# A Norm Optimal Iterative Learning Control Framework towards Internet-Distributed Hardware-in-the-Loop Simulation

Xinyi Ge, Mark J. Brudnak, Jeffrey L. Stein, and Tulga Ersal*

*Abstract* – **Internet-Distributed Hardware-In-the-Loop (ID-HIL) simulation couples geographically-dispersed hardware-in-the-loop setups into an integrated system simulation over the Internet. The challenge is that the Internet is a dynamic system on its own with delay, jitter, and loss that can interfere with the dynamics of the integrated system, lowering the simulation fidelity. This paper focuses on the effects of delay in a linear time invariant ID-HIL system and presents a Norm Optimal Iterative Learning Control (NO-ILC) based method to improve the fidelity of the integrated system. It is shown that the method guarantees monotonic convergence; i.e., with each iteration, the fidelity of the simulation will improve further. A recursive, data-driven calculation method is also presented that avoids the need for ad-hoc tuning of learning gains and does not require any knowledge about the setups that are being integrated. A simulated networked mass-spring-damper system is used to demonstrate the technique.**

## I. Introduction

Hardware-In-the-Loop (HIL) simulation can be viewed as a combination of physical and virtual prototyping: It is a setup that connects the physical prototypes of some components of a system with the virtual simulation of the rest of the system in closed loop. HIL, therefore, has many advantages such as cost effectiveness, rapid prototyping, and higher fidelity compared to purely virtual simulations [1]. These advantages lead to its wide adoption in many areas such as automotive [1, 2], aerospace [3, 4], manufacturing [5], robotics [6, 7].

Recently, Internet-Distributed Hardware-In-the-Loop (ID-HIL) simulation, as a networked HIL simulation, has attracted interest, because it allows different subsystems to be integrated into a system-level simulation regardless of their geographic locations, thereby fostering geographically dispersed concurrent systems engineering. ID-HIL has found applications in, for example, earthquake engineering [8-12] and automotive engineering [13-17].

One of the key challenges in ID-HIL simulation is ensuring fidelity. Fidelity here refers specifically to how close the dynamics of the network-integrated system are to the dynamics of the physically coupled system [18]. Several metrics exist in the literature to characterize fidelity, both in frequency domain [14, 19-21] and time domain [15, 17]. Methods proposed in literature to improve fidelity include selecting appropriate coupling points [22], using feedback [23-25] and observer based approaches [13, 14]. Recently, an Iterative Learning Control (ILC) based framework was proposed in [18] and a PD-type ILC was applied to increase the fidelity of an ID-HIL setup as a model-free control technique over the network.

This paper builds on the ILC-based framework developed in [18] and addresses the research question of how to design the learning algorithm so that a monotonic and fast convergence can be achieved in a linear time invariant (LTI) ID-HIL system. Even though ID-HIL systems are nonlinear in general, linear systems are considered within the scope of this paper as a first step. The significance of this research question is that a design method is desired to avoid ad hoc tuning of the learning gains. Furthermore, it is important to achieve monotonic convergence due to hardware safety concerns; aggressive transients can damage the hardware, even if asymptotic convergence may be ensured. Finally, a fast convergence is important to keep the iteration number needed to achieve a certain fidelity level as low as possible for minimum experimental cost.

To achieve these goals, this paper utilizes a Norm Optimal ILC (NO-ILC) approach. The key process in NO-ILC is to solve a quadratic cost function, which was originally formulated by Amann [26] and Lee [27] for single-input-single-output (SISO) LTI case. In [26], the input updating law is proposed as a combination of feedback and feed-forward using the system model. However, in an ID-HIL system, this feedback cannot be implemented without delay. Furthermore, a solution that does not require any knowledge about the system model is desired, because the unavailability of such a model may be the motivation for putting the hardware in the loop in the first place, or, even if the models are available, it may be infeasible to share them due to intellectual property or security reasons. In [27], the input updating law is obtained purely from the input/output data from the previous iteration and the system Markov parameters. However, in an ID-HIL system, the Markov parameters are unlikely to be known a priori. In [28] this NO-ILC idea is used in combination with an estimation of Markov parameters and validated on a linear motor position system. Nevertheless, the estimation process in [28] may cause singularity issues in an ID-HIL framework as will be discussed in detail later. Thus, this paper develops a recursive method to estimate the Markov parameters of an LTI ID-HIL system to be subsequently used in the NO-ILC approach. Furthermore, the SISO results are extended to the

---

This work was partially supported by the Automotive Research Center (ARC), a U.S. Army Center of Excellence in Modeling and Simulation of Ground Vehicles.

X. Ge, J.L. Stein, and T. Ersal are with the University of Michigan, Ann Arbor. M.J. Brudnak is with the U.S. Army RDECOM-TARDEC, Warren, MI.

* Corresponding author (tersal@umich.edu)

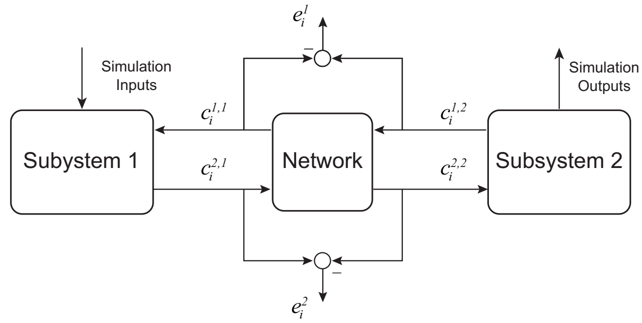


Fig. 1. Illustration of an ID-HIL system with two subsystems (Subsystem 1 and 2). The variable $c_i^{p,q}$ represents the $p$-th coupling variable on the Subsystem $q$ side of the network in the $i$-th iteration. The variable $e_i^j$ represents the instantaneous error in the $j$-th coupling variable in the $i$-th iteration and is used to characterize fidelity.

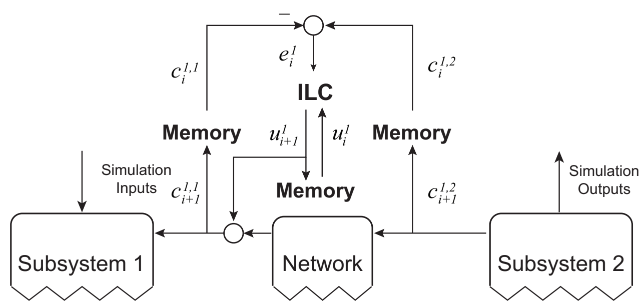


Fig. 2. Proposed ILC-based framework for iteratively improving ID-HIL fidelity. The variables $u_i^j$ and $e_i^j$ represent the $j$-th input and output for the ILC controller in the $i$-th iteration.

MIMO case, so that the method allows for fully accounting for the interactions between the coupling variables.

To summarize, the contributions of this paper are (1) a recursive formulation to estimate Markov parameters within the NO-ILC approach; (2) extension of the NO-ILC approach to MIMO case; and (3) application of the NO-ILC approach to the problem space of improving fidelity of ID-HIL simulations, providing a systematic way to achieve monotonic and fast convergence without a priori system knowledge.

The rest of this paper is organized as follows. Section II.A briefly reviews the ILC-based framework for the ID-HIL application and the fidelity metric. NO-ILC and its monotonic convergence criteria are reformulated for MIMO LTI system in Section II.B. The recursive formulation of the estimation of Markov parameters is given in Section II.C. In Section III, the theory is applied to a simulated networked mass-spring-damper system to highlight the potential performance of the proposed approach.

## II. Problem Formulation

### A. ILC-based ID-HIL Framework

ILC is a control strategy to improve tracking performance in systems operated under repeatable conditions [29, 30]. It has been applied in, for example, manufacturing [28], robotics [31] and chemical processing [32, 33], and recently has found application in network-integrated systems, as well, for the purpose of eliminating communication delays, like network control systems [34] and ID-HIL systems [18]. This section briefly reviews the framework developed in [18] to layout the foundation for the rest of this paper.

Consider the ID-HIL system in Fig. 1. The two subsystems, both of which may include hardware or models, are integrated over a network through coupling variables $c_i^1$ and $c_i^2$. The subscript $i$ is an index for the iteration. In general, the network includes delay, jitter, and loss. However, based on the assumption that losses are negligible and jitter can be avoided by introducing buffers, this work focuses on delay.

In this framework, the fidelity of the coupling variables is evaluated through the following error metrics [18]:

$$\begin{aligned} e_i^1(t) &= c_i^{1,2}(t) - c_i^{1,1}(t) \\ e_i^2(t) &= c_i^{2,1}(t) - c_i^{2,2}(t) \end{aligned} \tag{1}$$

Then, maximizing the simulation fidelity corresponds to minimizing the error metric $e_i^j$ for all coupling variables. Ideally, $e_i^j(t) = 0$ means the network effect is totally eliminated. Different weights may be used if some coupling errors are more critical than the others for a particular simulation output of interest in a given application. For simplicity, equal weight is placed on all coupling variables in this work.

Figure 2 shows the detailed implementation of the ILC-based framework for one of the coupling signals as an example. The error defined in (1) is provided offline as the tracking error to the ILC algorithm for the purpose of calculating the input sequence for the next iteration to attenuate the error.

Previous work used a PD-type learning function as a proof of concept. However, a systematic way to tune the learning gains was not provided. Furthermore, the convergence was not monotonic. Finally, ILC was applied to each coupling error independently; thus the interactions between the coupling variables were not taken into account in the ILC design. The rest of this paper addresses these problems in a linear time-invariant context.

### B. Norm Optimal ILC for MIMO Case with Iteration-invariant Disturbance

Within the ILC-based ID-HIL framework of this paper, the terms *input* and *output* are used to specifically refer to the ILC control inputs and coupling variable errors, respectively. The ID-HIL system may contain other external signals driving the system and other signals of interest that are observed; these signals are referred to as *simulation inputs* and *simulation outputs*. Given this terminology, it is important to note that the number of inputs and outputs in this framework are always equal, because each coupling variable induces exactly one input and one output. Furthermore, an ID-HIL system contains at least two coupling variables forming a closed loop, because a one-way coupling would defeat the purpose of co-simulating the subsystems together. Hence, a multi-input multi-output control problem arises with the special property of equal number of inputs and outputs. This section develops a norm optimal ILC approach for this special MIMO problem in the LTI context that guarantees a monotonic convergence without the need to know the dynamics of the subsystems that are being integrated.

Consider the following discrete LTI MIMO system with an iteration-invariant disturbance

$$\begin{cases} x_i(t+1) = Ax_i(t) + Bu_i(t) + B_w w_i(t) \\ y_i(t) = Cx_i(t) + Du_i(t) \end{cases} . \tag{2}$$

The state space matrices $A, B, C, D, B_w$ are assumed to be time and iteration invariant. The variables $i \in [0,k]$ and $t \in [0, T-1]$ denote the iteration and time index, with $T$ being the total number of time steps. The states, inputs, outputs, and disturbances are given by $x_i(t) \in \Re^n$ , $u_i(t) \in \Re^m$ , $y_i(t) \in \Re^m$ , $w_i(t) \in \Re^l$ , respectively, where $n, m$ , and $l$ denote the number of states, inputs/outputs, and disturbances, respectively. Simulation outputs are not shown, because they are not used for the control purposes.

Using the lifted system representation [35], the input-output relationship can be re-written as

$$\begin{aligned} &y_i(0) = Cx_i(0) + Du_i(0); \\ &y_i(1) = CAx_i(0) + CBu_i(0) + Du_i(1) + CB_w w_i(0); \\ &y_i(2) = CA^2 x_i(0) + CABu_i(0) + CBu_i(1) + Du_i(2) \\ &\quad +CAB_w w_i(0) + CB_w w_i(1); \\ &\quad \vdots \end{aligned} \tag{3}$$

which can be given in matrix form as

$$y_i = Gu_i + G_w w_i + \begin{bmatrix} C & \cdots & CA^{T-1} \end{bmatrix}^T x_i(0) \ , \tag{4}$$

where $G \in \Re^{mT \times mT}$ is the Markov matrix relating inputs $u_i \in \Re^{mT}$ and outputs $y_i \in \Re^{mT}$ , and $G_w \in \Re^{lT \times lT}$ is the matrix relating disturbances $w_i \in \Re^{lT}$ and outputs, with

$$y_i = \begin{bmatrix} y_i(0) \\ \vdots \\ y_i(T-1) \end{bmatrix}; u_i = \begin{bmatrix} u_i(0) \\ \vdots \\ u_i(T-1) \end{bmatrix}; G = \begin{bmatrix} D & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ CB & \ddots & \mathbf{0} & \mathbf{0} \\ \vdots & CB & \ddots & \mathbf{0} \\ CA^{T-2}B & \cdots & CB & D \end{bmatrix} \tag{5}$$

Let $\Delta y_i = \begin{bmatrix} y_i(0) - y_{i-1}(0) & \cdots & y_i(T-1) - y_{i-1}(T-1) \end{bmatrix}^T$ and $\Delta u_i = \begin{bmatrix} u_i(0) - u_{i-1}(0) & \cdots & u_i(T-1) - u_{i-1}(T-1) \end{bmatrix}^T$ denote the difference in outputs and inputs, respectively, between two consecutive iterations. As long as the disturbances are iteration-invariant and initial conditions are identical, the following relationship holds

$$\Delta y_i = G \Delta u_i \ . \tag{6}$$

Let $y_d(t) = \begin{bmatrix} y_d(0) & \cdots & y_d(T-1) \end{bmatrix}^T$ denote the desired output. Then, the error $e_i$ to be minimized by ILC can be defined as

$$\begin{aligned} e_i = y_d - y_i &= \begin{bmatrix} y_d(0) - y_i(0) \\ \vdots \\ y_d(T-1) - y_i(T-1) \end{bmatrix} \\ &= y_d - y_{i-1} - (y_i - y_{i-1}) = e_{i-1} - \Delta y_i = e_{i-1} - G\Delta u_i \end{aligned} \tag{7}$$

The aim of ILC is to modify the input signal $u_i$ to make the output $y_i$ match the desired trajectory $y_d$ . A very common way to evaluate the tracking performance is to look at the norm of the error signal $e_i$ . In this case, improving the ILC performance is equivalent to reducing the norm of the error signal. At the same time, input effort change $\Delta u_i$ is also a consideration in designing the ILC controller; large $\Delta u_i$ can lead to robustness issues. Hence, to find the best input $u_i$ , an optimization problem is formulated with the following cost function [26, 27]:

$$J_i(u_i) = \|e_i\|_Q^2 + \|u_i - u_{i-1}\|_R^2 \ . \tag{8}$$

Here, $Q$ and $R$ are diagonal weighting matrices pre-defined by the ILC designer. The first term penalizes the tracking error for each iteration and the second term penalizes the input difference between the current iteration and previous iteration. $J_i$ is the total cost, which is a function of the input sequence $u_i$ . The control goal is to find the input sequence so that the total cost is minimized, which translates to solving the following differential equation

$$\frac{\partial J_i(u_i)}{\partial \Delta u_i} = \frac{\partial \|e_i\|_Q^2}{\partial \Delta u_i} + \frac{\partial \|u_i - u_{i-1}\|_R^2}{\partial \Delta u_i} = 0 \ . \tag{9}$$

The first term in (9) can be derived as following

$$\begin{aligned} \frac{\partial \|e_i\|_Q^2}{\partial \Delta u_i} &= \frac{\partial}{\partial \Delta u_i}[(e_{i-1} - G\Delta u_i)^T Q (e_{i-1} - G\Delta u_i)] \\ &= \frac{\partial}{\partial \Delta u_i}[(e_{i-1}^T Q e_{i-1} - 2e_{i-1}^T QG\Delta u_i + \Delta u_i^T G^T QG \Delta u_i)] \end{aligned} \tag{10}$$

and the second term can be rewritten as

$$\frac{\partial \|u_i - u_{i-1}\|_R^2}{\partial \Delta u_i} = \frac{\partial}{\partial \Delta u_i}[\Delta u_i^T R \Delta u_i] \tag{11}$$

Combining (10) and (11), (9) leads to the following result

$$u_i = u_{i-1} + (G^T QG + R)^{-1} G^T Q e_{i-1} \tag{12}$$

The above equation is the ILC updating law for the input sequence. The current input sequence is calculated based on the input and error sequence from the previous iteration.

Once the input updating law is obtained, another consideration in ILC design is the convergence. For the input updating law above, the error dynamics can be derived as follows

$$e_i = e_{i-1} - G\Delta u_i = [I - G(G^T QG + R)^{-1} G^T Q] e_{i-1} \tag{13}$$

Hence, asymptotic convergence can be achieved if the spectral radius of the matrix multiplier of $e_{i-1}$ is less than 1; i.e.,

$$\rho(I - G(G^T QG + R)^{-1} G^T Q) < 1 \tag{14}$$

More ideally, however, monotonic convergence is preferred in an ID-HIL application, because bad transients may harm the hardware. This means the tracking error is desired to decrease after each iteration. From (13), the criterion for monotonic convergence can be obtained as

$$\|I - G(G^T QG + R)^{-1} G^T Q\| < 1 \tag{15}$$

These results agree with the results reported in [27] for the SISO case.

Notice that choosing $R = \mathbf{0}$ will make (14) and (15) zero. However, NO-ILC with $R = \mathbf{0}$ is the same as inverting the plant to get the ideal inputs, which typically is avoided in a real application. Hence, choosing $R$ can be regarded as adjusting a trade-off between ILC convergence rate and robustness. A smaller $R$ would increase the convergence rate, but potentially at the cost of robustness.

Consider the following choices for $Q$ and $R$

$$R = \lambda Q = \lambda I \tag{16}$$

where $I$ is the identity matrix and $\lambda$ is a positive scalar. Setting $Q=I$ corresponds to giving all coupling variable errors equal weight. Choosing $R=\lambda Q$ simplifies (14) and (15) as follows

$$\rho(I-G(G^T G+\lambda I)^{-1} G^T)<1 \tag{17}$$

$$\left\| I-G(G^T G+\lambda I)^{-1} G^T \right\|<1 \tag{18}$$

Then, $\lambda$ becomes the only design parameter to be tuned to adjust the trade-off between convergence rate and robustness.

In general, the calculation of $G$ can rely on the system matrices. In an ID-HIL application, however, it is desired to not require any knowledge about the system dynamics. This implies that $G$ needs to be identified from the available input and output data, which is addressed in the next section.

### C. Recursive Estimation of Markov Parameters

In an ID-HIL application, it is likely that the $G$ matrix, composed of the Markov parameters of the LTI system, is unknown or partially unknown. However, leveraging the iterative nature of the approach, the $G$ matrix could be identified. This section describes one way to perform this identification.

In the ILC-based ID-HIL framework shown in Fig. 2, the only a priori knowledge about the Markov parameters of the system is that the diagonal terms of *D,* and therefore *G,* are all -1 because of the direct feedthrough from the ILC input to the coupling error. Thus, initially, $G=diag(-1)$ can be used in (12) to calculate the input sequence for the next iteration. Note that, using $G=diag(-1)$ in (12) makes the ILC updating law a P-type ILC, a very common learning algorithm when system dynamics are unknown [36]. If $m$ iterations are thus performed for an $m$ input/output system, then the relationship in (6) could be used to find $G$ as

$$G=\Delta Y_i \cdot \Delta U_i^{-1} \tag{19}$$

with $\Delta Y_i$ and $\Delta U_i$ defined as

$$\Delta Y_i=\begin{bmatrix} \Delta\mathbf{y}_i(0) & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \Delta\mathbf{y}_i(T-1) & \cdots & \Delta\mathbf{y}_i(0) \end{bmatrix}$$
$$\Delta U_i=\begin{bmatrix} \Delta\mathbf{u}_i(0) & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \Delta\mathbf{u}_i(T-1) & \cdots & \Delta\mathbf{u}_i(0) \end{bmatrix} \tag{20}$$

where $\Delta\mathbf{y}_i(\mathrm{t})$ and $\Delta\mathbf{u}_i(\mathrm{t})$ are given by

$$\Delta\mathbf{y}_i(t)=\begin{bmatrix} \Delta y_i^1(t) & \cdots & \Delta y_{i-m+1}^1(t) \\ \vdots & \cdots & \vdots \\ \Delta y_i^m(t) & \cdots & \Delta y_{i-m+1}^m(t) \end{bmatrix}$$
$$\Delta\mathbf{u}_i(t)=\begin{bmatrix} \Delta u_i^1(t) & \cdots & \Delta u_{i-m+1}^1(t) \\ \vdots & \cdots & \vdots \\ \Delta u_i^m(t) & \cdots & \Delta u_{i-m+1}^m(t) \end{bmatrix} \tag{21}$$

For this particular ILC-based framework, however, $\Delta U_i$ is not guaranteed to have full rank after $m$ iterations. For instance, if coupling errors are initially zero (e.g., due to the time needed for the subsystems to start responding to simulations inputs), this would lead to a zero $\Delta\mathbf{u}_i(0)$, which in turn would lead to a singular $\Delta U_i$. Therefore, directly using (19) is, in general, not possible. Hence, a recursive formulation is developed in the remainder of this section. The formulation is explained on a simple example for space issues, but it can be generalized.

As an example, consider an ID-HIL system as shown in Fig. 1. Assume that Subsystem 1 and 2 have relative degrees of $r_1$ and $r_2$, respectively, and the Internet has $r_3$ step delay. After two iterations for this two input/output system, $\Delta\mathbf{u}_2(0)=\mathbf{0},\ldots,\Delta\mathbf{u}_2(r_1-1)=\mathbf{0}$, $\Delta\mathbf{u}_2(r_1)=[0,0;\alpha_1,\beta_1]$, …, $\Delta\mathbf{u}_2(r_1+r_2+r_3)=\left[0,0;\alpha_{r_2+r_3+1},\beta_{r_2+r_3+1}\right]$ for $\alpha_z \neq \beta_z \neq 0$, $\forall z \in [1, r_2+r_3+1]$. This makes $\Delta U_2$ singular. Recursion can now be applied to approximate the Markov matrix as follows:

Row 1 to $\mathrm{r}_1$ : leave $CA^{r_1-2}B,\ldots,CB,D$ as unidentified

---

Row $r_1+1: D\Delta\mathbf{u}_2(r_1)=\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}\begin{bmatrix} 0 & 0 \\ \alpha_1 & \beta_1 \end{bmatrix}=\Delta\mathbf{y}_2(r_1)$

$\Rightarrow$ solve for $D_{12}, D_{22}$

---

$\vdots$

Row $r_1+r_2+r_3+1: CA^{r_2+r_3-1}B\Delta\mathbf{u}_2(r_1)+\cdots$

$$+D\Delta\mathbf{u}_2(r_1+r_2+r_3)=\Delta\mathbf{y}_2(r_1+r_2+r_3)$$
$$=\begin{bmatrix} CA^{r_2+r_3-1}B_{11} & CA^{r_2+r_3-1}B_{12} \\ CA^{r_2+r_3-1}B_{21} & CA^{r_2+r_3-1}B_{22} \end{bmatrix}\begin{bmatrix} 0 & 0 \\ \alpha_1 & \beta_1 \end{bmatrix}+\cdots$$
$$+\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}\begin{bmatrix} 0 & 0 \\ \alpha_{r_2+r_3+1} & \beta_{r_2+r_3+1} \end{bmatrix}$$

$\Rightarrow$ solve for $CA^{r_2+r_3-1}B_{12}, CA^{r_2+r_3-1}B_{22}$

---

Row $r_1+r_2+r_3+2: CA^{r_2+r_3}B\Delta\mathbf{u}_2(r_1)+\cdots$

$$+D\Delta\mathbf{u}_2(r_1+r_2+r_3+1)=\Delta\mathbf{y}_2(r_1+r_2+r_3+1)$$
$$=\begin{bmatrix} CA^{r_2+r_3}B_{11} & CA^{r_2+r_3}B_{12} \\ CA^{r_2+r_3}B_{21} & CA^{r_2+r_3}B_{22} \end{bmatrix}\begin{bmatrix} 0 & 0 \\ \alpha_1 & \beta_1 \end{bmatrix}+\cdots$$
$$+CB\begin{bmatrix} 0 & 0 \\ \alpha_{r_2+r_3+1} & \beta_{r_2+r_3+1} \end{bmatrix}+\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}\Delta\mathbf{u}_2(r_1+r_2+r_3+1)$$

$\Rightarrow$ solve for $CA^{r_2+r_3}B_{12}, CA^{r_2+r_3}B_{22}, D_{11}, D_{21}$

---

$\vdots$

Row $T: CA^{T-2-r_1}B\Delta\mathbf{u}_2(r_1)+\cdots+D\Delta\mathbf{u}_2(T-1)=\Delta\mathbf{y}_2(T-1)$

$\Rightarrow$ solve for $CA^{T-2-r_1}B_{12}, CA^{T-2-r_1}B_{22}, CA^{T-r_1-r_2-r_3-3}B_{11}$, $CA^{T-r_1-r_2-r_3-3}B_{21}$

---

With recursion, only the last $4r_1+2r_2+2r_3-2$ Markov parameters are left unidentified in this case of 2 inputs/outputs. Their values can be set using the last set of estimated Markov parameters, with the rationale that the simulation is long enough for the Markov parameters to settle to their steady state values.

This recursion calculation can also be extended to MIMO cases with system input/output number $m>2$.

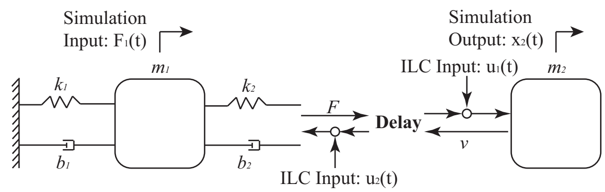

Fig. 3. Networked mass-spring-damper example

TABLE 1: PARAMETERS OF THE EXAMPLE SYSTEM

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $m_1$ | 1 kg | $b_1$ | 4 Ns/m |
| $m_2$ | 2 kg | $b_2$ | 1 Ns/m |
| $k_1$ | 5 N/m | $F_1(t)$ | 3+sin(0.3$t$) N |
| $k_2$ | 3 N/m | | |

## III. EXAMPLE

Consider the simulation of the networked mass-spring-damper system shown in Fig. 3. In this example, the simulation input is the force applied to mass $m_1$ and is treated as an iteration-invariant disturbance acting on the integrated system. The simulation output is the displacement of the mass $m_2$. The ILC outputs are the coupling errors (not shown in Fig 3 for clarity) and the ILC inputs are placed at the receiving ends of the network.

Table 1 lists the system parameters for this example. The network is considered as a pure, constant delay. The delay, $\tau$, is chosen as a multiplicity of the sampling frequency $Ts = 0.1$ s of the system; i.e., $\tau = 0.1$ s, $0.2$ s, etc.

Figure 4 illustrates how the network delay degrades the simulation fidelity. With a delayed integration, the simulation output will deviate from its original trajectory; i.e., the trajectory for the physically coupled system.

Applying NO-ILC to the networked system can help reduce the coupling errors and thus improve the fidelity of the simulation as illustrated in Fig 5 for the case of $\tau = 0.2$ s and $\lambda = 2$ as an example. Iteration 0 represents the two systems integrated over the network without ILC applied. Because this is a 2 input/output system, NO-ILC starts at the second iteration as explained in Section II.C, and the first two iterations are performed with $G = diag(-1)$, which corresponds to a P-type ILC. Once it is activated, NO-ILC successfully achieves a monotonic convergence ratio (as defined in (18)) of 0.92, and the 2-norm of the coupling error

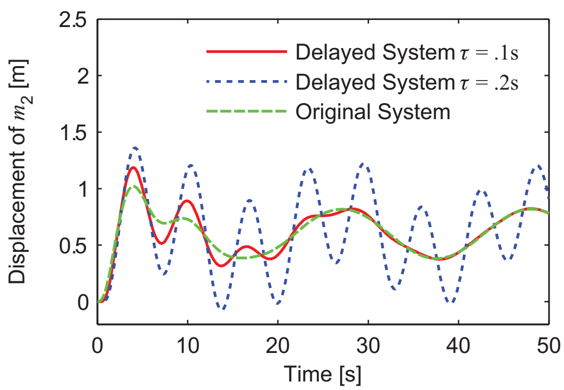

Fig. 4. Example degradation of simulation fidelity due to network delay

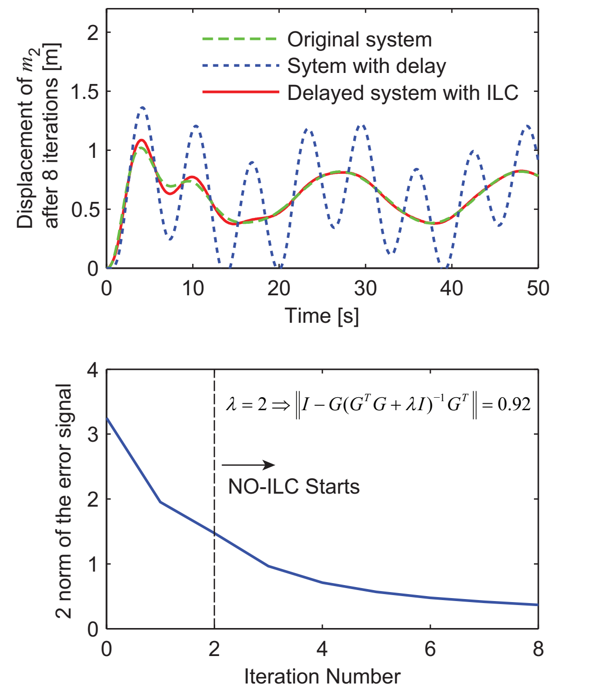

Fig. 5. Simulation result at the end of 8th iteration with $\tau = 0.2$ s (above) and the ILC performance (below)

reduces by 86.65% within 8 iterations. As a result of an improved coupling, the 2 norm of the simulation output error is reduced by 93.1%.

Results for several choices of $\lambda$ are summarized in Fig. 6. These results show that increasing $\lambda$ increases the monotonic convergence ratio (i.e.; slows the convergence), and reduces the coupling error reduction and simulation output error reduction percentages for a given number of iterations. This is expected to increase the robustness to modeling errors and non-repeated disturbances; however, investigation of robustness is beyond the scope of this paper and is left as future work. Extension to random delays and nonlinear systems are other important directions for future research.

## IV. CONCLUSION

A MIMO NO-ILC framework has been developed to improve simulation fidelity in LTI ID-HIL systems with constant network delays. A recursive formulation for Markov parameter estimation is introduced within NO-ILC.

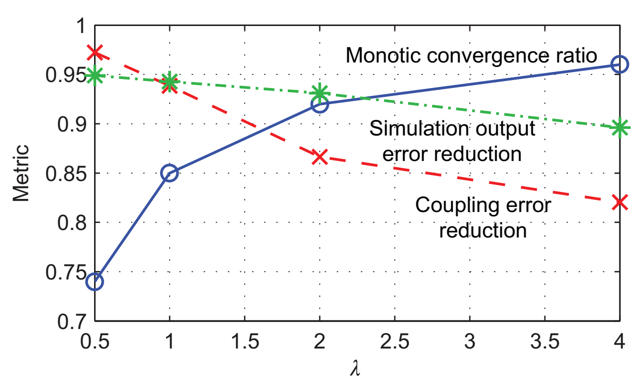

Fig. 6. Convergence and error metrics for $\tau = 0.2$ s and 8 iterations

The theory has been validated by a simulation-based example, showing that a monotonic improvement of fidelity can be achieved without a priori knowledge about the system dynamics. Results encourage further development for network-integrated nonlinear systems.

## V. References


[1] H. K. Fathy, Z. S. Filipi, J. Hagena, and J. L. Stein, "Review of hardware-in-the-loop simulation and its prospects in the automotive area," Modeling and Simulation for Military Applications, Kissimmee, FL, United states, vol. 6228, pp. SPIE, 2006.

[2] A. Kimura and I. Maeda, "Development of engine control system using real time simulator," Control Applications Intelligent Control and Computer Aided Control System Design, New York, NY, USA, pp. 157-63, 1996.

[3] J. Leitner, "A hardware-in-the-loop testbed for spacecraft formation flying applications," IEEE Aerospace Conference Proceedings, Piscataway, NJ, USA, vol. vol.2, pp. 2-615, 2001.

[4] Y. Xie, D. M. Vilathgamuwa, and T. King-Jet, "Robust adaptive control of a three-axis motion Simulator with state observers," IEEE/ASME Transactions on Mechatronics, vol. 10, no. 4, pp. 437-48, 2005.

[5] A. Ganguli, A. Deraemaeker, M. Horodinca, and A. Preumont, "Active damping of chatter in machine tools - demonstration with a 'hardware-in-the-loop' simulator," Journal of Systems and Control Engineering, vol. 219, no. I5, pp. 359-69, 2005.

[6] F. Aghili and J.-C. Piedboeuf, "Contact dynamics emulation for hardware-in-loop simulation of robots interacting with environment," 2002 IEEE International Conference on Robotics and Automation, Washington, D.C., vol. 1, pp. 523-529, 2002.

[7] G. D. White, R. M. Bhatt, C. P. Tang, and V. N. Krovi, "Experimental evaluation of dynamic redundancy resolution in a nonholonomic wheeled mobile manipulator," IEEE/ASME Transactions on Mechatronics, vol. 14, no. 3, pp. 349-57, 2009.

[8] S. Mahin, R. Nigbor, C. Pancake, R. Reitherman, and S. Wood, "The establishment of the NEES consortium," 2003 ASCE/SEI Structures Congress and Exposition: Engineering Smarter, pp. 181-182, 2003.

[9] B. Stojadinovic, G. Mosqueda, and S. A. Mahin, "Event-driven control system for geographically distributed hybrid simulation," Journal of Structural Engineering, vol. 132, no. 1, pp. 68-77, 2006.

[10] G. Mosqueda, B. Stojadinovic, J. Hanley, M. Sivaselvan, and A. M. Reinhorn, "Hybrid seismic response simulation on a geographically distributed bridge model," Journal of Structural Engineering, vol. 134, no. 4, pp. 535-543, 2008.

[11] P. Pan, M. Tada, and M. Nakashima, "Online hybrid test by internet linkage of distributed test-analysis domains," Earthquake Engineering and Structural Dynamics, vol. 34, no. 11, pp. 1407-1425, 2005.

[12] Y. Takahashi and G. L. Fenves, "Software framework for distributed experimental-computational simulation of structural systems," Earthquake Engineering and Structural Dynamics, vol. 35, no. 3, pp. 267-291, 2006.

[13] M. Compere, J. Goodell, M. Simon, W. Smith, and M. Brudnak, "Robust control techniques enabling duty cycle experiments utilizing a 6-dof crewstation motion base, a full scale combat hybrid electric power system, and long distance internet communications," Power Systems Conference, New Orleans, LA, United states, 2006.

[14] J. Goodell, M. Compere, M. Simon, W. Smith, R. Wright, and M. Brudnak, "Robust control techniques for state tracking in the presence of variable time delays," SAE World Congress, Detroit, MI, United states, 2006.

[15] T. Ersal, M. Brudnak, J. L. Stein, and H. K. Fathy, "Variation-based transparency analysis of an internet-distributed hardware-in-the-loop simulation platform for vehicle powertrain systems," ASME DSCC, Hollywood, CA, United states, pp. 1217-1224, 2010.

[16] T. Ersal, M. Brudnak, A. Salvi, J. L. Stein, Z. Filipi, and H. K. Fathy, "Development and model-based transparency analysis of an Internet-distributed hardware-in-the-loop simulation platform," Mechatronics, vol. 21, no. 1, pp. 22-9, 2011.

[17] T. Ersal, M. Brudnak, J. L. Stein, and H. K. Fathy, "Statistical transparency analysis in internet-distributed hardware-in-the-loop simulation," IEEE/ASME Transactions on Mechatronics, vol. 17, no. 2, pp. 228-38, 2012.

[18] T. Ersal, M. Brundnak, and J. L. Stein, "An Iterative Learning Control Approach To Improve Fidelity in Internet-Distributed Hardware-In-the-Loop Simulation," ASME Dynamic Systems and Control Conference, 2012.

[19] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," IEEE Transactions on Robotics and Automation, vol. 9, no. 5, pp. 624-37, 1993.

[20] K. Hashtrudi-Zaad and S. E. Salcudean, "Transparency in time-delayed systems and the effect of local force feedback for transparent teleoperation," IEEE Transactions on Robotics and Automation, vol. 18, no. 1, pp. 108-114, 2002.

[21] K. B. Fite, J. E. Speich, and M. Goldfarb, "Transparency and stability robustness in two-channel bilateral telemanipulation," Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control, vol. 123, no. 3, pp. 400-7, 2001.

[22] T. Ersal, R. B. Gillespie, M. J. Brudnak, J. L. Stein, and H. K. Fathy, "Effect of coupling point selection on distortion in internet-distributed hardware-in-The-loop simulation," International Journal of Vehicle Design, vol. 61, no. 1-4, pp. 67-85, 2013.

[23] Y. Yokokohji, T. Imaida, and T. Yoshikawa, "Bilateral teleoperation under time-varying communication delay," IEEE/RSJ International Conference, Kyongju, South Korea, vol. 3, pp. 1854-1859, 1999.

[24] P. G. Griffiths, R. B. Gillespie, and J. S. Freudenberg, "A fundamental tradeoff between performance and sensitivity within haptic rendering," IEEE Transactions on Robotics, vol. 24, no. 3, pp. 537-48, 2008.

[25] P. G. Griffiths, R. B. Gillespie, and J. S. Freudenberg, "A Fundamental Linear Systems Conflict Between Performance and Passivity in Haptic Rendering," IEEE Transactions on Robotics, vol. 27, no. 1, pp. 75-88, 2011.

[26] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control for discrete-time systems with exponential rate of convergence," IEE Proceedings-Control Theory and Applications, vol. 143, no. 2, pp. 217-24, 1996.

[27] K. W. C. a. L. J. H. Lee K.s., "Model-based Iterative learning Control with Quadratic Criterion for Linear Batch Process," Journal of Control, Automation and Systems Engineering, pp. 148, 1996.

[28] P. Janssens, G. Pipeleers, and J. Swevers, "A data-driven constrained norm-optimal iterative learning control framework for LTI systems," IEEE Transactions on Control Systems Technology, vol. 21, no. 2, pp. 546-51, 2013.

[29] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control: A learning-based method for high-performance tracking control," IEEE Control Systems Magazine, vol. 26, no. 3, pp. 96-114, 2006.

[30] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," International Journal of Control, vol. 73, no. 10, pp. 930-54, 2000.

[31] M. Norrlof, "An adaptive iterative learning control algorithm with experiments on an industrial robot," IEEE Transactions on Robotics and Automation, vol. 18, no. 2, pp. 245-51, 2002.

[32] J. H. Lee, K. S. Lee, and W. C. Kim, "Model-based iterative learning control with a quadratic criterion for time-varying linear systems," Automatica, vol. 36, no. 5, pp. 641-57, 2000.

[33] M. French and E. Rogers, "Nonlinear iterative learning by an adaptive Lyapunov technique," IEE Seminar Learning Systems for Control, London, UK, pp. 9-1, 2000.

[34] Y.-J. Pan, H. J. Marquez, and T. Chen, "Sampled-data iterative learning control for a class of nonlinear networked control systems," American Control Conference, Minneapolis, MN, United states, vol. 2006, pp. 3494-3499, 2006.

[35] J. van de Wijdeven, T. Donkers, and O. Bosgra, "Iterative learning control for uncertain systems: robust monotonic convergence analysis," Automatica, vol. 45, no. 10, pp. 2383-91, 2009.

[36] X. Jian-Xin, "A survey on iterative learning control for nonlinear systems," International Journal of Control, vol. 84, no. 7, pp. 1275-94, 2011.