Iterative Learning Control of Perspective Dynamic Systems

Lili Ma
Dept. of Aerospace and Ocean Engineering, Virginia Tech
215 Randolph Hall, Blacksburg, VA 24061, USA.

Kevin L. Moore *Senior Member, IEEE*
Division of Engineering, Colorado School of Mines
Brown Building 305, 1610 Illinois Street, Golden, CO 80401, USA.

YangQuan Chen *Senior Member, IEEE*
Center for Self-Organizing and Intelligent Systems (CSOIS), Dept. of ECE
4160 Old Main Hill, Utah State University, Logan, UT 84322-4160, USA
Emails: lma05@vt.edu, kmoore@mines.edu, yqchen@engineering.usu.edu

***Abstract*— This paper introduces the problem of iterative learning control (ILC) for perspective dynamic systems (PDS), referred to as ILC-PDS, where the task is to track a 3-D desired trajectory from information observed on the image plane of an imaging system, such as a camera. Unlike many ILC implementations in motion control applications, which use feedback from encoders, we assume measurements from the image plane are used as feedback to the ILC system. We focus our analysis on 2-D motion with a 1-D perspective measurement. It is shown analytically and via simulation that under the standard ILC assumption of identical initial conditions from trial-to-trial, tracking in the 2-D space can be achieved using the 1-D PDS image feedback.**

Key Words: Perspective dynamic systems, iterative learning control, perspective ILC.

I. Introduction

Iterative learning control, or ILC, is a technique for improving the transient response and tracking performance of processes, machines, equipment, or systems that execute the same trajectory, motion, or operation over and over [1]. In motion control applications most ILC implementations use encoder readings as feedback information. In this paper, we consider the problem where some type of imaging system is used to collect information about the plant. For our purposes we suppose the imaging information is visual information from a static camera, though we can generalize these ideas. Thus we are interested in controlling the motion of an iterative process where data from the image plane of the imaging system, i.e., the camera, is used as the feedback information for the ILC system. Consider, for example, an object that is moving along a 3-D trajectory in 3-D space whose motion is observed via a camera-type vision system. The data collected in the image plane of the camera is 2-D data. The transformation from 3-D to 2-D through the lens of a camera is called a perspective projection and the resulting dynamics in the image plane are called a perspective dynamic system. Then, if the motion is iterative our problem can be characterized as an iterative learning control (ILC) problem for a perspective dynamic system (PDS), which we refer to as the ILC-PDS problem. In this paper we formulate this problem and then give preliminary results for the case of 2-D motion with a 1-D perspective measurement. The paper is organized as follows. Introductions to the ILC and PDS problems are given in the remainder of this introductory section. In Section II, we give a general problem formulation for the ILC-PDS problem. Section III focuses on the 2-D problem by giving preliminary simulation results that show ILC will work for the PDS case if the standard initial condition reset assumption of ILC is satisfied. Section IV presents an analytical proof of this statement for the 2-D case. Section V concludes the paper.

A. Perspective Dynamic System (PDS)

A PDS is described by [2]:

$$\dot{x} = Ax + Bu, \quad y = [Cx], \tag{1}$$

where the projective observation function is defined as

$$\begin{aligned} Y : \mathbb{R}^n - \mathcal{B} &\longmapsto \mathbb{R}\,\mathbb{P}^{m-1}, \\ x &\longmapsto [Cx], \end{aligned} \tag{2}$$

where $[Cx]$ is the homogeneous line spanned by the nonzero vector $Cx \in \mathbb{R}^m$. The set $\mathcal{B}$ is defined as

$$\mathcal{B} = \{x : Cx = 0\}. \tag{3}$$

As seen from (1) and (2), a PDS is a linear dynamic system with a homogeneous observation function.

An example of a PDS when the object is moving according to an affine motion can be described as:

$$\begin{bmatrix} \dot{X}(t) \\ \dot{Y}(t) \\ \dot{Z}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}}_{A} \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix} + \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}}_{\mathbf{b}} u(t), \tag{4}$$

with $u(t) = 1, C = I_{3\times 3}$, and the outputs $y(t)$ defined to be

$$y(t) = [y_1, y_2]^T = [X(t)/Z(t), Y(t)/Z(t)]^T, \qquad (5)$$

Here $[X(t), Y(t), Z(t)]^T$ denotes the 3-D position of the moving object in the camera centered 3-D space and $(y_1(t), y_2(t))$ is its projection in the camera frame, which can be derived from the corresponding observations on the image plane.

The PDS has been found to be a good control theoretic framework for motion estimation problems where CCD cameras are used as sensors. Basic issues in PDS include the observability, identifiability, and controllability problems [2], [3]:

1) Observability: Assuming that A is known, estimate the initial state $X_0 = [X(0), Y(0), Z(0)]^T$ up to a homogeneous line from the output homogeneous observation function $y(t)$.
2) Identifiability: Assuming that A is unknown and the output $y(t)$ for $t > 0$ is given, identify A to the extent possible with X_0 up to a homogeneous line.
3) Controllability: Transmit from one state to another in the apparent motion on the image plane.

Using the PDS formulation, nonlinear observers applicable to the PDS [4], [5] have been used to estimate the states of a PDS, uniquely or to the extent possible, for simple cases when the object is undergoing some special motions, such as the rigid and pure rotational motions. As noted in [2], [3], perspective dynamic system theory not only reveals some existing knowledge in the computer vision, but also provides a theoretical analysis about which motion parameter can be identified and to what extent.

B. Iterative Learning Control (ILC)

Iterative learning control is motivated by the observation that if the system controller is fixed and if the system's initial operating conditions are the same each time it executes, then any errors in the output response will be repeated during each operation. These errors can be recorded during system operation and can then be used to compute modifications to the input signal that will be applied to the system during the next operation. That is, in ILC, refinements are made to the input signal after each trial until the desired performance level is reached. A key, and standard, ILC assumption is that the initial conditions of the system are reset at the beginning of each trial to the same value. In describing the technique of ILC, the word *iterative* is used because of the recursive nature of the system, and the word *learning* is used because of the refinement of the input signal based on past performance in executing a task [1].

An important arena for ILC applications is motion control, particularly robotics. In a robotics application it is common for ILC algorithms to operate based on measurements of joint angles, obtained from encoder measurements. However, the recent widespread availability and low cost of vision-based sensing motivates the idea of performing ILC in robotics applications using vision, which in turn motivates consideration of the ILC-PDS problem. To date, there are few contributions in the ILC literature where vision feedback is used and there have been no contributions that exploit PDS theory in an ILC problem. In the next section we formulate the ILC-PDS problem in general and study the case of 2-D motion with a 1-D perspective measurement in particular.

II. ILC Control of PDS

Consider again an object that is moving along a 3-D trajectory in 3-D space whose motion is observed via a camera-type vision system. Further suppose the motion is iterative in the sense that it starts from some initial condition, executes the trajectory, then resets to the (same) intitial conditions, and executes the trajectory again, over and over. The question we consider is whether the observations on the image plane, a 2-D projection trajectory, suffice to refine the 3-D motion trajectory from trial-to-trial. Our preliminary study shows that the homogeneous output information from the image plane can indeed help to improve the system performance, under the precondition that the initial position of the plant (the moving target) is identical with the desired position. This precondition is referred to as the Identical Initial Condition (IIC) in the ILC literature. We now state our problem formally as:

Definition 2.1: General ILC-PDS Formulation: [6]
Given:

1) Object moving according to a known, iterative motion.
2) Desired observed trajectory (u_d, v_d) on the image plane of a stationary camera.
3) Desired actual trajectory $[X_d(t), Y_d(t), Z_d(t)]^T$ in the object space.
4) Calibrated camera whose parameters, such as the camera's intrinsic parameters, distortion coefficients, and focal length, are known.

Task: Track $(u_d(t), v_d(t))$, along with the 3-D desired trajectory on the image plane and along the desired path $[X_d(t), Y_d(t), Z_d(t)]^T$ in the object space, possibly in the presence of repeatable uncertainties, where uncertainties could arise from calibration of the camera, measurements on the image plane, and about controlling the plant.
Additional Condition: With or without the identical initial condition (IIC).

III. Preliminary Study

The problem formulation in Definition. 2.1 is quite general. In the section, some preliminary simulation results are given in the ideal situation of no noise, both with or without the IIC condition. These simulation results show that, with the assumption of the identical initial condition, the homogeneous output information can be enough to make the moving object track a desired 3-D trajectory.

Consider a 2-D ILC-PDS system described as follows:

1) Motion Dynamics:

$$\begin{bmatrix} \dot{X}(t) \\ \dot{Y}(t) \end{bmatrix}_k = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} X(t) \\ Y(t) \end{bmatrix}_k + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u_k(t). \tag{6}$$

2) Homogeneous Output:

$$y_k(t) = [y_1(t)]_k = [X_k(t)/Y_k(t)], \tag{7}$$

where the output is assumed to be available from a calibrated 1-D camera. Here the index k denotes the trial number. It is assumed that the time variables satisfies $t \in [0, t_f]$, where t_f is the length of each trial.

The specific problem is to control the plant to track a desired 2-D trajectory, $[X_d(t), Y_d(t)]^T$, by adjusting the input $u_k(t)$ from trial-to-trial, using information $y_k(t) = X_k(t)/Y_k(t)$, which is the information directly available from the image plane.

Before theoretically proving the feasibility in the next section, the following simulations are conducted to give an intuitive idea of the problem. Let the motion parameters $[a_{i,j}, b_j]$ for $i, j = 1, 2$ in (6), and the initial conditions, be given by

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} -0.1 & 0.3 \\ 0.1 & -0.2 \end{bmatrix},$$
$$B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 0.15 \\ 0.2 \end{bmatrix}, \tag{8}$$
$$(X_d(0), Y_d(0)) = (X_k(0), Y_k(0)) = (1.0, 0.5).$$

The desired 2-D trajectory is generated by a sinusoidal input $u_d(t) = \sin(t)$, with the trial length defined by $t_f = 20$secs. Suppose for the first iteration, $u_1(t)$ is taken as the step input $u_1(t) = 1$. Apply the following ILC updating law

$$\begin{aligned} u_{k+1}(t) &= u_k(t) + k_1\, e_k(t) + k_2\, \dot{e}_k(t), \\ e_k(t) &= y_d(t) - y_k(t), \end{aligned} \tag{9}$$

where k_1 and k_2 are chosen to be

$$k_1 = k_2 = -0.4. \tag{10}$$

Notice that

1) The learning rates in (10) are chosen by trial and error. Detailed procedures to get the proper learning rate need to be investigated and this will be included in future investigations.
2) No knowledge of the plant dynamics (a_{ij}) are used.
3) The use of the derivative of the error in the update law. This is a key feature of ILC, which allows what appears to be non-causal processing, because the data was actually collected from the past trial.

Figure 1 shows the results after 41 iterations. The two plots in the first row show the homogeneous output $y_1(t)$ and the 2-D desired trajectory with those of the first iteration using $u_1(t) = 1$, where the desired curves are plotted in red and the actual in blue. The x-axis of the left two figures is time t. The second row in Fig. 1 shows the corresponding plots after 41 iterations with the ILC updating law (9) using learning rates (10). It can be observed that the ILC updating law in (9) helps to make the system output, along with the 2-D states, to track their desired values.

The results shown in Fig. 1 assume the IIC condition. When the IIC condition is not satisfied, that is when $(X_d(0), Y_d(0)) \neq (X(0), Y(0))$, will the tracking of the output $y_1(t)$ guarantee the tracking of the 2-D states using the ILC control law (9)? An intuitive answer to this question is "no," which is supported by the simulation results shown in Fig. 2 with $(X(0), Y(0)) = (1.2, 0.9)$. Here it can be observed that, using the ILC updating law (9), when the IIC condition cannot be satisfied, tracking in the perspective output cannot guarantee the tracking the 2-D states. That is, in this case, only the ratio $Y(t)/X(t)$ tracks the desired one, instead of $X(t)$ and $Y(t)$ tracking $X_d(t)$ and $Y_d(t)$ individually, as can be seen from the "2D Trajectory" from the plots in the second row of the figure.

With or without the IIC condition, plotting $\|e_k\|_2 = \|y_d - y_k\|_2$ vs. iteration shows a monotone convergence as can be seen in Figs. 3 and 4, respectively, though of course the norm of the error does not go to zero when the IIC condition is not satisfied.

IV. Proof

The previous section showed that for a 2-D motion observed through a 1-D perspective projection the ILC-PDS system could converge in both axes for the ILC control law (9). In this section, a proof of the convergence is presented.

To begin, from [7], we have the following Corollary:

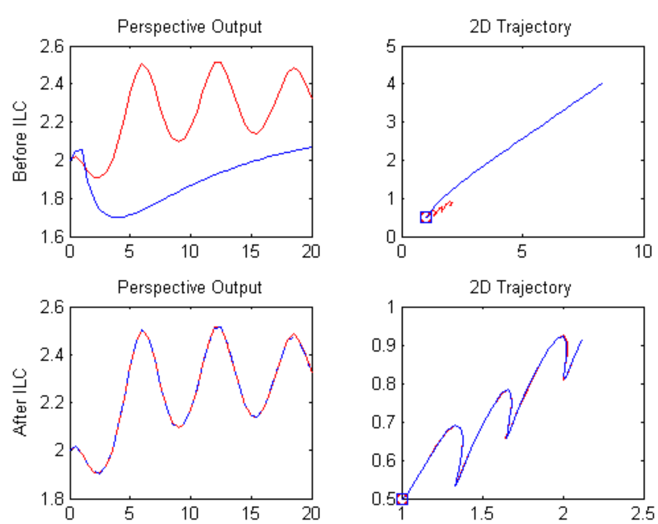

Fig. 1. Simulation results of the ILC control of a 2-D PDS system (under IIC).

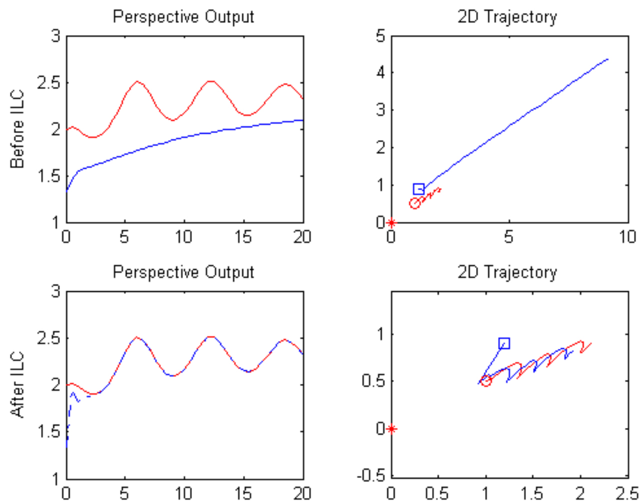

Fig. 2. Simulation results of the ILC control of a 2-D PDS system (without IIC).

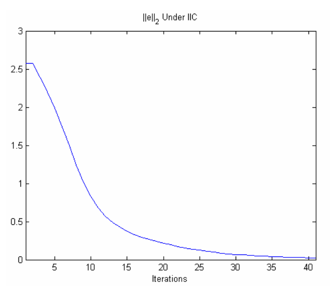

Fig. 3. Simulation results of $\|e_k\|_2 = \|y_d - y_k\|_2$ with the IIC condition.

Corollary 4.1: Suppose that the plant is described by:

$$\dot{x}(t) = a(t, x) + B_s(t)u(t), \quad y(t) = C_s x(t), \tag{11}$$

where C_s is a constant matrix, $B_s(t)$ depends on the state variable x that is bounded, and $a(t, x)$ satisfies the condition

$$\|a(t, x_1) - a(t, x_2)\| \leq \|x_1 - x_2\|. \tag{12}$$

Define

$$e_k(t) = \dot{y}_d(t) - \dot{y}_k(t) \tag{13}$$

and the following learning control scheme

$$\begin{aligned} \dot{v}_k(t) &= A_c(t)v_k(t) + B_c(t)e_k(t), \\ w_k(t) &= C_c(t)v_k(t) + D_c(t)e_k(t), \\ u_{k+1}(t) &= u_k(t) + w_k(t), \end{aligned} \tag{14}$$

is to be applied to this plant. Then, if

$$\|I_m - C_s B_s(t) D_c(t)\| \leq 1, \quad \forall t \in [0, t_f], \tag{15}$$

holds, the error defined by (13) converges to zero in the sense of

$$\|e_{k+1}(t)\|_q \leq b_0 \|e_k(t)\|_q, \tag{16}$$

where I_m denotes the m-dimensional identity matrix, $0 \leq b_0 \leq 1$, and $q \geq 0$. $\|x\|_q = \sup_{t \in [0, t_f]} e^{-qt}\|x\|$ is the so-called λ-norm often used in ILC [1]. The above equation implies

$$\|e_k(t)\|_q \to 0, \quad \text{as} \quad k \to \infty. \tag{17}$$

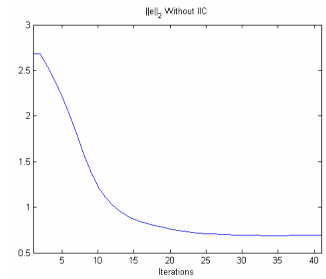

Fig. 4. Simulation results of $\|e_k\|_2 = \|y_d - y_k\|_2$ without the IIC condition.

With this result we can now state the following theorem:

Theorem 4.1: For the 2-D ILC-PDS system defined by (6) and (7), the ILC control law (9), under the IIC condition, results in

$$\lim_{k \to \infty} X_k(t) = X_d(t) \tag{18}$$

$$\lim_{k \to \infty} Y_k(t) = Y_d(t) \tag{19}$$

if $|1 - (b_1 - b_2 x_1) x_2 k_2| < 1$ for all $t \in [0, t_f]$.

Proof: The proof will be in two parts. First, we demonstrate that the ratio $X_k(t)/Y_k(t)$ goes to $X_d(t)/Y_d(t)$ for all t as $k \to \infty$. Then we show that this fact together with the IIC condition results in $X_k(t) \to X_d(t)$ and $Y_k(t) \to Y_d(t)$ for all t as $k \to \infty$.

(a) Convergence of the ratio: Dropping the trial index subscript k for convenience, and defining $y_1 = X(t)/Y(t)$ and $y_2 = 1/Y(t)$, calculate the derivative of $y(t) = (y_1(t), y_2(t))$ from (6) and (7) to be:

$$\begin{aligned} \dot{y}_1 &= a_{12} + (a_{11} - a_{22})y_1 - a_{21}y_1^2 + (b_1 - b_2 y_1)y_2\, u(t), \\ \dot{y}_2 &= -(a_{22} + a_{21}y_1)y_2 - b_2 y_2^2\, u(t), \end{aligned} \tag{20}$$

Substituting the symbol y with x in the above equation, we have:

$$\begin{aligned} \dot{x}_1 &= a_{12} + (a_{11} - a_{22})x_1 - a_{21}x_1^2 + (b_1 - b_2 x_1)x_2\, u(t), \\ \dot{x}_2 &= -(a_{22} + a_{21}y_1)x_2 - b_2 x_2^2\, u(t), \\ y(t) &= [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \end{aligned} \tag{21}$$

which is in the form of (11). Thus, in order to prove that the ILC updating law (9) makes the system (6) converge in the sense of (16), we only need to show that the ILC updating law in (9) is equivalent to, or a special case of, that in (14) with the $e_k(t)$ defined in (13). This is clearly seen to be true by comparing (14) with (9). By letting

$$B_s = [(b_1 - b_2 x_1)x_2, -b_2 x_2^2]^T, \quad C_s = [10], \tag{22}$$

and

$$A_c(t) = 0, \quad C_c(t)B_c(t) = k_1, \quad D_c(t) = k_2, \tag{23}$$

the ILC updating law in (9) is a special form of (14). Thus, we can conclude that if $|1 - (b_1 - b_2 x_1)x_2 k_2| < 1$ for all $t \in [0, t_f]$ then

$$\lim_{k\to\infty} \frac{d}{dt}\left\{\frac{X_d(t)}{Y_d(t)} - \frac{X_k(t)}{Y_k(t)}\right\} = 0 \tag{24}$$

Note that the condition $|1 - (b_1 - b_2 x_1)x_2 k_2| < 1$ must be true for a given problem and introduces some problem in the choice of the gains k_i, but the condition can always be satisfied under the assumption that the motion is bounded.

Next, further, define

$$Z_k(t) = \frac{X_d(t)}{Y_d(t)} - \frac{X_k(t)}{Y_k(t)}.$$

By the IIC assumption we have $Z_k(0) = 0$ and by the above argument, $\lim_{k\to\infty} \dot{Z}_k(t) = 0$. Thus,

$$\lim_{k\to\infty} Z_k(t) = \lim_{k\to\infty}\left(Z_k(0) + \int_0^t \dot{Z}_k(\tau)d\tau\right) \tag{25}$$

$$= Z_k(0) + \int_0^t \left(\lim_{k\to\infty}\dot{Z}_k(\tau)\right)d\tau \tag{26}$$

$$= 0 \tag{27}$$

That is, for all t

$$\lim_{k\to\infty}\left(\frac{X_d(t)}{Y_d(t)} - \frac{X_k(t)}{Y_k(t)}\right) = 0 \tag{28}$$

(b) Convergence of individual axes: Now, using the IIC condition, the convergence of the derivative of the ratio, (24), and the convergence of the ratio, (28), we want to show that $X_k(t) \to X_d(t)$ and $Y_k(t) \to Y_d(t)$.

First, write

$$\lim_{k\to\infty}\frac{X_k(t)}{Y_k(t)} = \frac{X(t)}{Y(t)} = \frac{X_d(t)}{Y_d(t)} \tag{29}$$

so our goal is to show $X(t) = X_d(t)$ and $Y(t) = Y_d(t)$. Let us suppose that

$$\begin{aligned} X(t) &= x_0 + x_1 t + x_2 t^2 + \cdots + x_k t^k + \cdots, \\ Y(t) &= y_0 + y_1 t + y_2 t^2 + \cdots + y_k t^k + \cdots, \\ X_d(t) &= x_0 + x_1^d t + x_2^d t^2 + \cdots + x_k^d t^k + \cdots, \\ Y_d(t) &= y_0 + y_1^d t + y_2^d t^2 + \cdots + y_k^d t^k + \cdots. \end{aligned} \tag{30}$$

We proceed by induction.

[Case 1.] Let $k = 1$, so that

$$\begin{aligned} X(t) &= x_0 + x_1 t, \\ Y(t) &= y_0 + y_1 t, \\ X_d(t) &= x_0 + x_1^d t, \\ Y_d(t) &= y_0 + y_1^d t. \end{aligned} \tag{31}$$

From $\frac{X(t)}{Y(t)} = \frac{X_d(t)}{Y_d(t)}$, we have:

$$\frac{x_0 + x_1 t}{y_0 + y_1 t} = \frac{x_0 + x_1^d t}{y_0 + y_1^d t} \tag{32}$$

or

$$\begin{aligned} x_0 y_0 + x_0 y_1^d t + x_1 y_0 t + x_1 y_1^d t^2 &= \\ y_0 x_0 + y_0 x_1^d t + y_1 x_0 t + y_1 x_1^d t^2 & \end{aligned} \tag{33}$$

Since the above is true for all t, we have:

$$\text{term } t^2: \quad x_1 y_1^d = y_1 x_1^d, \tag{34}$$

$$\text{term } t: \quad x_0 y_1^d + x_1 y_0 = y_0 x_1^d + y_1 x_0 \tag{35}$$

From (34), we have:

$$y_1 = (y_1^d / x_1^d)x_1. \tag{36}$$

Equation (35) becomes:

$$\begin{aligned} x_0(y_1^d - y_1) &= y_0(x_1^d - x_1) \\ x_0(y_1^d - \frac{y_1^d}{x_1^d}x_1) &= y_0(x_1^d - x_1) \\ x_0 y_1^d - x_0\frac{y_1^d}{x_1^d}x_1 &= y_0 x_1^d - y_0 x_1 \\ x_1(x_0\frac{y_1^d}{x_1^d} - y_0) &= x_0 y_1^d - y_0 x_1^d \\ \frac{x_1}{x_1^d}(x_0 y_1^d - y_0 x_1^d) &= x_0 y_1^d - y_0 x_1^d \\ x_1 &= x_1^d. \end{aligned} \tag{37}$$

Combining equations (36) and (37), we have $x_1 = x_1^d, y_1 = y_1^d$ and thus $X(t) = X_d(t)$ and $Y(t) = Y_d(t)$ for signals in (31).

[Case 2.] Consider (30) for $k = 2$[1]:

$$\begin{aligned} X(t) &= x_0 + x_1 t + x_2 t^2, \\ Y(t) &= y_0 + y_1 t + y_2 t^2, \\ X_d(t) &= x_0 + x_1^d t + x_2^d t^2, \\ Y_d(t) &= y_0 + y_1^d t + y_2^d t^2. \end{aligned} \tag{38}$$

Similar to the procedures in *Case 1.*, we have:

$$\frac{x_0 + x_1 t + x_2 t^2}{y_0 + y_1 t + y_2 t^2} = \frac{x_0 + x_1^d t + x_2^d t^2}{y_0 + y_1^d t + y_2^d t^2}$$

and cross-multiplying and equating powers of t gives

$$\text{term } t: \quad x_0 y_1^d + x_1 y_0 = y_0 x_1^d + y_1 x_0 \tag{39}$$

$$\text{term } t^2: \quad x_0 y_2^d + x_1 y_1^d + x_2 y_0 = y_0 x_2^d + y_1 x_1^d + y_2 x_0 \tag{40}$$

$$\text{term } t^3: \quad x_1 y_2^d + x_2 y_1^d = y_1 x_2^d + y_2 x_1^d \tag{41}$$

$$\text{term } t^4: \quad x_2 y_2^d = y_2 x_2^d \tag{42}$$

[1] This case is given to give a sense of the manipulations needed to continue to the more general induction proof of $(k \to k+1)$

Solving the above four equations (39)∼(42) using the Matlab symbolic toolbox gives:[2]

$$x_1 = x_1^d, \quad y_1 = y_1^d, \quad x_2 = x_2^d, \quad y_2 = y_2^d.$$

Thus $X(t) = X_d(t)$ and $Y(t) = Y_d(t)$.

Let us consider another way to prove by using the fact that we have proved, in *Case 1.*, that $x_1 = x_1^d, y_1 = y_1^d$. Due to this, (39) is satisfied automatically. Since we only need to show $x_2 = x_2^d, y_2 = y_2^d$, we only need two equations. Picking up equations (40) and (42) and notice that $x_1 y_1^d = y_1 x_1^d$ due to the result from *Case 1.*, we can have:[3]

$$\text{term}t^4: \qquad x_2 y_2^d = y_2 x_2^d, \tag{43}$$

$$\text{term}t^2: \qquad x_0 y_2^d + x_2 y_0 = y_0 x_2^d + y_2 x_0, \tag{44}$$

which is in a similar form as those in (34) and (35). Thus, we get $x_2 = x_2^d, y_2 = y_2^d$ directly (the procedures are similar to those from (36) to (37)). The above shows that, based on the proof in *Case 1.*, we have proved that $X(t) = X_d(t), Y(t) = Y_d(t)$ for the signals in (38).

[Case 3.] With the preparations from *Case 1.* and *Case 2.*, let us consider the general case of (30). Suppose $X(t) = X_d(t), Y(t) = Y_d(t)$ is true for k. That is, we have $x_i = x_i^d, y_i = y_i^d$ for $i = 1, \cdots, k$ and

$$\begin{aligned} &\frac{x_0 + x_1 t + x_2 t^2 + \cdots + x_k t^k}{y_0 + y_1 t + y_2 t^2 + \cdots + y_k t^k} \\ &= \frac{x_0 + x_1^d t + x_2^d t^2 + \cdots + x_k^d t^k}{y_0 + y_1^d t + y_2^d t^2 + \cdots + y_k^d t^k}. \end{aligned} \tag{45}$$

Consider the case for $k+1$ with

$$\begin{aligned} &\frac{(x_0 + x_1 t + x_2 t^2 + \cdots + x_k t^k) + x_{k+1} t^{k+1}}{(y_0 + y_1 t + y_2 t^2 + \cdots + y_k t^k) + y_{k+1} t^{k+1}} \\ &= \frac{(x_0 + x_1^d t + x_2^d t^2 + \cdots + x_k^d t^k) + x_{k+1}^d t^{k+1}}{(y_0 + y_1^d t + y_2^d t^2 + \cdots + y_k^d t^k) + y_{k+1}^d t^{k+1}} \end{aligned} \tag{46}$$

Then we have:

$$\begin{aligned} &(x_0 + \cdots + x_k t^k)(y_0 + \cdots + y_k t^k) \\ &+(x_0 + \cdots + x_k t^k) y_{k+1}^d t^{k+1} \\ &+(y_0 + \cdots + y_k t^k) x_{k+1} t^{k+1} + x_{k+1} y_{k+1}^d t^{2(k+1)} \\ &= (y_0 + \cdots + y_k t^k)(x_0 + \cdots + x_k t^k) \\ &+(y_0 + \cdots + y_k t^k) x_{k+1}^d t^{k+1} \\ &+(x_0 + \cdots + x_k t^k) y_{k+1} t^{k+1} + x_{k+1}^d y_{k+1} t^{2(k+1)} \end{aligned}$$

From the assumption for case k that $(x_0 + \cdots + x_k t^k)(y_0 + \cdots + y_k t^k) = (y_0 + \cdots + y_k t^k)(x_0 + \cdots + x_k t^k)$ and considering the remaining terms associated with $t^{2(k+1)}$ and t^{k+1} in the above equation, we have:

$$\text{term } t^{2(k+1)}: \qquad x_{k+1} y_{k+1}^d = y_{k+1} x_{k+1}^d, \tag{47}$$

$$\begin{aligned} \text{term } t^{k+1}: \qquad x_0 y_{k+1}^d + x_{k+1} y_0 &= \qquad (48) \\ y_0 x_{k+1}^d + y_{k+1} x_0, &\qquad (49) \end{aligned}$$

which is in a similar form as those in (34), (35) and (43), (44). Accordingly, we have $x_{k+1} = x_{k+1}^d, y_{k+1} = y_{k+1}^d$. Thus, $X(t) = X_d(t), Y(t) = Y_d(t)$ for the signals in (30) for $k = k + 1$.

■

V. Conclusion

In this paper we have formulated the problem of iterative learning control for perspective dynamic systems. For 2-D motion and 1-D perspective measurements we have shown by simulation and analysis that ILC can be used to force both axes of the system to a desired motion using only the perspective measurement, provided that the identical initial condition is satisfied. That is, tracking in the homogeneous outputs assures tracking in the system's states. However, when the IIC condition is violated, the above statement is no longer true. The next question, then, is if the IIC condition can be relaxed and at what expense. When the motion parameters of a moving object are known, perspective nonlinear observers can be applied to estimate the system states from its homogeneous outputs. An intuitive idea is to estimate the system's states, from which the ILC scheme can be built. This remains an open question for future research. It also remains to extend the 2-D result presented here to the 3-D case. Finally we comment that another direction of future research is to consider the perspective ILC idea for distributed parameter systems, for application to tasks such as periodic thermal therapy.

References

[1] Kevin L. Moore, "Iterative learning control: An expository overview," *Applied and Computational Controls, Signal Processing, and Circuits*, vol. 1, no. 1, pp. 425–488, 1998.

[2] Bijoy K. Ghosh and Clyde F. Martin, "Homogeneous dynamical systems theory," *IEEE Transactions on Automatic Control*, vol. 47, no. 3, pp. 462–472, Mar. 2002.

[3] Bijoy K. Ghosh, Hiroshi Inaba, and Satoru Takahashi, "Identification of Riccati dynamics under perspective and orthographic observations," *IEEE Transactions on Automatic Control*, vol. 45, no. 7, pp. 1267–1278, July 2000.

[4] Mrdjan Jankovic and Bijoy K. Ghosh, "Visually guided ranging from observations of points, lines and curves via an identifier based nonlinear observer," *Systems and Control Letters*, vol. 25, pp. 63–73, 1995.

[5] Joao P. Hespanha, "State estimation and control for systems with perspective outputs," in *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, Nevada, USA, Dec. 2002, pp. 2208–2213.

[6] Lili Ma, *Vision-Based Measurements for Dynamic Systems and Control*, Ph.D. thesis, Utah State University, 2004.

[7] Toshiharu Sugie and Toshiro Ono, "An iterative learning control law for dynamical systems," *Automatica*, vol. 27, no. 4, pp. 729–732, 1991.

[2]The Matlab command is: `[x1, x2, y1, y2] = solve('x0 * y1d + y0 * x1 - x0 * y1 - x1d * y0 = 0', 'x0 * y2d + x1 * y1d + x2 * y0 - x0*y2 - x1d * y1 - x2d * y0 = 0', 'x1 * y2d + x2 * y1d - x1d * y2 - x2d * y1 = 0', 'x2 * y2d - x2d * y2 = 0', 'x1, x2, y1, y2')`.

[3]Choosing equations (41) and (42) can also get the same result.