Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

An Iterative Predictive Learning Control Approach With Application to Energy Efficient Train Trajectory Tracking

Heqing Sun*. Zhongsheng Hou*. Dayou Li**
* Advanced Control Systems Laboratory, Beijing Jiaotong University, Beijing, China
(Tel: 86-10-51688617; e-mail: sunheqingbjtu@gmail.com; zhshhou@bjtu.edu.cn)
** University of Bedfordshire, Park Square, Luton, LU1 3JU, UK (email:Dayou.Li@beds.ac.uk)

Abstract: An approach of iterative predictive learning control (IPLC) is studied with consideration to both precise train trajectory tracking and energy efficient operation. Through designing the predictive cost function, the IPLC approach for input-affine nonlinear systems is formulated and solved in this paper. Its application to train operation is detailed to compromise between punctuality and energy consumption. Rigorous theoretical analysis confirms that the proposed approach can guarantee the asymptotic convergence of train speed and position to desired profiles along iteration axis. Simulation result shows its effectiveness and energy efficiency.

1. INTRODUCTION

Due to the large capacity, low energy consumption, and high punctuality rate, railway transportation plays a more and more important role in mass transit both within and between cities. Recently, high-speed railway and subway are developing with great achievements worldwide, and automatic driving is one of the key technologies for the train control and monitoring to ensure safety and comfort.

Speed regulation is one of the main tasks of automation train operation (ATO), along with programmed stopping, performance level regulation, etc.. At present, various control methods have been developed to deal with the train trajectory tracking problem, such as PID (Murtaza & Garg, 1989), fuzzy logic control (Chang & Xu, 2000), and nonlinear output regulator (Zhuan & Xia, 2008).

It is worth noting that the train, especially high-speed train and subway train, operates according to its operation diagrams on the same track strictly every day, every week, or even every year. This operation pattern leaves train operation an inherent outstanding feature, repetition. However, the majority of existing ATO methods, including aforementioned methods, neglect this significant unique characteristic of train motion, and a large amount of valuable historical information is wasted. Without learning from the recurrent train operation process, control performance of train operation cannot be improved no matter how many times it runs.

In control theory community, there is a control method called iterative learning control (ILC), just proposed for addressing the control problem of a system repeating the same control task on a finite interval. After first proposed by Arimoto, Kawamura, & Miyazaki (1984), ILC has been well studied both in theory (Xu & Tan, 2003; Ahn, Moore, & Chen, 2007) and in application (Bristow, Alleyne, & Zheng, 2004; Hou, Xu, & Yan, 2008). Compared to other control methods, ILC possesses capacity to learn and improve control performance from previous executions. Moreover, less priori knowledge of system model is required. All these characteristics indicate that it is an ideal tool to deal with train operation.

There are already a few works on ILC based train operation, such as terminal ILC based train station stop (Hou et al., 2011), norm optimal ILC based train trajectory tracking (Sun, Hou, & Li, 2012), and coordinated ILC based multi-train safe operation (Sun, Hou, & Li, 2013). However, none of these works takes energy efficient operation into account.

In general, the energy conservation of train operation are studied either by utilization of regenerative braking (Estima & Marques, 2012; Yang et al., 2013) or by cybernetic studies (Howlett & Pudney, 1995; Bocharnikov, Tobias, & Roberts, 2010; Ke, Lin, & Lai, 2011; Ke, Lin, & Yang, 2012). In this work, we focus on the latter. In existing studies, several control methods have been applied to energy efficient train operation, such as optimal control theory (Howlett & Pudney, 1995), genetic algorithm (Bocharnikov, Tobias, & Roberts, 2010), fuzzy logic control (Bocharnikov et al., 2007), max-min ant system (Ke, Lin, & Yang, 2012;), and combination of them (Bocharnikov et al., 2007; Ke, Lin, & Lai, 2011).

This paper aims at developing an iterative predictive learning control (IPLC) approach for the input-affine nonlinear systems. Note that the combination of various predictive control methods and ILC has been studied, but most of them are studied for linear systems (Lee & Lee, 2000; Shi, Gao, & Wu, 2007), and little relative theoretical research is on nonlinear systems (Balaji et al., 2007; Cueli & Bordons, 2008). Different from existing works, the proposed IPLC approach is designed for input affine nonlinear systems, and its prediction is along iteration axis, rather than time axis, which can improve the transient behavior along iteration axis. Its application to train operation makes a compromise between train tracking precision and energy consumption.

The other parts of this paper are organized as follows. Section 2 shows the train motion dynamics model and problem formulation. The IPLC approach for input-affine nonlinear systems is formulated and solved in Section 3. Three sets of simulation results are provided is Section 4. Finally, Section 5 concludes the paper.

2. PROBLEM FORMULATION

2.1 Train Motion Dynamics Model and Discretization

The motion dynamics model of a train is firstly proposed in Davis (1926), and then Hay gave a detailed description of the model in his monograph (Hay,1982) as follows,

$$dv/dt = u - w(v) - g(s), \tag{1}$$

$$ds/dt = v, \tag{2}$$

$$w(v) = c_a v^2 + c_v v + c_0, \tag{3}$$

$$g(s) = l \cdot \sin\left(\theta(s)\right), \tag{4}$$

where v (m/s) is train speed, u (N/kg) is traction or braking force on unit mass, s (m) is position of the train, $w(v)$ (N/kg) is the general resistance on unit mass, $g(s)$ (N/kg) is the additional resistance on unit mass caused by slope angle and degree of curvature, c_a, c_v, c_0 are coefficients of general resistance, depending on many factors, such as vehicle characteristic, weather condition, wind speed, wind direction, etc., l is coefficient of additional resistance, and $\theta(s)$ is slope angle at position s.

Above train motion model is continuous-time. However, discrete-time is more convenience for computer-based control methods and data storage. Therefore, we transform above equations into discrete-time difference equations. Here we consider the operation diagram over a fixed time interval $[0,T]$, i.e., $t\in[0,T]$. By choosing Δ as sample time, the whole time interval in discrete-time domain becomes $k=0,1,\cdots,K$, and $T=\Delta\cdot K$ holds. According to Euler Formula, differential equations (1)-(4) can be discretized as,

$$v(k+1)=f\left(v(k)\right)+h\left(s(k)\right)+\Delta\cdot u(k), \tag{5}$$

$$s(k+1)=s(k)+\Delta\cdot v(k), \tag{6}$$

where $f\left(v(k)\right)=-c_a\Delta\cdot v^2(k)-\left(c_v\Delta-1\right)\cdot v(k)-c_0\Delta$, and $h\left(s(k)\right)=-\Delta\cdot g\left(s(k)\right)$.

2.2 State Space Representation and Assumptions

By defining speed and position of the train as system states, i.e., $\boldsymbol{x}_n(k)=\left[v_n(k),s_n(k)\right]^T$, the train motion dynamics can be rewritten as following,

$$\boldsymbol{x}_n(k+1)=\boldsymbol{f}\left(\boldsymbol{x}_n(k)\right)+\boldsymbol{b}\cdot u_n(k), \tag{7}$$

where $\boldsymbol{f}\left(\boldsymbol{x}_n(k)\right)=\begin{bmatrix} f\left(v_n(k)\right)+h\left(s_n(k)\right) \\ \Delta\cdot v_n(k)+s_n(k)\end{bmatrix}$, $\boldsymbol{b}=\begin{bmatrix}\Delta\\0\end{bmatrix}$.

For the convergence analysis, some reasonable assumptions, which are common in ILC design, are made first.

Assumption 1. Function $\boldsymbol{f}(\cdot)$ is uniformly globally Lipschitz continuous on its compact set $\boldsymbol{\Omega}$ with respect to its arguments,

$$\left\|\boldsymbol{f}\left(\boldsymbol{x}_1(k)\right)-\boldsymbol{f}\left(\boldsymbol{x}_2(k)\right)\right\|\le k_f\cdot\left\|\boldsymbol{x}_1(k)-\boldsymbol{x}_2(k)\right\|, \tag{8}$$

where k_f is Lipschitz constant, compact set $\boldsymbol{\Omega}=\mathbf{V}\times\mathbf{S}$, $\mathbf{V}$ and $\mathbf{S}$ are the range of speed and position of train operation.

Assumption 2. The re-initialization condition is satisfied throughout the repeated iterations, i.e.,

$$\boldsymbol{x}_n(0)=\boldsymbol{x}_d(0), \quad \forall n, \tag{9}$$

where $\boldsymbol{x}_d(0)$ are initial values of the desired system states.

Assumption 3. There exists an appropriate control $u_d(k)$ which can drive the system states to track $\boldsymbol{x}_d(k+1)$ for system (7) over the whole finite interval $k\in[0,K-1]$, i.e.,

$$\boldsymbol{x}_d(k+1)=\boldsymbol{f}\left(\boldsymbol{x}_d(k)\right)+\boldsymbol{b}\cdot u_d(k). \tag{10}$$

Remark 1. Assumption 1 requests the train motion dynamics to be globally Lipschitz continuous, which can be naturally satisfied, since the train motion dynamics (7) is bounded continuously differentiable on its bounded compact set $\boldsymbol{\Omega}$. Assumption 2 can be also satisfied, because if the train operation task is set to run from one station to the next, the train always departs from one station after stopping for a while according to the prescheduled timetable, i.e., $v_n(0)=v_d(0)=0$, $s_n(0)=s_d(0)=0$. Assumption 3 is a reasonable assumption that the task assigned for control should be feasible.

3. ITERATIVE PREDICTIVE LEARNING CONTROL APPROACH

In this section, the iterative predictive model is first developed for prediction and controller design. And then the IPLC controller is designed by optimizing a cost function. At last, the convergence of the IPLC approach is studied.

3.1 Iterative Predictive Model

In order to predict the state sequence in future iterations, a predictive model in iteration domain will be constructed briefly. The interested reader can refer Sun & Hou (2013).

The prediction model is,

$$\boldsymbol{X}_{n,L}(k+1)=\mathbf{1}\cdot\mathbf{B}\cdot\Delta\boldsymbol{U}_{n,L}(k)+\mathbf{1}\cdot\hat{\boldsymbol{H}}_{n,L}(k), \tag{11}$$

where $\boldsymbol{X}_{n,L}(k+1)=\left[\boldsymbol{x}^T_{n|n,k}(k+1),\cdots,\boldsymbol{x}^T_{n+L-1|n,k}(k+1)\right]^T\in\mathbb{R}^{2L\times1}$,

$\boldsymbol{U}_{n,L}(k)=\left[u_n(k) \quad \cdots \quad u_{n+L-1}(k)\right]^T\in\mathbb{R}^{L\times1}$,

$\Delta\boldsymbol{U}_{n,L}(k)=\boldsymbol{U}_{n,L}(k)-\boldsymbol{U}_{n-1,L}(k)$ $\quad \mathbf{1}=\begin{bmatrix}\mathbf{I} & & \mathbf{0}\\ \vdots & \ddots & \\ \mathbf{I} & \cdots & \mathbf{I}\end{bmatrix}_{2L\times2L}$,

$$\mathbf{B}=\begin{bmatrix}\boldsymbol{b} & & \mathbf{0}\\ & \ddots & \\ \mathbf{0} & & \boldsymbol{b}\end{bmatrix}_{2L\times L},\ \hat{\boldsymbol{H}}_{n,L}(k)=\begin{bmatrix}\boldsymbol{x}_{n-1}(k+1)+\hat{\mathbf{A}}_n(k)\Delta\boldsymbol{x}_n(k)\\ \sum_{j=0}^{k-1}\prod_{i=j+1}^{k}\hat{\mathbf{A}}_{n+1}(i)\cdot\boldsymbol{b}\cdot\Delta u_{n+1}(j)\\ \vdots\\ \sum_{j=0}^{k-1}\prod_{i=j+1}^{k}\hat{\mathbf{A}}_{n+L-1}(i)\cdot\boldsymbol{b}\cdot\Delta u_{n+L-1}(j)\end{bmatrix}.$$

According to the definition of $\mathbf{A}_n(k)$,

$$\hat{\mathbf{A}}^T_n(k)=\left[\hat{\boldsymbol{f}}_{1,n}(k),\boldsymbol{f}_2\right], \tag{12}$$

where $\boldsymbol{f}^T_2=[\Delta \ \ 1]$.

From (11), we find that the sequence $\hat{\boldsymbol{f}}_{1,n}(k),\cdots,\hat{\boldsymbol{f}}_{1,n+L-1}(k)$ are still needed to be estimated. Here we use projection algorithm along iteration axis.

First, $\hat{\boldsymbol{f}}_{1,n}(k)$ will be predicted as,

$$\hat{f}'_{1,n}(k)=\hat{f}_{1,n-1}(k)+\frac{\gamma}{\mu+\|x_{n-1}(k)-x_{n-2}(k-1)\|^2}\cdot(x_{n-1}(k)-x_{n-2}(k))\times\{[1\ 0]\cdot(x_{n-1}(k+1)-x_{n-2}(k+1)-b\cdot(u_{n-1}(k)-u_{n-2}(k)))-(x_{n-1}(k)-x_{n-2}(k))^T\hat{\mathbf{f}}_{1,n-1}(k)\} \quad (13)$$

$$\hat{f}_{1,n}(k)=\begin{cases}\hat{f}_{1,\max} & \hat{f}'_{1,n-1}(k)\ge\hat{f}_{1,\max},\\ \hat{f}'_{1,n-1}(k) & \hat{f}_{1,\min}<\hat{f}'_{1,n-1}(k)<\hat{f}_{1,\max},\\ \hat{f}_{1,\min} & \hat{f}'_{1,n-1}(k)\le\hat{f}_{1,\min}.\end{cases} \quad (14)$$

And then $\hat{f}^T_{1,n+l}(k)=[\hat{\varphi}^1_{n+l}(k),\hat{\varphi}^2_{n+l}(k)]$ ($l=1,\cdots,L-1$) are,

$$\hat{\varphi}^1_{n+l}(k)=\Theta^1_{n+l-1}(k)^T\Psi^1_{n+l-1}(k), \quad (15)$$

$$\hat{\varphi}^2_{n+l}(k)=\Theta^2_{n+l-1}(k)^T\Psi^2_{n+l-1}(k), \quad (16)$$

$$\Theta^1_{n+l-1}(k)=\Theta^1_{n+l-2}(k)+\frac{\Psi^1_{n+l-2}(k)}{\delta^1+\|\Psi^1_{n+l-2}(k)\|^2}\times[\hat{\varphi}^1_{n+l-1}(k)-\Psi^1_{n+l-2}(k)^T\Theta^1_{n+l-2}(k)], \quad (17)$$

$$\Theta^2_{n+l-1}(k)=\Theta^2_{n+l-2}(k)+\frac{\Psi^2_{n+l-2}(k)}{\delta^2+\|\Psi^2_{n+l-2}(k)\|^2}\times[\hat{\varphi}^2_{n+l-1}(k)-\Psi^2_{n+l-2}(k)^T\Theta^2_{n+l-2}(k)], \quad (18)$$

where $\Theta^1_{n+l-1}(k)=[\theta^1_{1,n+l}(k),\theta^1_{2,n+l}(k),\cdots,\theta^1_{n_p,n+l}(k)]^T$, $\Theta^2_{n+l-1}(k)=[\theta^2_{1,n+l}(k),\theta^2_{2,n+l}(k),\cdots,\theta^2_{n_p,n+l}(k)]^T$, $\Psi^1_{n+l-1}(k)=[\hat{\varphi}^1_{n+l-1}(k),\hat{\varphi}^1_{n+l-2}(k),\cdots,\hat{\varphi}^1_{n+l-n_p}(k)]^T$, $\Psi^2_{n+l-1}(k)=[\hat{\varphi}^2_{n+l-1}(k),\hat{\varphi}^2_{n+l-2}(k),\cdots,\hat{\varphi}^2_{n+l-n_p}(k)]^T$, and n_p is a proper order.

Therefore, according to (12)-(18), all of the parameters in $\boldsymbol{H}_{n,L}(k)$ have been predicted, and the prediction model (11) can be used for controller design.

3.2 Controller Design

According to the demands of train trajectory tracking and energy efficient operation, the following quadratic prediction objective $J_{n,L}(k)$ is considered,

$$J_{n,L}(k)=\delta\boldsymbol{X}^T_{n,L}(k+1)\cdot\mathbf{Q}\cdot\delta\boldsymbol{X}_{n,L}(k+1)+\Delta\boldsymbol{U}^T_{n,L}(k)\cdot\mathbf{R}\cdot\Delta\boldsymbol{U}_{n,L}(k)+\boldsymbol{U}^T_{n,L}(k)\cdot\mathbf{S}\cdot\boldsymbol{U}_{n,L}(k), \quad (19)$$

where $\delta\boldsymbol{X}_{n,L}(k)=[\delta\boldsymbol{x}^T_{n|n,k}(k),\cdots,\delta\boldsymbol{x}^T_{n+L-1|n,k}(k)]^T$, $\delta\boldsymbol{x}_{n+l|n,k}(k)=\boldsymbol{x}_d(k)-\boldsymbol{x}_{n+l|n,k}(k)$ ($l=0,1,\cdots,L-1$), $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{S}$ are weighting matrices.

Note that weighting matrix $\mathbf{S}$ is on the input force, which determines the energy consumption in the prediction horizon. Specifically, if $\mathbf{S}=\mathbf{0}$, the cost function is irrelevant to the energy consumption, and only focuses on the tracking control. Intuitively, the larger $\mathbf{S}$ is, the controller will be more energy efficient, which would be illustrated in simulations.

From (11), it yields,

$$\delta\boldsymbol{X}_{n,L}(k+1)=\mathbf{1}\cdot\delta\hat{\boldsymbol{H}}_{n,L}(k)-\mathbf{1}\cdot\mathbf{B}\cdot\Delta\mathbf{U}_{n,L}(k), \quad (20)$$

where $$\delta\hat{\boldsymbol{H}}_{n,L}(k)=\begin{bmatrix}\delta\hat{\boldsymbol{h}}_{n,1}(k)\\ -\sum_{j=0}^{k-1}\prod_{i=j+1}^{k}\hat{\mathbf{A}}_{n+1}(i)\cdot\boldsymbol{b}\cdot\Delta u_{n+1}(j)\\ \vdots\\ -\sum_{j=0}^{k-1}\prod_{i=j+1}^{k}\hat{\mathbf{A}}_{n+L-1}(i)\cdot\boldsymbol{b}\cdot\Delta u_{n+L-1}(j)\end{bmatrix},$$

$\delta\hat{\boldsymbol{h}}_{n,1}(k)=\delta\boldsymbol{x}_{n-1}(k+1)-\hat{\mathbf{A}}_n(k)\Delta\boldsymbol{x}_n(k)$.

By substituting (20) into (19) and using the optimal condition $\frac{\partial J_{n,L}(k)}{\partial\boldsymbol{U}_{n,L}(k)}=0$, the update law for IPLC can be got,

$$\boldsymbol{U}_{n,L}(k)=(\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\mathbf{B}+\mathbf{R}+\mathbf{S})^{-1}(\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\mathbf{B}+\mathbf{R})\boldsymbol{U}_{n-1,L}(k)+(\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\mathbf{B}+\mathbf{R}+\mathbf{S})^{-1}\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\cdot\delta\hat{\boldsymbol{H}}_{n,L}(k). \quad (21)$$

In terms of receding horizon principle, only the control input in current iteration is actually executed into the system,

$$u_n(k)=\boldsymbol{g}\cdot(\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\mathbf{B}+\mathbf{R}+\mathbf{S})^{-1}(\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\mathbf{B}+\mathbf{R})\boldsymbol{U}_{n-1,L}(k)+\boldsymbol{g}\cdot(\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\mathbf{B}+\mathbf{R}+\mathbf{S})^{-1}\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\cdot\delta\hat{\boldsymbol{H}}_{n,L}(k). \quad (22)$$

where $\boldsymbol{g}=[1\ \ 0\ \ \cdots\ \ 0]\in\mathbb{R}^{1\times L}$.

Here, the procedure for the IPLC implementation will be summarized. First, assume that the system has executed at sample time k in n-th iteration.

Step 1. $\mathbf{A}_n(k)$ is estimated by (12)-(14);

Step 2. $\mathbf{A}_{n+l}(k)$ ($l=1,\cdots,L-1$) are predicted by (15)-(18);

Step 3. A control sequence at sample time k in n-th, $(n+L-1)$-th iterations can be computed by (21), and only the first element, namely the control input at sample time k in n-th iteration, is implemented (22).

3.3 Convergence Analysis

Theorem 1. For system (7) satisfying Assumption 1-3, if the system is controlled by the IPLC update law in (22), together with parameter prediction algorithms (12)-(18), convergence of the controlled system along iteration domain is guaranteed by any symmetric positive definite matrix $\mathbf{Q}$, $\mathbf{R}=r\mathbf{I}$ ($r>0$), and $\mathbf{S}$ satisfying that the eigenvalues of $\boldsymbol{\kappa}$ are less than 1, where,

$$\boldsymbol{\kappa}=(\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\mathbf{B}+\mathbf{R}+\mathbf{S})^{-1}(\mathbf{R}+\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}(\mathbf{B}-\mathbf{b}_0)), \quad (23)$$

$$\mathbf{b}_0=\begin{bmatrix}\boldsymbol{b} & \boldsymbol{0} & \cdots & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & & \\ \vdots & & \ddots & \\ \boldsymbol{0} & & & \boldsymbol{0}\end{bmatrix}. \quad (24)$$

Proof. Due to the space limiation, only a concise proof is given here. For further detail, please refer to Sun & Hou (2013).

From (7) and (21), we obtain,

$$\boldsymbol{U}_{n,L}(k)=(\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\mathbf{B}+\mathbf{R}+\mathbf{S})^{-1}\times(\mathbf{R}+\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}(\mathbf{B}-\mathbf{b}_0))\boldsymbol{U}_{n-1,L}(k)+(\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\mathbf{B}+\mathbf{R}+\mathbf{S})^{-1}\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\cdot\delta\hat{\boldsymbol{H}}'_{n,L}(k), \quad (25)$$

where,

$$\delta\hat{\boldsymbol{H}}'_{n,L}(k)=\begin{bmatrix}\delta\hat{\boldsymbol{h}}'_{n,1}(k)\\ -\sum_{j=0}^{k-1}\prod_{i=j+1}^{k}\hat{\mathbf{A}}_{n+1}(i)\cdot\boldsymbol{b}\cdot\Delta u_{n+1}(j)\\ \vdots\\ -\sum_{j=0}^{k-1}\prod_{i=j+1}^{k}\hat{\mathbf{A}}_{n+L-1}(i)\cdot\boldsymbol{b}\cdot\Delta u_{n+L-1}(j)\end{bmatrix},\tag{26}$$

$$\begin{aligned}\delta\hat{\boldsymbol{h}}'_{n,1}(k)&=\sum_{j=0}^{k-1}\prod_{i=j+1}^{k}\mathbf{\Phi}_{n-1}(i)\cdot\boldsymbol{b}\cdot\delta u_{n-1}(j)\\ &\quad+\boldsymbol{b}\cdot u_d(k)-\sum_{j=0}^{k-1}\prod_{i=j+1}^{k}\mathbf{A}_n(i)\cdot\boldsymbol{b}\cdot\Delta u_n(j).\end{aligned}\tag{27}$$

By defining $\mathbf{U}_{n,L}=\left[\mathbf{U}_{n,L}^T(0),\cdots,\mathbf{U}_{n,L}^T(K-1)\right]^T$, (25) becomes,

$$\boldsymbol{U}_{n,L}=\mathbf{L}_{u,1}\cdot\boldsymbol{U}_{n-1,L}+\mathbf{L}_{u,n,2}\cdot\boldsymbol{U}_{n-1,L}+\mathbf{L}_{u,n,3}\cdot\boldsymbol{U}_{n,L}+\mathbf{L}_{u,n,4}\cdot\boldsymbol{U}_d,\tag{28}$$

where $\mathbf{L}_{u,1}=\begin{bmatrix}\boldsymbol{\kappa}&&\\&\ddots&\\&&\boldsymbol{\kappa}\end{bmatrix}$, $\mathbf{L}_{u,n,2}=\begin{bmatrix}\boldsymbol{0}&&&\\ \gamma_n^{1,0}&\ddots&&\\ \vdots&\ddots&\ddots&\\ \gamma_n^{K-1,0}&\cdots&\gamma_n^{K-1,K-2}&\boldsymbol{0}\end{bmatrix}$,

$\mathbf{L}_{u,n,3}=\begin{bmatrix}\boldsymbol{0}&&&\\ \boldsymbol{\beta}_n^{1,0}&\ddots&&\\ \vdots&\ddots&\ddots&\\ \boldsymbol{\beta}_n^{K-1,0}&\cdots&\boldsymbol{\beta}_n^{K-1,K-2}&\boldsymbol{0}\end{bmatrix}$, $\mathbf{L}_{u,n,4}=\begin{bmatrix}\boldsymbol{\alpha}&&&\\ \boldsymbol{\alpha}_n^{1,0}&\ddots&&\\ \vdots&\ddots&\ddots&\\ \boldsymbol{\alpha}_n^{K-1,0}&\cdots&\boldsymbol{\alpha}_n^{K-1,K-2}&\boldsymbol{\alpha}\end{bmatrix}$,

$$\boldsymbol{\kappa}=\left(\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\mathbf{B}+\mathbf{R}+\mathbf{S}\right)^{-1}\left(\mathbf{R}+\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\left(\mathbf{B}-\mathbf{b}_0\right)\right),$$

$$\gamma_n^{i,j}=\left(\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\mathbf{B}+\mathbf{R}+\mathbf{S}\right)^{-1}\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\begin{bmatrix}\prod_{l=j+1}^{i}\left(\mathbf{A}_n(l)-\mathbf{\Phi}_{n-1}(l)\right)\cdot\boldsymbol{b}\\ \prod_{l=j+1}^{i}\mathbf{A}_{n+1}(l)\cdot\boldsymbol{b}\\ \vdots\\ \prod_{l=j+1}^{i}\mathbf{A}_{n+L-1}(l)\cdot\boldsymbol{b}\end{bmatrix},$$

$$\boldsymbol{\beta}_n^{i,j}=-\left(\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\mathbf{B}+\mathbf{R}+\mathbf{S}\right)^{-1}\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\begin{bmatrix}\prod_{l=j+1}^{i}\mathbf{A}_n(l)\cdot\boldsymbol{b}\\ \vdots\\ \prod_{l=j+1}^{i}\mathbf{A}_{n+L-1}(l)\cdot\boldsymbol{b}\end{bmatrix},$$

$$\boldsymbol{\alpha}=\left(\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\mathbf{B}+\mathbf{R}+\mathbf{S}\right)^{-1}\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\cdot\begin{bmatrix}\boldsymbol{b}\\ \boldsymbol{0}\\ \vdots\\ \boldsymbol{0}\end{bmatrix},$$

$$\boldsymbol{\alpha}_n^{i,j}=\left(\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\mathbf{B}+\mathbf{R}+\mathbf{S}\right)^{-1}\mathbf{B}^T\mathbf{1}^T\mathbf{Q}\mathbf{1}\begin{bmatrix}\prod_{l=j+1}^{i}\mathbf{\Phi}_{n-1}(l)\cdot\boldsymbol{b}\\ \boldsymbol{0}\\ \vdots\\ \boldsymbol{0}\end{bmatrix}.$$

From (28), it derives,

$$\boldsymbol{U}_{n,L}=\mathbf{\Gamma}_n^{-1}\mathbf{K}_n\boldsymbol{U}_{n-1,L}+\mathbf{\Gamma}_n^{-1}\mathbf{L}_{u,n,4}\cdot\boldsymbol{U}_d,\tag{29}$$

where $\mathbf{\Gamma}_n=\mathbf{I}-\mathbf{L}_{u,n,3}=\begin{bmatrix}\mathbf{I}&&&\\ -\boldsymbol{\beta}_n^{1,0}&\ddots&&\\ \vdots&\ddots&\ddots&\\ -\boldsymbol{\beta}_n^{K-1,0}&\cdots&-\boldsymbol{\beta}_n^{K-1,K-2}&\mathbf{I}\end{bmatrix}$,

$$\mathbf{K}_n=\mathbf{L}_{u,1}+\mathbf{L}_{u,n,2}=\begin{bmatrix}\boldsymbol{\kappa}&&&\\ \gamma_n^{1,0}&\ddots&&\\ \vdots&\ddots&\ddots&\\ \gamma_n^{K-1,0}&\cdots&\gamma_n^{K-1,K-2}&\boldsymbol{\kappa}\end{bmatrix}.$$

Note that $\mathbf{\Gamma}_n$ is a block lower triangular matrix with unit matrix as its diagonal block, thus $\mathbf{\Gamma}_n^{-1}\mathbf{K}_n$ and $\boldsymbol{\kappa}$ share the same eigenvalues.

Since $\mathbf{\Phi}_n(k)$ is bounded for $k=0,1,\cdots,K$ and $n=1,2,\cdots$, there exist a matrix $\bar{\mathbf{Z}}=\begin{bmatrix}\left|\boldsymbol{\kappa}^{0,0}\right|&&&\mathbf{0}\\ \left|\mathbf{h}^{0,1}\right|&\ddots&&\\ \vdots&\ddots&\ddots&\\ \left|\mathbf{h}^{K-1,0}\right|&\cdots&\left|\mathbf{h}^{K-1,K-2}\right|&\left|\boldsymbol{\kappa}^{K-1,K-1}\right|\end{bmatrix}$ satisfying the following inequality,

$$\mathbf{\Gamma}_n^{-1}\mathbf{K}_n\le\bar{\mathbf{Z}}_1,\ \forall n=1,2,\cdots,\tag{30}$$

$$\mathbf{\Gamma}_n^{-1}\mathbf{L}_{u,n,4}\le\bar{\mathbf{Z}}_2,\ \forall n=1,2,\cdots.\tag{31}$$

Combining (29)-(31), it derives,

$$\mathbf{U}_{n,L}\le\bar{\mathbf{Z}}_1\cdot\mathbf{U}_{n-1,L}+\bar{\mathbf{Z}}_2\cdot\boldsymbol{U}_d.\tag{32}$$

If all of the eigenvalues of $|\boldsymbol{\kappa}|$ are less than 1, $\bar{\mathbf{Z}}_1$ is a stable matrix. Therefore, from (32) we have the asymptotic convergence of $\boldsymbol{u}_n$ along iteration axis.

Remark 2. Note that all elements in matrix $|\boldsymbol{\kappa}^{k,k}|$ are known, i.e., $\mathbf{B}$, $\mathbf{Q}$, and $\mathbf{R}$, which means its eigenvalues can be calculated directly and accurately. As for train motion dynamic system, exact values of system parameters, such as c_0, c_v, c_a, l, are not necessary. This indicates that it is more suitable to apply IPLC to train trajectory tracking problem than other model based control approaches in practice.

4. NUMERICAL SIMULATIONS

In this section, we will verify the validity of the proposed IPLC approach and the effects of its predictive horizon and weighting matrices through numerical simulations. The train motion dynamic system is applied and simulated in MATLAB.

The chosen railway track is 36.28km long with an upgrade of 9.68km in length. Sample time is chosen to be 0.01s. Fig. 1 shows the route vertical profile. The actual parameters of the train are listed below, which are only for simulation of train motion, $c_a=1.5\times10^{-6}$, $c_v=7.5\times10^{-5}$ and $c_0=1.66\times10^{-2}$. The additional resistance $g(s)$ is a piecewise continuous function of the displacement as shown in Fig. 2. The desired state profiles, i.e., speed and position trajectory profiles, is given in Fig. 3.

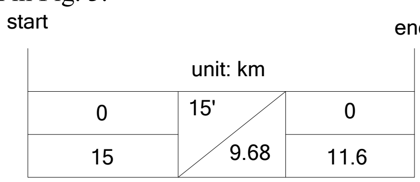

Fig. 1. Vertical profile of the track

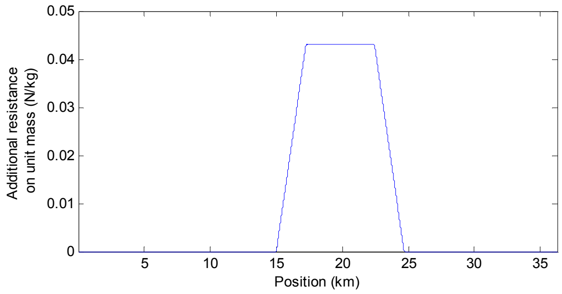

Fig. 2. Additional resistance along the track

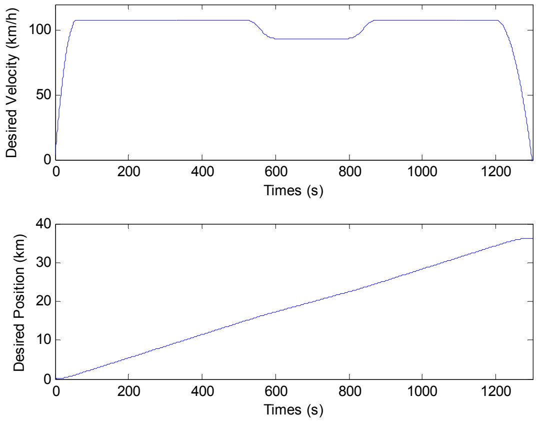

Fig. 3. Desired speed and position trajectory profile

Three sets of simulations are provided here. At first, the tracking ability of the proposed IPLC and its superiority to non-predictive one is shown. And then the effect of weighting matrix $\mathbf{R}$ on the convergence rate is illustrated. At last, IPLC algorithms with different $\mathbf{S}$ are simulated to demonstrate its function of energy conservation.

Firstly, two cases are simulated to show the effectiveness and practicability of IPLC. Case I, IPLC algorithm with predictive horizon 1 ($L=1$), which equals to non-predictive optimal ILC, and $\mathbf{Q}=\mathbf{I},\mathbf{R}=5,\mathbf{S}=0.5$; Case II, IPLC algorithm with predictive horizon 3 ($L=3$), and $\mathbf{Q}=\mathbf{I},\mathbf{R}=5\cdot\mathbf{I},\mathbf{S}=0.5\cdot\mathbf{I}$.

Fig. 4 gives the learning errors of Case I and II. Here learning error is defined as the root mean square (RMS) of the state errors over the whole time interval. From Fig. 4, it can be observed clearly that by applying the proposed IPLC, actual train speed and position converge to the desired ones just after a few iterations. What's more, the converged learning error of IPLC is smaller than that of optimal ILC, where illustrates the superiority of predictive part in the proposed approach.

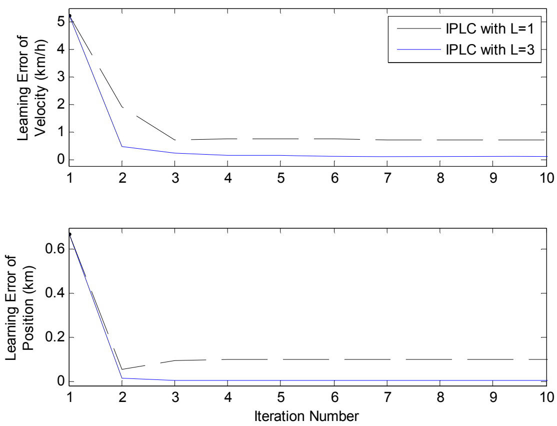

Fig. 4. Learning errors by IPLC with different predictive horizons (Case I, II)

Secondly, three cases are simulated to show the effect of weighting matrix $\mathbf{R}$. The predictive horizons of Case III, IV and V are all 5 ($L=5$), and their weighting matrices $\mathbf{Q},\mathbf{S}$ are set to be $\mathbf{Q}=\mathbf{I},\mathbf{S}=0.5\cdot\mathbf{I}$. The different weighting matrices $\mathbf{R}$ are applied: Case III. $\mathbf{R}=\mathbf{0}$, Case IV. $\mathbf{R}=30\cdot\mathbf{I}$, and Case V. $\mathbf{R}=50\cdot\mathbf{I}$.

Fig. 5 provides learning errors of Case III, IV and V. Actual train speed and position converge to the desired ones by applying all these three IPLC algorithms. Moreover, the larger the weighting matrix R is, the faster the convergence rate of learning error is.

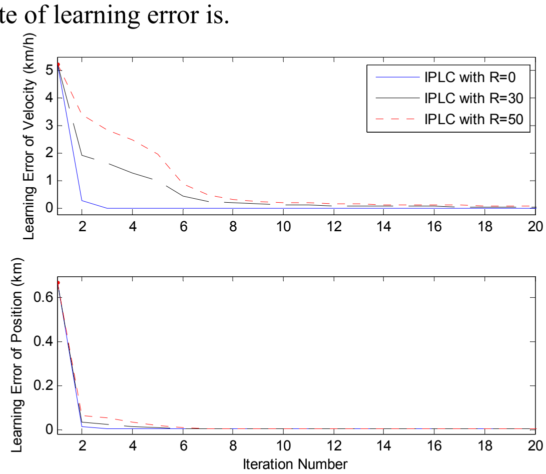

Fig. 5. Learning errors by IPLC with different $\mathbf{R}$ (Case III, IV, V)

Thirdly, three cases are provided to illustrate the function of weighting matrix $\mathbf{S}$. The predictive horizons are all 5 ($L=5$), and their weighting matrices $\mathbf{Q}$, $\mathbf{R}$ are set to be $\mathbf{Q}=\mathbf{I}$, $\mathbf{R}=5\cdot\mathbf{I}$. The different weighting matrices $\mathbf{S}$ here are: Case VI. $\mathbf{S}=\mathbf{0}$, Case VII. $\mathbf{S}=\mathbf{I}$, and Case VIII. $\mathbf{S}=2\cdot\mathbf{I}$.

From Fig. 6, it can also be observed that the learning errors converge along iteration axis by means of IPLC with different $\mathbf{S}$. In addition, the larger the weighting matrix $\mathbf{S}$ is, the larger the converged errors are. Note that in Fig. 7, the larger the weighting matrix $\mathbf{S}$ is, the less the energy cost is, which validates the function of weighting matrix $\mathbf{S}$ in the cost function design. By combining Fig. 6 and Fig. 7, it can conclude that the smaller the converged learning errors are, the more energy will be consumed, which indicates the compromise ought to be made between converged learning errors and energy consumption.

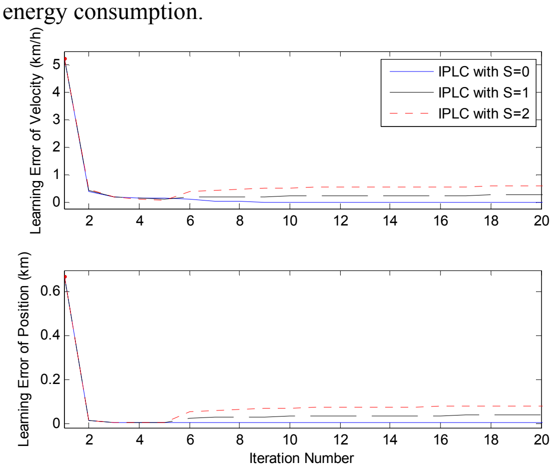

Fig. 6. Learning errors by IPLC with different $\mathbf{S}$ (case VI, VII, VIII)

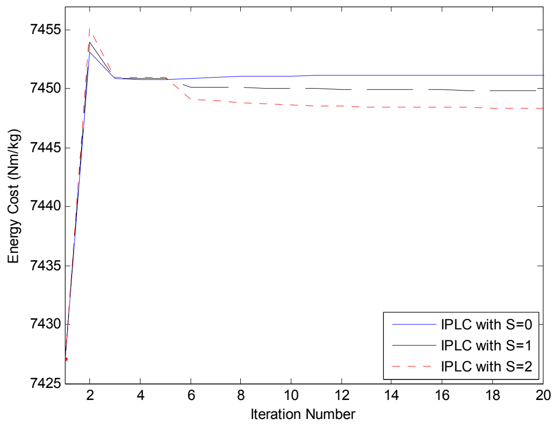

Fig. 7. Energy cost by IPLC with different **S** (case Ⅵ, Ⅶ, Ⅷ)

5. CONCLUSIONS

An approach of iterative predictive learning control (IPLC) is proposed for input-affine nonlinear systems. The application of IPLC to train trajectory tracking and energy efficient operation is further detailed by capturing and utilizing the repeatability of train motion dynamics. Moreover, when applying IPLC, convergence condition of the train motion system is solely determined by sample time, and other unavailable parameters will not affect its asymptotic convergence. Rigorous theoretic analyses and simulations show the feasibility of the proposed IPLC.

ACKNOWLEDGEMENT

This work was supported by International Cooperation Program (61120106009) of National Science Foundation, China.

REFERENCES

Amann, N. Owens, D.H. and Rogers, E. (1996). Iterative learning control for discrete-time systems with exponential rate of convergence. *IEE Proceedings Control Theory Applications*, **143(2)**, 217–224.

Ahn, H.S., Moore, K.L., and Chen, Y.Q. (2007). *Iterative learning control: robustness and monotonic convergence for interval systems*. Springer-Verlag, London .

Arimoto, S. Kawamura, S. and Miyazaki, F. (1984). Bettering operation of robots by learning. Journal of Robotics Systems, **1(2)**, 123-140.

Balaji, S., Fuxman, A., Lakshminarayanan, S., Forbes, J.F., and Hayes, R.E. (2007). Repetitive model predictive control of a reverse flow reactor. *Chemical Engineering Science*, **62(8)**, 2154-2167.

Bocharnikov, Y.V., Tobias, A.M., and Roberts, C. (2010). Reduction of train and net energy consumption using genetic algorithms for trajectory optimisation. *IET Conference on Railway Traction Systems*.

Bristow, D.A., Alleyne, A.G., and Zheng, D. (2004). Control of a microscale deposition robot using a new adaptive time-frequency filtered iterative learning control. *Proc. IEEE Amer. Control Conf.*, 5144-5149

Chang, C.S. and Xu, D.Y. (2000). Differential evolution based tuning of fuzzy automatic train operation for mass rapid transit system. *IEE Proc.-Electr. Power Appl.*, **147(3)**, 206-212.

Cueli, J.R. and Bordons, C. (2008). Iterative nonlinear model predictive control. stability, robustness and applications. *Control Engineering Practice*, **16(9)**, 1023-1034.

Davis, W.J. (1926). Traction resistance of electric locomotives and cars. *General Electric Review*, **29(10)**, 685-708.

Estima, J.O. and Marques, A.J. (2012). Efficiency analysis of drive train topologies applied to electric/hybrid vehicle. *IEEE Trans. Vehicular Technology*, **61(3)**, 1021-1031.

Hay, W.W. (1982). *Railroad Engineering, 2nd ed*. John Wiley and Sons, New York.

Howlett, P.G. and Pudney, P.J. (1995). *Energy-efficient train control*. Spring, Britain.

Hou, Z.S., Wang, Y., Yin, C.K., and Tang, T. (2011). Terminal iterative learning control based station stop control of a train. *Int. J. Control*, **84(7)**, 1263-1274.

Hou, Z.S., Xu, X., and Yan, J.W.. (2008). An iterative learning approach for density control of freeway traffic flow via ramp metering. *Transport. Res. Part C*, **16(1)**, 71-97.

Ke, B.R., Lin, C.L., and Lai, C.W. (2011). Optimization of train-speed trajectory and control for mass rapid transit systems. *Control Engineering Practice*, **19(7)**, 675-687.

Ke, B.R., Lin, C.L., and Yang, C.C. (2012). Optimisation of train energy-efficient operation for mass rapid transit systems. *IET Intelligent Transportation Systems*, **6(1)**, 58-66.

Lee, J.H. Lee, K.S. and Kim, W.C. (2000). Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica*, **36(5)**, 641–657

Murtaza, M.A. and Garg, S.B.L. (1989). Dynamic response of a railway vehicle air brake system. *Int. J. Vehicle Design*, **10(4)**, 481-496.

Shi, J., Gao, F.R., and Wu, T.J. (2007). Single-cycle and multi-cycle generalized 2D model predictive iterative learning control (2D-GPILC) schemes for batch processes. *Journal of Process Control*, **17(9)**, 715-727.

Sun, H.Q. and Hou, Z.S. (2013). An Iterative Predictive Learning Control Approach With Application to Train Trajectory Tracking. *Asian Control Confere*.

Sun, H.Q., Hou, Z.S., and Li, D. (2012). A norm optimal iterative learning approach based train trajectory tracking approach. *51st IEEE Conference on Descision and Control*, 3966-3971.

Sun, H.Q., Hou, Z.S., and Li, D. (2013). Coordinated iterative learning control schemes for train trajectory tracking with overspeed protection. *IEEE Trans. Automation Science and Engineering*, **10(2)**, 323-333.

Xu, J.X. and Tan, Y. (2003). *Linear and nonlinear iterative learning control*. Springer-Verlag, Berlin.

Yang, X., Li, X., Gao, Z., Wang, H., and Tang, T. (2013). A cooperative scheduling model for timetable optimization in subway systems. *IEEE Trans. Intelligent Transportation Systems*, **14(1)**, 438-447.

Zhuan, X. and Xia, X. (2008). Speed regulation with measured output feedback in the control of heavy haul trains. *Automatica*, **44(1)**, 242-247.