracking and vibration mitigation performance, the proposed scheme showed the benefits of sensor fusion utilizing end-effector measurements for learning process, as well as the advantage of the hybrid two-stage ILC scheme to deal with the mismatched dynamics. For future work, it is worthwhile to further investigate the systematic way of hybrid two-stage ILC scheme synthesis, especially the optimal tuning of the two learning gains in (31) for any given trajectory.

REFERENCES

[1] Bristow, D. A., and Tharayil, M., 2006. "A survey of iterative learning control: A learning-based method for high-performance tracking control". *IEEE Control Systems Magazine*(June), pp. 96–114.

[2] Arimoto, S., Kawamura, S., and Miyazaki, F., 1984. "Bettering operation of Robots by learning". *Journal of Robotic Systems*, **1**(2), pp. 123–140.

[3] Chen, W., Kong, K., and Tomizuka, M., 2009. "Hybrid Adaptive Friction Compensation of Indirect Drive Trains". In Proceedings of the 2009 ASME Dynamic Systems and Control Conference (DSCC), Vol. 2, pp. 313–320.

[4] Wallén, J., Norrlof, M., and Gunnarsson, S., 2008. "Arm-side evaluation of ILC applied to a six-degrees-of-freedom industrial robot". In Proceedings of the 17th IFAC World Congress, pp. 13450–13455.

[5] Miyazaki, F., Kawamura, S., Matsumori, M., and Arimoto, S., 1986. "Learning control scheme for a class of robot systems with elasticity". In Proceedings of the 25th IEEE Conference on Decision and Control,, Vol. 25, pp. 74–79.

[6] Luca, A. D., and Ulivi, G., 1992. "Iterative Learning Control of Robots with Elastic Joints". In Proceedings of IEEE International Conference on Robotic and Automation, Vol. 3, pp. 1920–1926.

[7] Chen, W., and Tomizuka, M., 2012. "Load Side State Estimation in Elastic Robots with End-effector Sensing". In Proceedings of the 2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM).

[8] Quigley, M., Brewer, R., Soundararaj, S., Pradeep, V., Le, Q., and Ng, A., 2010. "Low-cost accelerometers for robotic manipulator perception". In the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6168–6174.

[9] Cheng, P., and Oelmann, B., 2010. "Joint-Angle Measurement Using Accelerometers and Gyroscopes A Survey". *IEEE Transactions on Instrumentation and Measurement*, **59**(2), pp. 404–414.

[10] Henriksson, R., Norrlof, M., Moberg, S., Wernholt, E., and Schon, T., 2009. "Experimental comparison of observers for tool position estimation of industrial robots". In Proceedings of the 48th IEEE Conference on Decision and Control (CDC), pp. 8065–8070.

[11] Axelsson, P., Karlsson, R., and Norrlöf, M., 2011. Bayesian State Estimation of a Flexible Industrial Robot. Tech. Rep. LiTH-ISY-R-3027, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, Oct.

[12] Hakvoort, W., Aarts, R., van Dijk, J., and Jonker, J., 2007. "Model-based iterative learning control applied to an industrial robot with elasticity". In Proceedings of IEEE Conference on Decision and Control (CDC), pp. 4185–4190.

[13] Chen, W., and Tomizuka, M., 2012. "A Two-Stage Model Based Iterative Learning Control Scheme for a Class of MIMO Mismatched Linear Systems". In Proceedings of the ASME 2012 International Symposium on Flexible Automation (ISFA).

[14] Wang, C.-C., 2008. "Motion Control of Indirect-drive Robots: Model Based Controller Design and Performance Enhancement Based on Load-side Sensors". PhD thesis, University of California at Berkeley.

[15] Padieu, F., and Su, R., 1990. "An H_∞ approach to learning control systems". *International Journal of Adaptive Control and Signal Processing*, **4**(6), Nov., pp. 465–474.

[16] Digalakis, V., Rohlicek, J., and Ostendorf, M., 1993. "ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition". *IEEE Transactions on Speech and Audio Processing*, **1**(4), pp. 431–442.