Iterative Learning Control for Constrained Linear Systems Using Successive Projection

David H. Owens and Bing Chu

Abstract—This paper considers iterative learning control for linear systems with convex control input constraints. First, the constrained ILC problem is formulated in a novel successive projection framework. Then, based on this projection method, a constrained ILC algorithm is proposed to solve this constrained ILC problem. The results show that, when perfect tracking is possible, the proposed algorithm can achieve perfect tracking. When perfect tracking is not possible, the algorithm can exhibit a form of practical convergence to a “best approximation”. The effect of weighting matrices on the performance of the algorithm is also discussed and finally, numerical simulations are given to demonstrate the effectiveness of the proposed method.

I. INTRODUCTION

Iterative learning control (ILC) is a control method for improving tracking performance of systems that execute the same task repeatedly by learning from the past actions. Applications of ILC can be widely found in industrial robot manipulator, chemical batch process, some medical equipment and manufacturing, etc. Originating from robotics, ILC now attracts more general research interest [1], [2].

In many practical applications, the systems are under some constraints due to physical limitations or performance requirements. Hence, the ILC design must take these constraints into account. However, most of the current ILC research is based on assumed unconstrained systems [1], [2] and few results have been reported regarding the constrained case in the literature. [3] proposes a novel nonlinear controller for process systems with input constraints and the learning scheme only needs a little knowledge of the process model. [4] considers ILC problem with soft constraints and uses Lagrange multiplier methods to solve this problem. [5] uses quadratic optimal design to formulate the constrained ILC problem and suggests quadratic optimal design has the capability of dealing with constraints. And more recently, [6] uses constrained convex optimization technique to solve the constrained ILC problem for linear systems with saturation constraints.

In this paper, ILC design problem with general convex input constraints is discussed. Full details and proofs can be found in [7] and an extended version of these results has been submitted to International Journal of Control. It is shown that the constrained ILC problem can be formulated in a recently developed successive projection framework of ILC [8], which provides an intuitive but rigorous method for system analysis and design. Based on this, a systematic approach for constraints handling is provided and a constrained ILC algorithm is proposed to solve this problem. The convergence analysis shows that when perfect tracking is possible, the proposed algorithm can achieve perfect tracking. When perfect tracking is not possible, the algorithm converges to asymptotic value representing a “best fit” solution. It is also found that the input and output weighting matrices have an interesting effect on the convergence properties of the algorithm.

The paper is organized as follows. In Section II, the constrained ILC problem is formulated. In Section III, the successive projection method is introduced and the constrained ILC problem is interpreted using this successive projection formulation. In Section IV, the constrained ILC algorithm is proposed and its convergence properties derived. In Section V, numerical simulations are presented to demonstrate the effectiveness of the proposed methods and finally, conclusions are given in Section VI.

II. PROBLEM FORMULATION

For simplicity, the formulation is described for linear discrete time systems but more generally applies to linear systems in Hilbert spaces described by equations of the form $y = Gu + d$ where $u$, $y$ are the system input and output respectively, $G$ is a bounded linear operator from an input Hilbert space to an output Hilbert space and $d$ represents other effects including the effect of initial state conditions. For more details see [9]. Note that the abstract formulation describes many situations of interest including continuous linear space model, discrete time model and differential delay model of system dynamics.

Consider the following discrete time, linear time-invariant system

$$x_k(t + 1) = Ax_k(t) + Bu_k(t)$$
$$y_k(t) = Cx_k(t),$$

(1)

where $t$ is the time index (i.e. sample number), $k$ is the iteration number and $u_k(t), x_k(t), y_k(t)$ are input, state and output of the system on iteration $k$. The initial condition $x_k(0) = x_0, k = 1, 2, \ldots$ is the same for all iterations. The control objective is to track a given reference signal $r(t)$ defined on a finite duration $t \in [0, N]$ (i.e. $t$ is the sample number for time series of length $N + 1$) and to do so by repeated execution of the task and data transfer from task to task. Mathematically, at the final time $t = N$, the state is reset to $x_0$ and time is reset to $t = 0$, a new iteration is started and, again, the system is required to track the same reference.
Before presenting the main results, the operator form of the dynamics is demonstrated using the well-known, so-called lifted-system representation, which provides a straightforward \(N \times N\) matrix approach in the analysis of discrete-time ILC [10], [11].

Assume, for simplicity, the relative degree of the system is unity (i.e. the generic condition \(CB \neq 0\) is satisfied), then system model (1) on the \(k^{th}\) iteration can be expressed in an equivalent form

\[
y_k = Gu_k + d,
\]

where \(G\) and \(d\) are the \(N \times N\) and \(N \times 1\) matrices

\[
G = \begin{bmatrix}
CB & 0 & \cdots & 0 \\
CAB & CB & \cdots & 0 \\
CA^2B & CAB & \cdots & \ddots \\
\vdots & \ddots & \ddots & \cdots & CB \\
CAN^{-1}B & \cdots & CB & 0
\end{bmatrix}
\]  

and \(k\) represents the iteration number. As the most important signal vector is the tracking error vector \(e = r - y\), then, without loss of generality, it can be assumed that \(d = 0\) by incorporating it into the reference signal (i.e. replacing \(r\) by \(r - d\)). Hence (2) becomes

\[
y_k = Gu_k,\]

where \(G\) is nonsingular and hence invertible.

The above representation of the original system (1) is called the lifted-system representation. This approach changes the original ILC problem into a MIMO tracking problem [10], [11]. Note that the above lifted-system form can be easily extended to situation when the system relative degree is larger than one. All the following discussions will be based on the lifted-system representation.

Tracking error improvements from iteration to iteration are achieved in ILC using the following general control updating law

\[
u_{k+1} = f(e_{k+1}, \ldots, e_{k-s}, u_k, \ldots, u_{k-r}),
\]

where \(e_k\) is the tracking error from the \(k^{th}\) trial/iteration and is defined as \(e_k = r - y_k\). When \(s > 0\) or \(r > 0\), (6) is called a high order updating law. This paper only considers algorithms of the form \(u_{k+1} = f(e_{k+1}, e_k, u_k)\). For higher order algorithms, please refer to [12], [13] and the references therein.

The ILC Algorithm Design Problem: The ILC algorithm design problem can now be stated as finding a control updating law (6) such that the system output has the asymptotic property that \(e_k \rightarrow 0\) as \(k \rightarrow \infty\).

There are many design methods to solve the ILC problem. The one used here is based on a quadratic (norm) optimal formulation [14] where, at each iteration, a performance index is minimized to obtain the system input time series vector to be used for that iteration. The basis of this paper is Norm-Optimal ILC (NOILC) which uses the following performance index

\[
J_{k+1}(u_{k+1}) = \|e_{k+1}\|^2_Q + \|u_{k+1} - u_k\|^2_R,
\]

minimized subject to the constraint that \(e_{k+1} = r - Gu_{k+1}\). \(G\) is the operator form of the system (1) and \(Q\) and \(R\) are positive definite weighting matrices. Also \(\| \cdot \|^2_R\) denotes the quadratic form \(e^T Q e\) and similarly with \(\| \cdot \|^2_Q\). Solving this optimization problem gives the following optimal choice for the time series vector \(u_{k+1}\)

\[
u_{k+1} = u_k + R^{-1}GTQe_{k+1}
\]

which, when \(k \rightarrow \infty\), asymptotically achieves perfect tracking. This well-known NOILC algorithm has many appealing properties including implementation in terms of Riccati state feedback. More details on NOILC can be found in [9], [14], [15], [16], [17].

In practical applications, system constraints are widely encountered. There are different kinds of constraints, e.g., input constraint, input rate constraint and state or output constraint. Constraints can be divided into two classes: hard constraints and soft constraints. Hard constraints are constraints on magnitude(s) at each point in time, for example, the output limits on actuators. Soft constraints are constraints that are applied to the whole function rather than its point-wise values e.g. constraints on total energy usage. The input constraints are often hard constraints. This paper only considers the input constraint. Suppose the input is constrained to be in a set \(\Omega\), which is taken to be a closed convex set in some Hilbert space \(H\). In practice, the set \(\Omega\) is often simple one. For example, the following constraints are often encountered:

\[
\Omega = \{ u \in H : |u(t)| \leq M(t) \}
\]

\[
\Omega = \{ u \in H : \lambda(t) \leq |u(t)| \leq \mu(t) \}
\]

\[
\Omega = \{ u \in H : 0 \leq u(t) \}
\]

If there are no constraints, the ILC design problem is relatively easy to solve and there are many design methods in the literature. However, when constraints are present, the problem becomes more complicated. The problem is to decide how to incorporate the constraints into the design process while retaining known performance properties. In the following sections, the successive projection method proposed by Owens and Jones in [18] is used to interpret iterative learning control, and a systematic approach for constraints handling in ILC is then proposed in the form of two new algorithms. The algorithms are related to but distinct from recently published work [8] where successive projection was used to accelerate norm optimal ILC.

III. INTERPRETATION OF ILC USING SUCCESSIVE PROJECTION

In this section, the concept of successive projection is summarised and its use in the ILC problem is demonstrated.
A. Successive Projection Method: An Overview

The successive projection method in the form described by Owens and Jones [18] is a technique for finding a point in the (assumed non-empty) intersection \( K_1 \cap K_2 \) of two closed, convex sets \( K_1 \) and \( K_2 \) in some real Hilbert space \( H \). The basic idea is to first select an initial iterate \( k_0 \) in \( H \). Subsequent points are obtained successively by projection of previous iterates onto one and then the other of the two convex sets. It is formally described in the following theorem.

**Theorem 1:** [18] Let \( K_1 \subset H, K_2 \subset H \), be two closed convex sets in a real Hilbert space \( H \) with \( K_1 \cap K_2 \) non-empty. Define

\[
K_j = \begin{cases} \{k_j \} & j \text{ odd} \\ K_2, & j \text{ even} \end{cases}
\]

Then, given the initial guess \( k_0 \in H \), the sequence \( \{k_j\}_{j \geq 0} \) satisfying

\[
\|k_j - k_{j-1}\| = \min_{k \in K_j} \|k - k_{j-1}\|, \quad j \geq 1 \tag{9}
\]

with \( k_j \in K_j, j \geq 1 \), is uniquely defined for each \( k_0 \in H \) and satisfies

\[
\|k_{j+1} - k_j\| \leq \|k_j - k_{j-1}\|, \quad j \geq 2. \tag{10}
\]

Furthermore, for any \( x \in K_1 \cap K_2 \),

\[
\|x - k_j\|^2 \geq \|x - k_{j+1}\|^2 + \|k_{j+1} - k_j\|^2 \tag{11}
\]

so that the sequence \( \{\|x - k_{j+1}\|\}_{j \geq 0} \) is monotonically decreasing and \( \{k_j\}_{j \geq 0} \) continuously gets closer to every point in \( K_1 \cap K_2 \). In addition

\[
\sum_{j=1}^{\infty} \|k_{j+1} - k_j\|^2 \leq \|x - k_1\|^2 \tag{12}
\]

so that, for each \( \epsilon > 0 \), there exists an integer \( N \) such that for \( j \geq N \)

\[
\inf_{k \in K_{j+1}} \|k - k_j\| < \epsilon. \tag{13}
\]

That is, the iterates \( k_j \in K_j \) become arbitrarily close to \( K_{j+1} \).

Moreover, when \( K_1 \cap K_2 \) is empty, the algorithm converges in the sense that \( \|k_{j+1} - k_j\| \to d(K_1, K_2) \) defining the minimum distance \( d(K_1, K_2) \) between the two sets \( K_1 \) and \( K_2 \).

The process is illustrated in Fig. 1(a) which indicates convergence schematically to a point in the intersection \( K_1 \cap K_2 \). In [18], this convergence is proved and a number of related and improved iterative schemes are presented. Here, the one related to our ILC results is used. For more details please see [18].

B. Interpretation of ILC with Input Constraints

Consider the ILC design problem initially without constraints. If the original system is injective, then for every achievable \( r(t) \), there exists a unique input \( u^*(t) \) such that

\[
r(t) = [Gu^*](t). \tag{14}
\]

The task of the ILC control law is to iteratively find a series of inputs such that \( u_k \to u^* \) as \( k \) tends to infinity. That is equivalent to iteratively finding the unique point \( (0, u^*) \in H = \mathbb{R}^N \times \mathbb{R}^N \) in the intersection of the following two sets in \( H \):

- \( S_1 = \{(e, u) \in H : e = r - Gu\} \)
- \( S_2 = \{(e, u) \in H : e = 0\} \)

The successive projection method then can be applied to generate an algorithm with the defined convergence properties. In general it is required to verify that these two sets are closed and convex in \( H \). This is trivially satisfied, for example, in finite dimensional time series spaces such as \( H = \mathbb{R}^N \times \mathbb{R}^N \). In this case, the inner product will be taken to be

\[
\langle (e, u), (z, v) \rangle = e^T Q z + u^T R v. \tag{14}
\]

with \( Q > 0, R > 0 \) symmetric positive definite and the associated induced norm will be \( \| (e, u) \| = \sqrt{\langle (e, u), (e, u) \rangle} \).

Then, using successive projection method in Theorem 1, the well-known NOILC algorithm can be easily derived [8].

---

(for more details, please refer to [8] which uses the concepts to successfully accelerate norm optimal ILC). It is shown that the convex constrained ILC problem can be formulated in the successive projection framework, the consequence of which is that a systematic approach for constraints handling is produced with known convergence properties. The notation in [18] is adopted in order to be consistent with the original paper and make the proof of our results more understandable. The notation \( r, k, t \) is also used elsewhere in the paper to denote other variables, parameters or signals. This should cause no confusion as their meaning can be inferred from the context.
which is illustrated geometrically in Fig. 1(b). Its convergence properties can also be easily derived. For more details, please refer to [8].

Now consider the constrained ILC problem discussed in Section 2. The problem is to find the intersection of the following closed, convex sets in $H = \mathbb{R}^N \times \mathbb{R}^N$:

- $S_1 = \{(e, u) \in H : e = r - Gu\}$
- $S_2 = \{(e, u) \in H : e = 0\}$

under the constraint $S_3 = \{(e, u) \in H : u \in \Omega\}$. Note that, it is normal that $S_1 \cap S_2 \cap S_3$ is either a singleton pair $(e, u) = (0, u^*)$ solving the ILC problem or it is the empty set $\emptyset$. In this second case, perfect tracking is not achievable due to the introducing of input constraint $\Omega$.

There are three sets in the constrained problem. It seems the results in Owens and Jones [18] can not be directly used. However, set $S_3$ can be associated with either $S_1$ (yielding two sets $S_1 \cap S_3$ and $S_2$) or $S_2$ (yielding two sets $S_2 \cap S_3$ and $S_1$) and also notice that the intersection of two closed convex sets is still a closed convex set. Then, the original 3-set problem becomes a 2-set problem, which is to find the intersection of $K_1 = S_1$ (resp.$S_1 \cap S_3$) and $K_2 = S_2 \cap S_3$ (resp.$S_2$).

The successive projection method in Section III-A hence generates two new iterative algorithms for the constrained ILC problem, which are demonstrated in the following two sections. In what follows, we do not specify the exact form of the constraints other than that they are closed and convex.

### IV. Constrained ILC Algorithm

To construct a constrained ILC algorithm, take $K_1 = S_1$ and $K_2 = S_2 \cap S_3$ to be the closed, convex sets in Theorem 1, which can be expressed as follows.

- $K_1 = \{(e, u) \in H : e = r - Gu\}$
- $K_2 = \{(e, u) \in H : e = 0, u \in \Omega\}$

The following constrained ILC algorithm can be constructed and is illustrated schematically in Fig. 2(a) and Fig. 2(b). Taking $K_1 = S_1 \cap S_3$ and $K_2 = S_2$ yields an alternative (see [7]).

#### A. Algorithm Description

**Algorithm 1:** Constrained ILC Algorithm Given any initial input $u_0$ satisfying the constraint with associated tracking error $e_0$, the input sequence $u_{k+1}, k = 0, 1, 2, \cdots$, defined by the solution of the input unconstrained NOILC optimization problem

$$u_k = \arg\min_u \left\{ ||r - Gu||_Q^2 + ||u - u_k||_R^2 \right\}$$ (15)

followed by the simple input projection step

$$u_{k+1} = \arg\min_{u \in \Omega} ||u - \tilde{u}_k|| \in \Omega$$ (16)

also satisfies the constraint and iteratively solves the constrained ILC problem.

**Remark 1:** Note that the projection step of Algorithm 1 requires the solution of the problem (16). It seems this may need the application of some optimization methods.

However, in practice the input constraint $\Omega$ is often a pointwise constraint and the solution of (16) can be computed easily. For example, when $\Omega = \{u \in H : |u(t)| \leq M(t)\}$, the solution is simply as follows,

$$u_{k+1}(t) = \begin{cases} M(t) & \tilde{u}_k(t) > M(t) \\ \tilde{u}_k(t) & |\tilde{u}_k(t)| \leq M(t) \\ -M(t) & \tilde{u}_k(t) < -M(t) \end{cases}$$ (17)

for $t = 0, \cdots, N-1$.

#### B. Convergence Analysis

This section discusses the convergence properties of Algorithm 1. As mentioned in Section III-B, due to the introducing of the input constraints $\Omega$, there may be no intersection of $S_1, S_2$ and $S_3$, which means perfect tracking of the reference signal may be not possible. In this case, the convergence properties may have some difference. Hence the results are presented in two parts: $S_1 \cap (S_2 \cap S_3) \neq \emptyset$ and $S_1 \cap (S_2 \cap S_3) = \emptyset$.

1) $S_1 \cap (S_2 \cap S_3) \neq \emptyset$: In this case, perfect tracking of the reference signal is possible with a unique input $u^*$. The theorem below directly follows from Theorem 1.

**Theorem 2:** When perfect tracking is possible, Algorithm 1 solves the ILC problem in the sense that

$$\lim_{k \to \infty} e_k = 0, \lim_{k \to \infty} u_k = u^*.$$ (18)

Moreover, this convergence is monotonic with respect to the following performance index,

$$J_k = ||E e_k||_Q^2 + ||Fe_k||_R^2$$ (19)
In Fig. 4, the tracking performance of Algorithm 1 is shown.

where

\[ e_k = r - Gu_k \]
\[ E = I - G \left( G^T Q G + R \right)^{-1} G^T Q \]
\[ F = \left( G^T Q G + R \right)^{-1} G^T Q \]

(20)

Algorithm 1 first computes the NOILC solution and then projects this solution onto the constraint. This approach is much simpler than the previously described algorithm in the sense that the computational load is much less and hence is a simpler way to implement successive projection in practice. Intuitively, this strategy may, however, lead to other problems such as a slower convergence rate.

Example 1: Consider the following system

\[ G(s) = \frac{4.2130s - 2.5164}{s^2 - 0.1312s + 3.6624}, \]

(21)

which is sampled using a zero-order hold and a sampling time of 0.1s. The trial length is 20s, zero initial conditions are assumed and the reference signal is generated by the square-wave input shown in Fig. 3. The constraint is \(|u(t)| \leq 1, t = 0, 1, \ldots\), which is satisfied by the input \(u^*\) so that perfect tracking is possible. The initial input is chosen to be \(u_0 = 0\). The simulation evaluates the performance of Algorithm 1 over 100 iterations. The weighting matrices are chosen to be \(Q = R = I\) for simplicity. The norm of the tracking error from 2th to 11th iteration is plotted and shown in Fig. 4.

From the figure, it is clear that Algorithm 1 may not produce monotonic convergence in the tracking error norm.

Although Algorithm 1 may not maintain monotonic convergence in the tracking error, it has the property that the distance between the \(k^{th}\) input and the optimal solution is decreasing monotonically, which is shown in the following theorem.

Theorem 3: When perfect tracking is possible, Algorithm 1 has the property that, for all \(k \geq 0\) and for all \(u_0\) and \(u^*\)

\[ \|u_{k+1} - u^*\| \leq \|u_k - u^*\|, \]

(22)
i.e., the input iterates approach the solution monotonically in norm.

2) \(S_1 \cap (S_2 \cap S_3) = 0\): In this case, perfect tracking is not possible and only an approximation of the original input \(u^*\) can be achieved. The following theorem describes algorithm behaviour.

Theorem 4: When perfect tracking is not possible, Algorithm 1 converges to point \(u^*_k\) which is uniquely defined by the following optimization problem,

\[ u^* = \arg \min_{u \in \Omega} \left\{ \|Ee\|^2_Q + \|Fe\|^2_R \right\}. \]

Moreover, this convergence is monotonic with respect to the following performance index,

\[ J_k = \|Ee_k\|^2_Q + \|Fe_k\|^2_R \]

(24)

where

\[ e = r - Gu \]
\[ E = I - G \left( G^T Q G + R \right)^{-1} G^T Q \]
\[ F = \left( G^T Q G + R \right)^{-1} G^T Q \]

(25)

Remark 2: For the constrained ILC problem, the best result we can achieve in terms of tracking error is defined by the following QP problem

\[ u^* = \arg \min_{u \in \Omega} \|r - Gu\|^2. \]

(26)

Compared to Theorem 4, it can be found that Algorithm 1 actually minimizes weighted norm of tracking error. In this case, only nearly optimal performance can be achieved.

C. Effect of Weighting Matrices Q and R

In this section, the effect of weighting matrices \(Q\) and \(R\) on the convergence properties of Algorithm 1 is discussed.

According to (15), the weighting matrices \(Q\) and \(R\) provide scaling on the tracking error and the change of input. Intuitively, if \(Q\) is fixed, then a smaller \(R\) implies larger

![Fig. 3. The input signal](image)

![Fig. 4. The tracking performance of Algorithm 1](image)
acceptable change of input, and which in turn, results in smaller tracking error. This leads to faster convergence rate.

Consider SISO systems with scalar weighing $Q$ and $R$. Choose $Q = 1$ and consider the effect of variation of $R$. When perfect tracking is possible, perfect tracking can be achieved and smaller $R$ will result in faster convergence. When perfect tracking is not possible, reducing $R$ will again result in faster convergence rate but the asymptotic error changes. This can be explained as follows. Algorithm 1 converges to the solution of the following problem

$$u^*_s = \arg \min_{u \in \Omega} \left\{ \| (I - G (G^T Q G + R)^{-1} G^T Q) e \|^2_Q + \| (G^T Q G + R)^{-1} G^T Q e \|^2_R \right\}$$

(27)

When $R \to \infty$, the first term of the last equation becomes $\|e\|^2_Q$ and the second term becomes zero. Hence, the optimization problem becomes

$$u^*_s = \arg \min_{u \in \Omega} \|e\|^2_Q$$

(28)

This is the constrained optimal solution and is the best result that can be achieved with constrained control. However, in this case, since the weighting of input change $R$ is very large, the convergence rate is expected to be very slow. On the other hand, when $R \to 0$, it can be seen the first term of (27) becomes zero and the second term becomes $\|G^{-1} e\|^2_R$, which can be further written as $\|u - u^*\|^2_R$, where $u^*$ is the unique input generating the reference signal. Hence, the optimization problem becomes

$$u^*_s = \arg \min_{u \in \Omega} \|u - u^*\|^2_R$$

(29)

This is just the projection of $u^*$ onto the constraint set $\Omega$. Clearly the tracking error may be larger than that of the constrained optimal solution. However, in this case, the convergence rate is fast.

From the discussion above, it can be seen that when perfect tracking is not possible, the weighting matrix $R$ provides a compromise between the convergence rate and the tracking performance. This is illustrated in the following example.

Example 2: Consider the following system

$$G(s) = \frac{s + 4}{s^2 + 5s + 6},$$

(30)

which is sampled using a zero-order hold and a sampling time of 0.1s. The trial length is 20s, zero initial conditions are assumed and the reference signal is generated by the sine-wave input shown in Fig. 5. The constraint set is defined by $|u(t)| \leq 0.8$, $t = 0, 1, \cdots$, which doesn’t contain the input $u^*$. The initial input is chosen to be $u_0 = 0$. Six simulations are shown with the weighting matrices $Q = I$ and $R = 3, 1, 0.1, 0.05, 0.01, 0.001$, respectively. The results are shown in Fig. 6 and Fig. 7.

From the figure, it can be seen that smaller $R$ results in faster convergence and the weighting matrix $R$ does have an effect on the asymptotic performance/accuracy with larger values of $R$ giving smaller asymptotic error norms. The asymptotic tracking error norm of Algorithm 1 against...
different weighting matrices $R$ is also plotted and shown in Fig. 8. Note that the lower horizontal line is the tracking error norm with the input (28) and the upper one is (29).

When the system is MIMO or the weighting matrices are not scalar, the effect of weighting matrices would not be so easy to analyze but a similar pattern could be expected.

V. NUMERICAL SIMULATION

In this section, two examples are given to demonstrate the effectiveness of the proposed methods. First, consider the following example where perfect tracking is achievable.

Example 3: Consider the following non-minimum phase system

$$G(s) = \frac{s - 4}{s^2 + 5s + 6},$$

which is sampled using a zero-order hold and a sampling time of 0.1s. The trial length is 20s, zero initial conditions are assumed and the reference signal is generated by the square-wave input shown in Fig. 3. The constraint is $|u(t)| \leq 1, t = 0, 1, \cdots$, which just contains the input $u^*$. The initial input is chose to be $u_0 = 0$. The simulation compares the NOILC, and the proposed constrained ILC Algorithm 1 with over 1000 iterations. For simplicity, the weighting matrices are chosen to be $Q = R = I$. The results are shown in Fig. 9 and Fig. 10.

Note that in this example, perfect tracking is possible. According to Theorem 2, perfect tracking can be achieved by the proposed algorithm. However, it is expected that the constraint will be active during the iterations, which means the resulting input of NOILC may violate the constraint.

From Fig. 9, it can be seen that Algorithm 1 is approaching perfect tracking, which verifies the previous expectations. During the first iterations, as $u_0 = 0$, $u_k$ increases in pointwise magnitude gradually and doesn’t violate the constraint in any of the two algorithms. In subsequent iterations, the input computed using NOILC then begins to violate the constraint and differences begin to emerge. It is interesting to see that Algorithm 1 outperforms NOILC at this stage.

The second example is to illustrate what will happen if perfect tracking is not possible.

Example 4: Consider the same non-minimum phase system

$$G(s) = \frac{s - 4}{s^2 + 5s + 6},$$

as Example 3. The reference signal is generated by the sine-wave input shown in Fig. 5. All the settings are exactly the same except the constraint is replaced by $|u(t)| \leq 0.8, t = 0, 1, \cdots$, so that perfect tracking is not possible. The results are shown in Fig. 11 and Fig. 12.

From Fig. 11, it can be seen that Algorithm 1 does not converge to the constrained optimal solution. Instead, further analysis on the data shows it converges to the solution of (23), which verifies Theorem 4. It can also be found that there is only a very slight difference between the asymptotic value and the constrained optimal solution. Fig. 12 shows the original input and resulting input of the three algorithms at the 200th iteration. It is noticed that the resulting final input of Algorithm 1 is not just putting saturation on the original input, instead, it adds some compensation, which is quite
interesting. Note that Algorithm 1 achieves nearly optimal performance using quite simple computation.

VI. CONCLUSION

Following the success of (unconstrained) norm-optimal iterative learning control, this paper discusses iterative learning control for linear systems with convex input constraints, a situation that approximates to situations met often in practice. First, the constrained ILC problem has been formulated in a novel successive projection framework. Then, based on this projection method, a constrained ILC algorithm has been proposed to solve the constrained ILC problem. It has been shown that, when perfect tracking is possible, the proposed algorithm can achieve perfect tracking. When perfect tracking is not possible, the algorithm has been shown to provide useful approximate solutions to the constrained ILC problem. The effect of weighting matrices on the performance of the algorithm has also been discussed and numerical simulations have been given to demonstrate their effectiveness.

Although the presentation has concentrated on sampled data systems (for reasons of both simplicity and practical relevance), the Hilbert space context of successive projection indicates that the ideas and results apply more widely and, in particular, to the case of continuous time systems with no change in the abstract form of the algorithms or results. The realization of these results will however change.

REFERENCES