Iterative Learning Control for Linear Hybrid System with Iteration-varying References

1st Byungjun Kim
Department of Mechanical Engineering (of KAIST.)
Daejeon, Republic of Korea
asx9608@kaist.ac.kr

2nd Seibum B.Choi*
Department of Mechanical Engineering (of KAIST.)
Daejeon, Republic of Korea
sbchoi@kaist.ac.kr

***Abstract*—Based on the advancements in iterative learning control (ILC), this paper proposes a novel approach tailored for linear hybrid systems with iteration-varying references. By extending previous research, the method accurately estimates all system modes, even if there are mode transitions with iteration-varying mode transition time, ensuring precise calculation of control inputs. The estimation begins by identifying the first mode and decoupling it from the next mode to facilitate accurate estimation. The mode estimator and control law were mathematically proven to converge with increasing iterations, a conclusion further supported by simulations validating their performance.**

***Index Terms*—iterative learning control, linear hybrid system, iteration-varying, mode transition**

I. Introduction

To enhance the performance of controllers, there are methods to utilize model information and provide feedback using sensor signals effectively. Another approach is to utilize past control information. Iterative learning control is a technique that performs well when conducting repetitive control tasks [1], [2]. There are various types of controllers that leverage past control information. However, unlike other learning techniques, such as machine learning and reinforcement learning, Iterative Learning Control (ILC) possesses a significant advantage in intuitively demonstrating mathematical convergence. Moreover, aiming for perfect tracking through repetitive control enhances the transient performance of the control system. Hence, it is extensively used in applications involving repetitive tasks, such as robots, automated processes, and chemical processes [3]–[5]. Nonetheless, it is essential to note that ILC exhibits good performance only under considerably restrictive conditions, which limits its effectiveness.

ILC is commonly applied in situations where the same task is performed repeatedly. Consequently, it often applies only under conditions with identical initial conditions, references, and repetitive disturbances. Numerous research efforts have been undertaken to mitigate these restrictive conditions. Research has proposed methods such as initial rectification to address the challenge of adapting to initial conditions that vary with each iteration [6], [7]. Various methods have been proposed to eliminate disturbances, such as employing observers to estimate and remove different disturbances encountered in each iteration [8], [9]. Research efforts have also been undertaken to apply ILC to changing references rather than fixed ones. The most common approach involves utilizing existing data to identify the parameters of the system model, enabling adaptation to changing references. Since this method involves modeling estimation, it allows for generating feedforward inputs for variously changing references, ensuring convergence as the number of trials increases. In the case of a linear system, it is possible to estimate the system model by estimating the system's Markov parameters, as Oh proposed [10]. For nonlinear systems cases, a model is established by iteratively estimating the model's parameters in the iteration domain using adaptive methods [11]–[13]. However, existing research addressing iteration-varying references has limitations. Previous studies have only addressed cases where the system remains unchanged or changes at the same time in each iteration. System changes often depend on the state in many situations, necessitating new research to address these scenarios.

Existing methods commonly encounter divergence in hybrid systems that undergo changes midway. Research on applying ILC to hybrid systems is scarce. KD Mishra proposed ILC for hybrid systems and applied it to vehicle gearshift control [14], [15]. However, it is limited to situations where all references remain unchanged. Research on classifying systems in linear hybrid systems is a significant field, utilizing system inputs and outputs for classification [16], [17]. Apart from estimator design and learning, various methods exist for this purpose. Mainly, considerable research is done on identifying switched linear systems, often involving generating residuals and using them to estimate modes through mode estimators [18], [19]. However, for mode estimation, this method typically requires a system model of the system's modes, posing a challenge for applications like ILC, where the system model is primarily unknown.

This paper proposes an Iterative Learning Control (ILC) approach for linear hybrid systems that can converge even when the reference changes. Building upon Oh's ILC method for linear systems with changing references [10], this paper aim

This research was partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2024-00346702); the BK21+ program through the NRF funded by the Ministry of Education of Korea.

to ensure convergence even when the system changes during the control period. Unlike previous research, which assumed the system changes simultaneously or does not change in each iteration, the proposed method addresses situations where the system changes dependently on its state and output, which is typical in hybrid systems. Based on input and output data, estimate the system of the first mode and then estimate the subsequent mode's system within the iteration domain based on the estimated first mode system.

Bayesian approach for model identification, the focus is on modeling uncertainty. This method estimates the posterior probability distribution of system parameters and model structure based on observed data, but this process requires numerical techniques and significant computational effort [21], [22]. In contrast, the proposed method optimizes the model parameters through an iterative learning process. Leveraging iterative data aims to estimate a more accurate model with the advantage of faster convergence. In conclusion, the proposed method is advantageous for quickly improving performance in iterative environments compared to the Bayesian approach.

Ultimately, this paper demonstrates the convergence of errors as the number of control iterations increases, confirming this effect through examples. While the proposed method applies to hybrid systems with two modes, it can be extended to hybrid systems with multiple modes.

II. Problem formulation

Consider the discrete-time linear hybrid system as follows:

$$\begin{aligned} x_k(t+1) &= A_i x_k(t) + B_i u_k(t) \\ y_k(t) &= C_i x_k(t), \end{aligned} \tag{1}$$

where t is the discrete time steps, k is the iteration number, and $i = 1...p$ is the number of system mode. $x \in \mathbb{R}^n$ is the system state variables. $u \in \mathbb{R}^m$ is the system input. $y \in \mathbb{R}^q$ is the system output.

Assumption 1: The system mode is dependent on the system output $Y_{r,i-1} < y_k(t) <= Y_{r,i}$. If the system output $y_k(t)$ lies between the residual of the $(i-1)th$ mode, denoted as $Y_{r,i-1}$ and the residual of the ith mode, denoted as $Y_{r,i}$, then the system is in the ith mode. It is assumed that the reference values determining the system modes, $Y_{r,i-1}$, are known in advance.

Therefore, this chapter introduces a method for estimating the system matrix of a hybrid system. Then use this estimation to propose a method for designing control inputs in the iteration domain to converge the error to zero.

If control is executed for the same duration in each iteration starting from the initial condition $x_k(0) = 0$, the hybrid system described above can be represented as the following lifted system as

$$Y_k = G_k U_k. \tag{2}$$

where Y_k, and U_k are

$$\begin{aligned} Y_k &= \begin{bmatrix} y_k^T(1) & y_k^T(2) & \cdots & y_k^T(N) \end{bmatrix}^T \\ U_k &= \begin{bmatrix} u_k^T(0) & u_k^T(1) & \cdots & u_k^T(N-1) \end{bmatrix}^T, \end{aligned}$$

and system matrix $G^k \in \mathbb{R}^{qN \times mN}$ is represented as

$$G_k = \begin{bmatrix} G_k^1 & 0 \\ G_k^{12} & G_k^2 \end{bmatrix}$$

$$G_k^1 = \begin{bmatrix} C_1B_1 & 0 & 0 & 0 \\ C_1A_1B_1 & C_1B_1 & 0 & 0 \\ C_1A_1^2B_1 & C_1A_1B_1 & C_1B_1 & 0 \\ \vdots & \cdots & \ddots & \ddots & \vdots \\ C_1A_1^{N_1^j-1}B_1 & C_1A_1^{N_1^j-2}B_1 & \cdots & C_1B_1 \end{bmatrix}$$

$$G_k^2 = \begin{bmatrix} C_2B_2 & 0 & 0 & 0 \\ C_2A_2B_2 & C_2B_2 & 0 & 0 \\ C_2A_2^2B_2 & C_2A_2B_2 & C_2B_2 & 0 \\ \vdots & \cdots & \ddots & \vdots \\ C_2A_2^{N-N_1^j-1}B_2 & C_1A_1^{N-N_1^j}B_2 & \cdots & C_2B_2 \end{bmatrix}$$

$$G_k^{12} = \begin{bmatrix} C_2A_2A_1^{s_1}B_1 & C_2A_2A_1^{s_2}B_1 & \cdots & C_2A_2B_1 \\ C_2A_2^2A_1^{s_1}B_1 & C_2A_2^2A_1^{s_2}B_1 & \cdots & C_2A_2^2B_1 \\ \vdots & \vdots & \ddots & \vdots \\ C_2A_2^{N-N_1^j}A_1^{s_1}B_1 & C_2A_2^{N-N_1^j}A_1^{s_2}B_1 & \cdots & C_2B_1 \end{bmatrix}, \tag{3}$$

where $s_p = N_1^j - p$. Also in the paper, although there is the possibility of expanding the system's modes, for simplicity, only the case with two modes($i = 1, 2$) is discussed.

The objective of the control input $u_k(t)$ at times $t : [0, 1, ..., T-1]$ is to maintain uniform boundedness across all iterations and to accurately follow the iteration-varying reference $r_k(t)$ at times $t : [0, 1, ..., T]$ as k goes infinity.

If the system is a linear system and not a hybrid system, G_k consists only of G_k^1 without G_K^2 and G_k^{12}. Then according to Oh's proposed method, updating the control input as (4) results in the tracking error converging to zero as the iteration number increases.

$$\begin{aligned} U_{k+1} &= U_k + H_k(R_{k+1} - Y_k) \\ H_k &= \hat{G}_k^{-1}, \end{aligned} \tag{4}$$

where $R_k = \begin{bmatrix} r_k^T(1) & r_k^T(2) & \cdots & r_k^T(N) \end{bmatrix}^T$. The matrix gain H_k used in (4) is based on the inverse of the plant, denoted as G_k^{-1}. It calculates the required input for the next iteration by inversely computing the input needed between the r_{k+1} and the y_k. With Oh's proposed method alongside (4), it is possible to estimate g_1 and calculate G_k in the iteration domain using (5) [10]. g_1 is the reverse of the last row of $G_{1,k}$, and by estimating it, $G_{1,k}$ can be calculated. For further details and convergence analysis regarding this, please refer to Oh's method.

$$\begin{aligned} g_{1,k} &= g_{1,k-1} + (\bar{Y}_{1,k} - g_{1,k-1}\bar{U}_{1,k})P_{1,k} \\ P_{1,k} &= \bar{U}_{1,k}^{-1} \end{aligned} \tag{5}$$

where $\bar{Y}_{1,k}$, g_1, and $\bar{U}_{1,k}$ are

$$
\begin{aligned}
g_1 &= \begin{bmatrix} C_1B_1 & C_1A_1B_1\cdots & C_1A_1^{N^j-1}B_1 \end{bmatrix} \\
\bar{Y}_{1,k} &= \begin{bmatrix} y_k(1) & y_k(2) & \cdots & y_k(N^j) \end{bmatrix} \\
\bar{U}_{1,k} &= \begin{bmatrix} u_k(0) & u_k(1) & u_k(2) & \cdots & u_k(N^j-1) \\ 0 & u_k(0) & u_k(1) & \cdots & u_k(N^j-2) \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & u_k(0) \end{bmatrix}.
\end{aligned}
$$

This method cannot directly apply to the plant in (1). In the case of a linear system that is not a hybrid system, g can be estimated in the iteration domain based on (5) because the g remains constant in the iteration domain. However, in a hybrid system, the timing of mode transitions changes when the reference changes. As a result, the g changes with each mode transition, making it impossible to estimate g in the same manner as before. Since the length of mode 1 varies with each iteration due to the changing reference, G_k^2 and G_k^{12} also changes accordingly. Therefore, updates cannot be performed because the number of times A_1 and A_2 contained in g_2 vary each iteration.

III. Estimation and Control design

This section introduces methods for estimating system matrix G_k in linear hybrid systems, as well as control design and convergence analysis. The proposed method can be briefly examined through a schematic diagram 1. Existing research methods are utilized in estimation. Specifically, the estimation process employs techniques mentioned in the previous section, such as Oh's estimation method and the Eigenvalue Realization Algorithm (ERA). The estimation process involves estimating the system of the first mode and using it to determine the A_1, B_1, and C_1 matrices of system mode 1 in iterative learning identification 1. These matrices are then utilized to identify the second mode of the system using the proposed method in iterative learning identification 2, then ultimately resulting in accurately estimating G_k for control purposes.

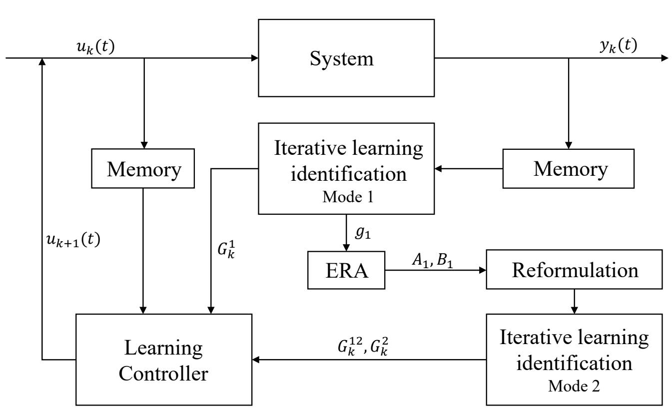


Fig. 1: Schematic of ILC for linear hybrid system

Since Iterative Learning Identification 1 and the Eigenvalue Realization Algorithm (ERA) utilize existing research methods, a brief mention of them will suffice. This section introduces the methods for estimating the second mode system and discusses convergence analysis.

A. Identification Mode 1 and ERA

For System mode 1, using existing methods allows us to estimate the Markov parameter g_1. Since g_1 does not change in the iteration domain, using (5) enables us to estimate its actual value. Since the timing of system mode changes varies with each iteration, the size of g_1 changes with each iteration, and updates are conducted according to its size. Subsequently, employing the Eigenvalue Realization Algorithm (ERA) algorithm on the estimated Markov parameter allows us to obtain the system matrices A and B in (1). However, in this study, the requirement is not for individual A_1 and B_1 matrices but for the controllability matrix R_c. Using singular value decomposition as shown in (6), R_c can be obtained from $H(0)$.

$$
\begin{aligned}
H(0) &= \begin{bmatrix} CB & CAB & \cdots & CA^{N_1^j/2}B \\ CAB & CA^2B & \cdots & CA^{N_1^j/2+1}B \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N_1^j/2}B & CA^{N_1^j/2+1}B & \cdots & CA^{N_1^j}B \end{bmatrix} \\
&= PDQ^T \simeq P_nD_nQ_n^T \\
R_c &= \begin{bmatrix} B & AB & A^2B & \cdots & A^{N_1^j/2-1}B \end{bmatrix} = D_n^{1/2}Q_n^T
\end{aligned} \tag{6}
$$

where n is the degree of the system. It is important to note that the length of mode 1 varies with each iteration, thus the value of N_1^j also changes accordingly in each iteration. As the size of $H(0)$ increases, R_c obtains a more accurate value. Therefore, R_c is not updated every iteration; instead, it is updated when the length of mode 1 exceeds that of the previous lengths.

B. Identification Mode 2

As shown in (2) and (3), the output of system mode 2 is influenced not only by the system's Mode 2 matrices, A_2 and B_2, but also by the dynamics of system mode 1 and the moments of mode transition. Therefore, direct estimation is not feasible, and decoupling is required to remove the influence of the remaining parts except for system mode 2.

The output for mode 2 can be expressed as follows, based on (2) and (3).

$$
\begin{aligned}
y_k(N^j+m) = &\sum_{s=1}^{N^j} C_2A_2^mA_1^{N^j-s}B_1u_k(s-1) \\
&+ \sum_{s=1}^{m} C_2A_2^{s-1}B_2u_k(N^j+s-1).
\end{aligned} \tag{7}
$$

For simplicity, denote $u_T = \sum_{s=1}^{N_j} A_1^{N^j-s}B_1u_k(s-1)$. u_T is composed entirely of known values, and can be calculated using R_c as

$$\begin{aligned} u_T &= R_c\tilde{U}_k = D_n^{1/2}Q_n^T\tilde{U}_k \\ \tilde{U}_k &= \begin{bmatrix} u_k(N^j-1) & u_k(N^j-2) & \cdots & u_k(0) \end{bmatrix}^T. \end{aligned} \tag{8}$$

Then, using (7), $Y_{2,k}$ can be rewritten as

$$\begin{aligned} y_k(N_j+m) &= C_2A_2^m u_T + \sum_{s=1}^{m} C_2A_2^s B_2 u_k(N^j+s-1) \\ \bar{Y}_{2,k} &= g_2\bar{U}_{2,k}, \end{aligned} \tag{9}$$

where $\bar{Y}_{2,k}$, g_2, and $\bar{U}_{2,k}$ are

$$\begin{aligned} \bar{Y}_{2,k} &= \begin{bmatrix} y_k(N^j+1) & y_k(N^j+2) & \cdots & y_k(N) \end{bmatrix}^T \\ g_2 &= \begin{bmatrix} C_2B_2 & C_2A_2 & C_2A_2B_2 & C_2A_2^2 & C_2A_2^2B_2 & \cdots \end{bmatrix} \end{aligned}$$

$$\bar{U}_{2,k} = \begin{bmatrix} u_k(N^j+1) & u_k(N^j+2) & \cdots & u_k(N) \\ u_T & 0 & \cdots & 0 \\ 0 & u_k(N^j+1) & \cdots & u_k(N-1) \\ 0 & u_T & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & u_T \end{bmatrix}.$$

The zeros adjacent to u_T in the matrix $\bar{U}_{2,k}$ represent zero matrices adjusted to the size of u_T. By rearranging the output $y_k(N^j+m)$ as in (9), it is possible to separate the g_2 that influences only system mode 2 using u_T. Since u_T can be computed via R_c as shown in (8), it can be separated into known and unknown parts of $\bar{Y}_{2,k}$. Like g_1, g_2 remains constant in the iteration domain, allowing the design of estimation techniques for convergence in the iteration domain as (10).

$$\begin{aligned} g_{2,k+1} &= g_{2,k} + (\bar{Y}_{2,k} - g_{2,k-1}\bar{U}_{2,k})P_{2,k} \\ P_{2,k} &= \bar{U}_{2,k}^{-1} \end{aligned} \tag{10}$$

Similarly to the approach used for system mode 1, by setting $P_{2,k} = \bar{U}_{2,k}^{-1}$, it can be confirmed that $\|g_2 - g_{2,k}\|$ converges to zero as k increases, as depicted in (11).

$$\begin{aligned} &\|g_2 - g_{2,k+1}\|\|U_{2,k+1}\| \\ &\qquad \le \|g_2 - g_{2,k}\|\|I - U_{2,k}P_{2,k}\|\|U_{2,k+1}\|. \end{aligned} \tag{11}$$

However, unlike system mode 1, the presence of zero matrices in $\bar{U}_{2,k}$ indicates that it is not full rank. Consequently, even if $\bar{U}_{2,k}^{-1}$ is multiplied to $\bar{Y}_{2,k}$, an accurate determination of g_2 is impossible. Therefore, data accumulation in the iteration domain and collective updates are implemented. Iterations are combined in intervals equivalent to the sum of the matrix size of C_2B_2 and C_2A_2. For instance, if C_2B_2 is 1×1 and C_2A_2 is 1×2, updates occur once every third iteration (i.e., at iterations $1, 4, 7, \cdots$). In conclusion (10) can be rewritten as follows.

$$\begin{aligned} g_{2,z(k+1)+1} &= g_{2,zk+1} + (\bar{Y}_{zk} - g_{2,k-1}\bar{U}_{zk})P_{zk} \\ P_{zk} &= \bar{U}_{zk}^{-1}, \end{aligned} \tag{12}$$

where z is the sum of the matrix size of C_2B_2 and C_2A_2. The $\bar{Y}_{zk}$ and $\bar{U}_{zk}$ are

$$\begin{aligned} \bar{Y}_{zk} &= \begin{bmatrix} \bar{Y}_k & \bar{Y}_{k+1} & \cdots & \bar{Y}_{k+z-1} \end{bmatrix}^T \\ \bar{U}_{zk} &= \begin{bmatrix} \bar{U}_k & \bar{U}_{k+1} & \cdots & \bar{U}_{k+z-1} \end{bmatrix}. \end{aligned}$$

C. Control design and convergence analysis

Using the obtained plant information, we can design an ILC logic that converges to zero error as the iteration number k increases, as follows:

$$\begin{aligned} U_{k+1} &= H_{k+1}^{-1}H_kU_k + H_{k+1}^{-1}(R_{k+1} - Y_k) \\ H_{k+1} &= \hat{G}_{k+1} \end{aligned} \tag{13}$$

where G_k and G_{k+1} can be constructed using the estimated $g_{1,k}$ and $g_{2,k}$ obtained previous section. In estimating G_{k+1}, considering the system matrices (A_1, B_1, C_1) and (A_2, B_2, C_2) is crucial, but it is equally important to anticipate the timing of mode changes. The mode changing time can be determined using (13), where H_k replaces H_{k+1}^{-1}, and estimated y obtained by multiplying U with $\hat{G}_k$ is utilized to compute the timing. This approach is feasible because mode transitions are determined by output, implying that they are solely affected by the system of the previous mode and inputs up to the transition. Therefore, whether using $k+1$ or k makes little difference.

Theorem 1: When the input law specified in (13) for the hybrid linear system with evolving reference is utilized, the tracking error converges along the iteration axis.

Proof 1: The tracking error$(R_{k+1} - Y_{k+1})$ can be expressed as

$$R_{k+1} - Y_{k+1} = R_{k+1} - G_{k+1}U_{k+1}. \tag{14}$$

Let $G_k = H_k + \epsilon_{1,k}$, $G_{k+1}H_{k+1}^{-1} = I + \epsilon_{2,k}$ where $\epsilon_{1,k}, \epsilon_{2,k}$ are the estimation error. Using (13), (14) can be rewritten as

$$\begin{aligned} &R_{k+1} - Y_{k+1} \\ &\quad = R_{k+1} - G_{k+1}(H_{k+1}^{-1}H_kU_k + H_{k+1}^{-1}(R_{k+1} - Y_k)) \\ &\quad = R_{k+1} - Y_k - G_{k+1}H_{k+1}^{-1}(R_{k+1} - Y_k) \\ &\qquad\quad + (\epsilon_{1,k} + \epsilon_{2,k}H_k)U_k. \end{aligned} \tag{15}$$

By adding and subtracting $R_k + G_{k+1}H_{k+1}^{-1}R_k$ to (16) and let $(\epsilon_{1,k} + \epsilon_{2,k}H_k)U_k = \delta_k$, $R_{k+1} - R_k = \Delta R_{k+1}$. Then it can be simplified as

$$\begin{aligned} &e_{k+1} \\ &= (I - G_{k+1}H_{k+1}^{-1})e_k + (I - G_{k+1}H_{k+1}^{-1})\Delta R_{k+1} + \delta_k \\ &= \prod_{i=1}^{k}(I - G_{i+1}H_{i+1}^{-1})e_1 + \sum_{j=1}^{k}\prod_{i=1}^{k+1-j}(I - G_{i+1}H_{i+1}^{-1})\Delta R_{j+1} \\ &\quad + \sum_{j=1}^{k-1}\prod_{i=1}^{k-j}(I - G_{i+1}H_{i+1}^{-1})\delta_j + \delta_k. \end{aligned} \tag{16}$$

As k increases, based on (5) and (12), $g_{1,k}$ and $g_{2,k}$ respectively converge to g_1 and g_2, allowing the accurate determination of G_k. Since $H_k = \hat{G}_k$, $||I - G_{i+1}H_{i+1}^{-1}|| < 1$ and is almost close to zero. Additionally, as estimation error δ_k converges to zero as k increases, it is confirmed that the tracking error eventually converges to zero.

IV. NUMERICAL EXAMPLES

In this section, the proposed method's effectiveness is validated using the linear hybrid system previously employed in existing studies [20].

The mass-spring-damper system, composed of two modes, can be schematically illustrated as depicted in Fig.2.

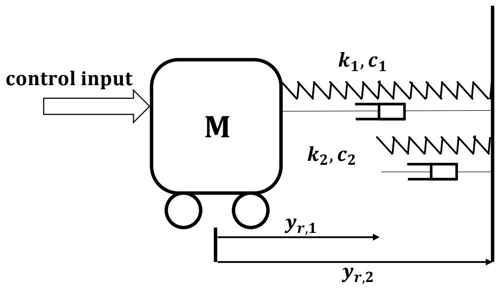


Fig. 2: The Schematic diagram of the linear hybrid system

M denotes the cart's mass, spring constant is k and damping coefficient is c. The system's input u represents the force exerted on the cart, while the output y is the distance from the starting point. $y_{r,1}$ marks the point of mode transition in the system. For the continuous system, the system matrix is

$$A_1 = \begin{bmatrix} 0 & 1 \\ -\frac{k_1}{M} & -\frac{c_1}{M} \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 1 \\ -\frac{k_1+k_2}{M} & -\frac{c_1+c_2}{M} \end{bmatrix},$$
$$B = \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \end{bmatrix},$$

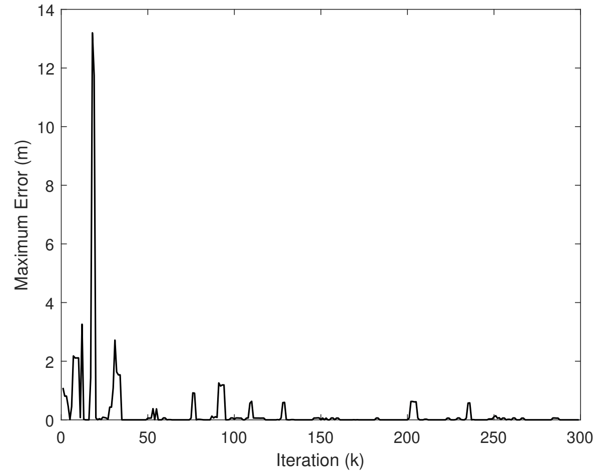
where the model parameters are set to $M = 1kg$, $k_1 = 100Nm$, $k_2 = 50Nm$, $c_1 = 0.7Nm/s$, $c_2 = 0.3Nm/s$. Two simulations were conducted, with the reference changing for each iteration. The criterion for mode transition in the output $y_{r,1}$, is set to 0.6. Additionally, the reference is not set to revert to the previous mode.

In the first simulation, the shape of the reference remained consistent, while only the magnitude varied from each iteration. For the second simulation, both the frequency and magnitude of the reference were set to vary with each iteration.

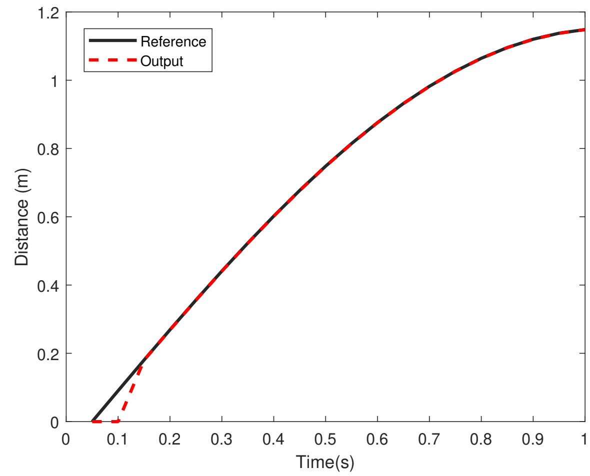
$$r_1(t,k) = p_1(k)\sin(0.01\pi(t-1))$$
$$r_2(t,k) = p_1(k)\sin(0.01\pi p_2(k)(t-1)),$$

where $p_1(k)$ ranges from 1 to 1.5, and $p_2(k)$ ranges from 1 to 1.7, both being rational numbers. The sum of the matrix sizes of C_2B_2 and C_2A_2 is 3, so updates were performed every third iteration.

Fig. 3 demonstrate the convergence of reference 1. As the value of k increases, it is evident that the tracking error converges well to 0. At the $300th$ iteration, the output precisely follows the reference. However, the system fails to track the reference in the initial step due to its second-order relative degree. As the number of iterations progresses, there are instances where the error converges to 0 and then reappears. The primary reason for this is when mode transitions occur outside the range previously learned, necessitating new learning and resulting in errors. The learned data determine the lengths of g_1 and g_2. If mode transitions occur at points not previously encountered, either shorter or longer, it requires updating g_1 and g_2 accordingly. The simulation results show that the model maintains a smooth output despite midway changes. This is due to the influence of the sampling time; the actual simulation result is not continuous but consists of discrete values plotted. At the point where the model changes, the results between sampling times may exhibit abrupt variations rather than smooth transitions. To address this, the system should be controlled with a smaller sampling time.



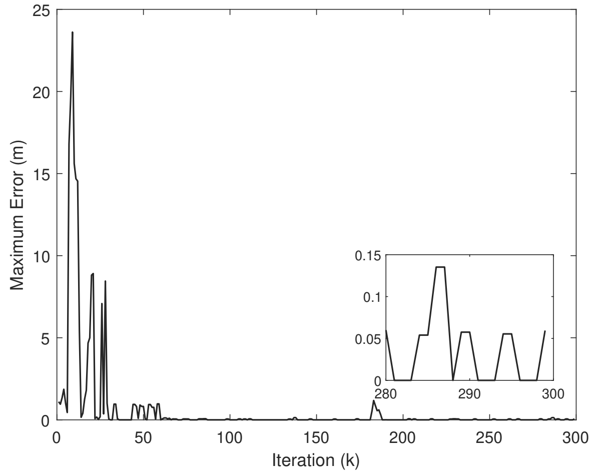
(a) Maximum error along the iteration axis (reference 1)



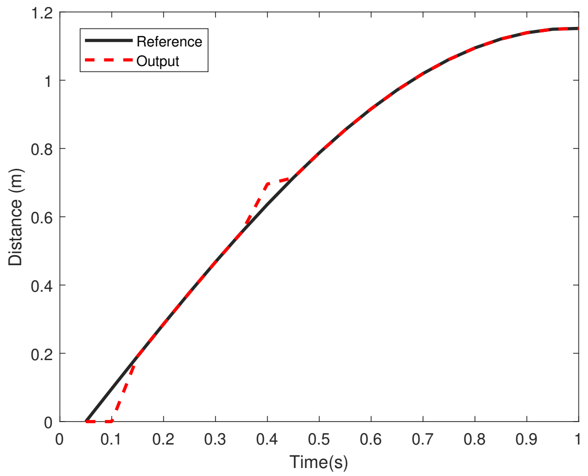
(b) Reference and output at 300th iteration (reference 1)

Fig. 3: Simulation results for reference 1

Fig. 4 depicts the results for reference 2. Similar to reference 1, it can be observed that the error converges to zero as the number of iterations increases. In Fig. 4b This can also be observed in Fig. 3a, where occasional minor errors (0.05) occur instead of perfect convergence to zero. This discrepancy arises from errors in predicting when the model will change, despite the model converging accurately.



(a) Maximum error along the iteration axis (reference 2)



(b) Reference and output at 300th iteration (reference 2)

Fig. 4: Simulation results for reference 2

V. Conclusions and Future Works

A. Conclusions

This paper proposes an Iterative Learning Control (ILC) logic capable of converging for Linear Hybrid Systems with iteration-varying references. Advancing previous research, The proposed method estimates the first mode and then decouples it from the subsequent modes using the estimated system mode 1, enabling accurate estimation of the following mode. The convergence of both the mode estimator and the iterative learning control law is proven. Through simulations, the effectiveness of proposed method is verified.

B. Future Works

This study has the advantage of applying linear hybrid systems, but many areas still need improvement. Firstly, it is essential to predict model transitions' timing accurately. Although the proposed method resulted in minor errors when precise prediction was not achieved, there is room for improvement to ensure convergence to zero error. Secondly, further advancement is needed to develop a logic that can converge even when the criteria(output) for model transitions are unknown.

References

[1] Bristow, D. A., Tharayil, M., Alleyne, A. G. ”A survey of iterative learning control.” IEEE control systems magazine, 26(3), 2012, pp 96-114.

[2] Moore, Kevin L. ”Iterative learning control for deterministic systems.”, 2012.

[3] Bondi, Paola, Giuseppe Casalino, and Lucia Gambardella. ”On the iterative learning control theory for robotic manipulators.” IEEE Journal on Robotics and Automation 4.1, 1988, pp 14-22.

[4] Ratcliffe, James D., et al. ”Norm-optimal iterative learning control applied to gantry robots for automation applications.” IEEE Transactions on Robotics 22.6, 2006, pp 1303-1307.

[5] Mezghani, Mouhiba, et al. ”Application of iterative learning control to an exothermic semibatch chemical reactor.” IEEE Transactions on Control Systems Technology 10.6, 2002, pp 822-834.

[6] Xu, Jian-Xin, Rui Yan, and YangQuan Chen. ”On initial conditions in iterative learning control.” 2006 American Control Conference. IEEE, 2006.

[7] Sun, Mingxuan, and Danwei Wang. ”Iterative learning control with initial rectifying action.” Automatica 38.7, 2002, pp 1177-1182.

[8] Chen, YangQuan, and Kevin L. Moore. ”Harnessing the nonrepetitiveness in iterative learning control.” Proceedings of the 41st IEEE Conference on Decision and Control, 2002.. Vol. 3. IEEE, 2002.

[9] Yu, Shuwen, and Masayoshi Tomizuka. ”Performance enhancement of iterative learning control system using disturbance observer.” 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. IEEE, 2009, pp 987-992

[10] Oh, Se-Kyu, and Jong Min Lee. ”Stochastic iterative learning control for discrete linear time-invariant system with batch-varying reference trajectories.” Journal of Process Control 36 ,2015, pp 64-78.

[11] Li, Xiao-Dong, Tommy WS Chow, and Lee Lung Cheng. ”Adaptive iterative learning control of non-linear MIMO continuous systems with iteration-varying initial error and reference trajectory.” International Journal of Systems Science 44, 2013, pp 786-794.

[12] Xu, Jian-Xin, and Jing Xu. ”On iterative learning from different tracking tasks in the presence of time-varying uncertainties.” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34.1, 2004, pp 589-597.

[13] Li, Xuefang, Dong Shen, and Jian-Xin Xu. ”Adaptive iterative learning control for MIMO nonlinear systems performing iteration-varying tasks.” Journal of the Franklin Institute 356.16, 2019, pp 9206-9231.

[14] Mishra, Kirti D., and K. Srinivasan. ”Iterative learning control for hybrid systems.” Dynamic Systems and Control Conference. Vol. 84270. American Society of Mechanical Engineers, 2020.

[15] Mishra, Kirti D., Guy Cardwell, and Krishnaswamy Srinivasan. ”Automated calibration of gearshift controllers using iterative learning control for hybrid systems.” Control Engineering Practice 111 (2021): 104786.

[16] Paoletti, Simone, et al. ”Identification of hybrid systems a tutorial.” European journal of control 13.2-3 (2007): 242-260.

[17] Pillonetto, Gianluigi. ”A new kernel-based approach to hybrid system identification.” Automatica 70 , 2016, pp 21-31.

[18] Vidal, Rene, et al. ”An algebraic geometric approach to the identification of a class of linear hybrid systems.” 42nd IEEE international conference on decision and control (IEEE Cat. No. 03CH37475). Vol. 1. IEEE, 2003.

[19] Domlan, Elom Ayih, José Ragot, and Didier Maquin. ”Active mode estimation for switching systems.” 2007 American Control Conference. IEEE, 2007.

[20] Mishra, Kirti Deo. ”Robust Iterative Learning Control for Linear and Hybrid Systems with Applications to Automotive Control.” The Ohio State University, 2020.

[21] Peterka. ”Bayesian approach to system identification” Trends and Progress in System identification. Pergamon, 1981.239-304.

[22] Juloski, Aleksandar Lj, Siep Weiland, and WP Maurice H. Heemels. ”A Bayesian approach to identification of hybrid systems.” IEEE Transactions on Automatic Control 50.10 (2005): 1520-1533.