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)
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 $c_j^1$ and $c_j^2$. 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]:

$$
e_i^1(t) = e_{i-1}^{1,2}(t) - e_{i-1}^{1,1}(t)$$

$$
e_i^2(t) = e_{i-1}^{2,2}(t) - e_{i-1}^{2,1}(t)$$

(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
\[
\begin{align*}
\dot{x}_i(t+1) &= Ax_i(t) + Bu_i(t) + B_w(t) \\
y_i(t) &= Cx_i(t) + Du_i(t)
\end{align*}
\]
(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 \(x_i(t) \in \mathbb{R}^n\), \(u_i(t) \in \mathbb{R}^m\), \(y_i(t) \in \mathbb{R}^p\), \(w_i(t) \in \mathbb{R}^q\), respectively, where \(n, m, p, q\) and \(I\) 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
\[
y_i(t) = Cx_i(t) + Du_i(t); \quad y_i(t) = Cx_i(t) + Du_i(t) + B_w(t); \quad y_i(t) = Cx_i(t) + Du_i(t) + B_w(t) + C B_w(t) + B_w(t); \quad \text{and can be given in matrix form as}
\]
\[
y_i = G u_i + G_w w_i + \left[ C \cdots C A T^{-1} \right]^T x_i(t), \quad (4)
\]
where \(G \in \mathbb{R}^{m \times nT}\) is the Markov matrix relating inputs \(u_i \in \mathbb{R}^m\) and outputs \(y_i \in \mathbb{R}^n\), and \(G_w \in \mathbb{R}^{m \times qT}\) is the matrix relating disturbances \(w_i \in \mathbb{R}^q\) and outputs, with
\[
y_i(t) = \begin{bmatrix} y_i(t) \\ y_i(t-1) \end{bmatrix}; u_i = \begin{bmatrix} u_i(t) \\ u_i(t-1) \end{bmatrix}; G = \begin{bmatrix} D & 0 & 0 & 0 \\ C B & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ C A T^{-2} B & \cdots & C B D \end{bmatrix}
\]
(5)
Let \(\Delta y_i = \begin{bmatrix} y_i(t) - y_i(t-1) \\ y_i(t-1) - y_i(t-2) \end{bmatrix}\) and \(\Delta u_i = \begin{bmatrix} u_i(t) - u_i(t-1) \\ u_i(t-1) - u_i(t-2) \end{bmatrix}\) 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
\[
\Delta y_i = G \Delta u_i. \quad (6)
\]
Let \(y_i(t) = \begin{bmatrix} y_i(t) \\ y_i(t-1) \end{bmatrix}\) denote the desired output. Then, the error \(e_i\) to be minimized by ILC can be defined as
\[
e_i = y_i - y_i = \begin{bmatrix} y_i(t) - y_i(0) \\ y_i(t-1) - y_i(t-1) \end{bmatrix}
\]
(7)
\[
e_i = y_i - y_i - \Delta y_i = e_i - G \Delta u_i.
\]
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]:
\[
J(u_i) = \left\| e_i \right\|_Q + \left\| u_i - u_{i-1} \right\|_R. \quad (8)
\]
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. \(J\) 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
\[
\frac{\partial J(u_i)}{\partial \Delta u_i} = \frac{\partial}{\partial \Delta u_i} \left(\left\| e_i \right\|_Q + \left\| u_i - u_{i-1} \right\|_R \right) = 0. \quad (9)
\]
The first term in (9) can be derived as following
\[
\frac{\partial}{\partial \Delta u_i} \left\| e_i \right\|_Q = \frac{\partial}{\partial \Delta u_i} \left(\left\| e_i \right\|_Q \right) \quad \text{and the second term can be rewritten as}
\]
\[
\frac{\partial}{\partial \Delta u_i} \left\| u_i - u_{i-1} \right\|_R = \frac{\partial}{\partial \Delta u_i} \left[ \left\| u_i - u_{i-1} \right\|_R \right] = \frac{\partial}{\partial \Delta u_i} \left(\left\| u_i - u_{i-1} \right\|_R \right)
\]
Combining (10) and (11), (9) leads to the following result
\[
u_i = u_i - (G^T Q G + R)^{-1} G^T Q e_{i-1}. \quad (12)
\]
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
\[
e_i = e_{i-1} - G \Delta u_i = [I - G^T Q G + R]^{-1} G^T Q e_{i-1}. \quad (13)
\]
Hence, asymptotic convergence can be achieved if the spectral radius of the matrix multiplier of \(e_{i-1}\) is less than 1; i.e.,
\[
\rho(I - G^T Q G + R)^{-1} G^T Q) < 1 \quad (14)
\]
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
\[
\left\| (I - G^T Q G + R)^{-1} G^T Q \right\|_\infty < 1 \quad (15)
\]
These results agree with the results reported in [27] for the SISO case.

Notice that choosing \(R = 0\) will make (14) and (15) zero. However, NO-ILC with \(R = 0\) 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 \(R\):
\[
R = \lambda I \quad (16)
\]
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

\[
\begin{align*}
\rho(I - G(G^T G + \lambda I)^{-1} G^T) &< 1 \\
\|I - G(G^T G + \lambda I)^{-1} G^T\| &< 1
\end{align*}
\]

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 defined as

\[
\text{errors equal weight. Choosing } R = \lambda Q \text{ simplifies (14) and (15) as follows }
\]

\[
\begin{align*}
\rho(I - G(G^T G + \lambda I)^{-1} G^T) &< 1 \\
\|I - G(G^T G + \lambda I)^{-1} G^T\| &< 1
\end{align*}
\]

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.

\[
\begin{align*}
\rho(I - G(G^T G + \lambda I)^{-1} G^T) &< 1 \\
\|I - G(G^T G + \lambda I)^{-1} G^T\| &< 1
\end{align*}
\]

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.
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 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 $T_s = 0.1$ s of the system; i.e., $\tau = 0.1, 0.2$ 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 $G = \text{diag}(-1)$, 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 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.
The theory has been valid 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


