2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Bumpless Transfer for a Flexible Adaptation of Iterative Learning Control

David J. Hoelzle, *Student Member*, Andrew G. Alleyne, *Senior Member*, and Amy J. Wagoner Johnson

***Abstract*— This work builds upon a framework for improving trajectory flexibility in systems controlled by Iterative Learning Control (ILC). Here we focus on positioning systems, decomposing a class of trajectories into motion primitives, termed *basis tasks*. The correct input signal for each *basis task* is identified in a training routine with ILC. The main development of this paper is a framework to intelligently apply these *basis task* specific input signals using an adaptation of bumpless transfer techniques. Bumpless transfer is reoriented to seamlessly transition between open-loop ILC signals without attenuating signal content away from the transition points. Experimental results display the effectiveness of the proposed approach on a serial planar positioning robot. Two conditions on *basis task* sequencing are tested. One which satisfies constraints imposed by previous work, and a relaxed trajectory constraint case designed to further explore trajectory flexibility. Bumpless transfer recovers some of the performance lost by constraint relaxation.**

I. INTRODUCTION

Iterative Learning Control (ILC) is an effective methodology for precision control for systems that track repeated trajectories [1]. Input signals generated by ILC are applied in open-loop or as a supplement to feedback control, typically achieving performances beyond what is capable by feedback alone. The ILC algorithm relies on trajectory repetition to learn an approximate inverse signal of the dynamics, disturbances, and unmodeled dynamics [2]. Therein lies a primary limitation of ILC. If the trajectory changes, the ILC algorithm must be reinitiated.

The targeted applications for this work are advanced manufacturing systems or material interrogation systems that require both high precision and process flexibility. Applications include micro-Robotic Deposition [3], electrohydrodynamic jet printing [4], and atomic force microscopy [5]. These systems are versatile in that they require no tooled setups, having little impedance from concept to product so that designs can be flexibly interchanged. However, the reliance of ILC on process repetition is not well aligned with these applications, inhibiting efficient implementation. The objective of this research is to explore adaptations of ILC that are as flexible as these applications.

The development of trajectory flexibility within the ILC framework has been stated previously as an objective in ILC research [1], [6]. Previous attempts at adding trajectory flexibility have de-emphasized the time specificity of ILC, instead focusing on learning a set of dynamics, and applying learned signals to a similar set of dynamics [6]–[10].

Recent work by the authors developed a very different adaptation of ILC from those listed above, maintaining the time specificity of ILC [3]. [3] focused on learning a set of tasks, termed *basis tasks*, through a training routine, and investigating effective methods to apply this *basis task* information to construct complex trajectories. This approach, termed the *basis task* approach to ILC (BTILC), orchestrates a tradeoff between trajectory flexibility and some performance degradation. For flexible manufacturing systems, this tradeoff may allow for improved trajectory tracking over feedback control alone, without sacrificing the innate flexibility of these systems.

This work builds on the BTILC framework. Importantly, it investigates a relaxed trajectory constraint where the *basis tasks* learned in a training routine do not have similar initial conditions when applied in a new trajectory. This constraint relaxation allows for a larger set of operation trajectories to be accomplished with the same training information, although tracking performance is lost. A novel bumpless transfer scheme for open-loop signals is presented that regains some lost performance.

The paper is organized as follows. BTILC definitions, implementation, and performance information is given in Section II. Section III introduces the bumpless transfer scheme. The experimental setup and results are given in Sections IV and V, respectively. Section VI gives concluding remarks and discusses future directions.

II. SYSTEM SETUP

This section briefly introduces BTILC. Explicit details can be found in [3].

A. System

Consider the single-input single-output (SISO) linear time-invariant (LTI) operator, H_d, represented in discrete-time state-space form in:

$$H_d \triangleq \begin{cases} x_d(k+1) = \mathbf{A_d}x_d(k) + \mathbf{B_d}u_d(k) \\ y_d(k) = \mathbf{C_d}x_d(k) + \mathbf{D_d}u_d(k) \end{cases}. \quad (1)$$

Manuscript submitted April, 6 2011. This work was supported in part by NSF Grant 09-00184 ARRA.

Andrew G. Alleyne is with the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (phone: 217-244-9993; fax: 217-244-6534; email: alleyne@illinois.edu).

David J. Hoelzle is with the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (email: hoelzle2@illinois.edu).

Amy J. Wagoner Johnson is with the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (email: ajwj@illinois.edu).

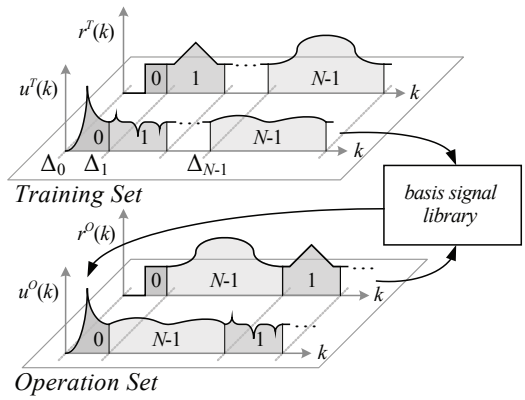

Fig. 1. *Visual depiction of BTILC. Basis tasks are learned in a training routine and the corresponding basis signal information is stored in the basis signal library. In the operation set, basis signals are applied as specified by logic applied to a schedule of basis tasks.*

$y_d(k) \in \mathbb{R}^K$, and $u_d(k) \in \mathbb{R}^K$ where K is the signal length. $\mathbf{A_d} \in \mathbb{R}^{\rho\times\rho}$, $\mathbf{B_d} \in \mathbb{R}^{\rho\times 1}$, $\mathbf{C_d} \in \mathbb{R}^{1\times\rho}$, and $\mathbf{D_d} \in \mathbb{R}^{1\times 1}$ are appropriately sized state-space system matrices and $x_d(k) \in \mathbb{R}^{\rho\times K}$ is a vector of the operator states where ρ is the number of states. H_d can be either an open-loop stable plant or the plant sensitivity function of a stabilizable plant with stabilizing feedback. That is, $\max_i \mid \lambda_i \mid < 1$ where λ_i is an eigenvalue of $\mathbf{A_d}$. k is the discrete-time index and t_s is the sample period. Subscript d denotes a specific operator. For the serial robot used in Section IV, d denotes either the x or y axes.

B. Definitions

Definition 1: Training, Operation, and Learning Sets. All signals belong to a specific set, Fig. 1. The *training set* corresponds to signals applied or measured during the training routine; denoted by the superscript T. The *operation set* corresponds to signals specific to the manufacturing operation; denoted by the superscript O. *Learning set* signals are analogous to the common implementation of ILC where ILC is applied at each trial; given the superscript $L(j)$ where j is the iteration index.

Definition 2: Basis Task $\triangleq$ $r_n(k)$. Each *basis task* is defined by the reference signal designed to complete the given task, $r_n(k)$, where subscript n is the task index. *Basis tasks* are defined on the domain $k \in [0, K_n - 1]$ where K_n is the *basis task* signal length.

Definition 3: R^O $\triangleq$ *Operation Space.* The *operation space* is the set of all *basis tasks* which constitute an operation. In general, a given operation is comprised of N *basis tasks*.

Definition 4: $y(k) \triangleq H_d\left(u(k)\right); e(k) \triangleq r(k) - y(k)$. Standard controls notation is used for the operator output, $y(k)$, input, $u(k)$, and error signal, $e(k)$. Input signals $u_n(k)$ corresponding to a *basis task* n are termed *basis signals*.

Definition 5: Adjacency Matrix. The sequence of *basis tasks* in either the *training set* or *operation set* is given by its binary *adjacency matrix*, $\mathbf{T} \in \mathbb{Z}_2^{N\times N}$ or $\mathbf{O} \in \mathbb{Z}_2^{N\times N}$, respectively. Given an entire trajectory $r(k) \in \mathbb{R}^K$, a matrix entry $c_{i,j} = 1$ if $\exists$ a $r(\Delta_{m_j} + K_j - 1)$ and $r(\Delta_{m_i})$ such that

$$r(k) = \{\ldots, \underbrace{r_j(K_j - 1)}_{r(\Delta_{m_j}+K_j-1)}, \underbrace{r_i(0)}_{r(\Delta_{m_i})}, \ldots\} \tag{2}$$

where Δ_{m_n} is the *basis task* transition time index at a given multiplicity of that task m_n. Else, $c_{i,j} = 0$. In words, $c_{i,j} = 1$ if there exists a transition from *basis task* j to i.

Definition 6: Set Equivalence. The *training set* and the *operation set* are equivalent if $\mathbf{T} + \mathbf{O} = \mathbf{T}$, where $+$ is the logical OR operator for matrices; $a_{i,j} + b_{i,j} = c_{i,j}$ for all $i = \{0, \ldots, N-1\}$ and $j = \{0, \ldots, N-1\}$. In words, every *basis task* transition in the *operation set* has been learned in the *training set*.

C. Basis Task Approach to ILC

1) Training Set: BTILC is given schematically in Fig. 1. Essentially, a training routine is selected such that all *basis tasks* are contained within the training routine.

$$r^T(k) = \sum_{n=0}^{N-1}\sum_{m_n=0}^{M_n-1} r_n(k - \Delta_{m_n}) \left(\mathfrak{s}(k - \Delta_{m_n}) - \mathfrak{s}(k - \Delta_{m_n} - K_n)\right) \tag{3}$$

where $\mathfrak{s}$ is the unit step function. There may be multiplicities of a given task, $m_n = 0, \ldots, M_n - 1$. The constraint

$$\Delta_{i+1} \geq \Delta_i + K_i \tag{4}$$

is applied so that *basis tasks* do not overlap in time. Here, $i = \{a, b, \ldots\}$ are ordered *basis task* indices in the *training set*. ILC, [1], is applied to system (1) using the reference signal in (3) until a desired performance is achieved. Given (4), the *basis tasks* and the corresponding *basis signals* are separated in time, (5).

$$u^T(k) = \sum_{n=0}^{N-1}\sum_{m_n=0}^{M_n-1} u_n(k - \Delta_{m_n}) \left(\mathfrak{s}(k - \Delta_{m_n}) - \mathfrak{s}(k - \Delta_{m_n} - K_n)\right) \tag{5}$$

Individual *basis signals*, $u_n(k)$, are stored in memory to be extracted during the *operation set*. The structured database for *basis signal* storage is termed the *basis signal library*.

2) Operation Set: Basis signals are extracted from the *basis signal library* based on logic applied to the set of instructions dictating the trajectory. *Basis signal* extraction is shown schematically in Fig. 1. Given an instructed *basis task*, the corresponding *basis signal* is applied to the plant with the appropriate time shift.

$$u^O(k) = \sum_{i=0} \mathbf{u_i}(k - \mathbf{\Delta_i}) \left(\mathfrak{s}(k - \mathbf{\Delta_i}) - \mathfrak{s}(k - \mathbf{\Delta_i} - \mathbf{K_i})\right) \tag{6}$$

where $\mathbf{u} = \{u_a(k), u_b(k), \ldots\}$ and $\mathbf{\Delta} = \{\Delta_a, \Delta_b, \ldots\}$ are appropriately designed ordered vectors of *basis signals* and

time shifts, respectively. Important considerations for *basis signal* extraction are given in [3].

D. Performance

Consider the *training set* $r^T(k)$. $r^T(k)$ is comprised of temporally sequenced *basis tasks*, (3), where each *basis task* is active over an interval starting at a task-transition index Δ_i. ILC is applied, converging to a $u^T(k)$ that gives the performance

$$e^T(k) = r^T(k) - H_d\left(u^T(k)\right) \tag{7}$$

where $u^T(k)$ and $e^T(k)$ can be decomposed temporally. At the task-transition indices, Δ_i, operator H_d has the following states, $\mathbf{X^T} = \{x_a^T, x_b^T, \ldots\}$, that effectively give the initial conditions for each task.

In the *operation set*, however, the sequence of *basis tasks* in $r^O(k)$ is different than the sequence in $r^T(k)$. The states at the transition indices, x_n^O, will be different because of historical sequencing differences or because there is not set equivalence, as described in *Definition 6*.

$$x_n^O = x_n^T + \delta x_n \tag{8}$$

where δx_n is the difference in states between the *training* and *operation sets* at each task transition. Therefore, there will be performance degradation because each $u^T(k)$ is identified for the *training set* transition states x_n^T not the *operation set* transition states x_n^O. The resultant error signal will be

$$e_n^O(k) = e_n^T(k) + \mathbf{C_d A_d}^k \delta x_n \tag{9}$$

which can be easily shown by properties of linear systems.

Remark: Performance degradation at the transition indices is discussed in greater detail in [3]. The salient point from this discussion is that performance degradation is minimized when there is set equivalence, as given in *Definition 6*. However, this condition may be restrictive in some cases. To weaken this condition, we can neglect set equivalence and only impose a trajectory smoothness condition

$$\begin{aligned} r_i^O(K_i - 1) &= r_{i+1}^O(0) \\ (q^{-1} - 1) r_i^O(K_i - 1) &= (1 - q) r_{i+1}^O(0) \end{aligned} \tag{10}$$

where $i = \{a, b, \ldots\}$ are consecutive *basis tasks* in the *operation set*, $\Delta_i + K_i = \Delta_{i+1}$, and q is the forward shift operator, $qx(k) = x(k+1)$. Consider the positioning system in Fig. 2 tracking an arbitrary trajectory that satisfies the weakened condition (10). The lack of set equivalence leads to performance degradation, Fig. 3, where the performance degrades immediately after the transition indices, approaching feedback control performance, $e_d^{L(0)}(k)$.

III. BUMPLESS TRANSFER

ILC produces a supplemental signal that is an inverse of the system dynamics, repeated disturbances, and uncertainties [2]. Smoothness constraints such as (10) do not account for higher order derivatives in the reference signal. Given that the ILC input signal achieves an inverse of the dynamics and

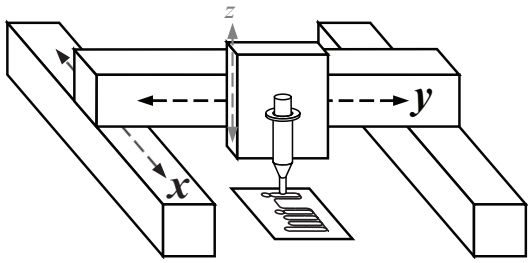

Fig. 2. *Diagram of a serial positioning system tracking an arbitrary trajectory that satisfies* (10)*. Highlighted are the x and y axes.*

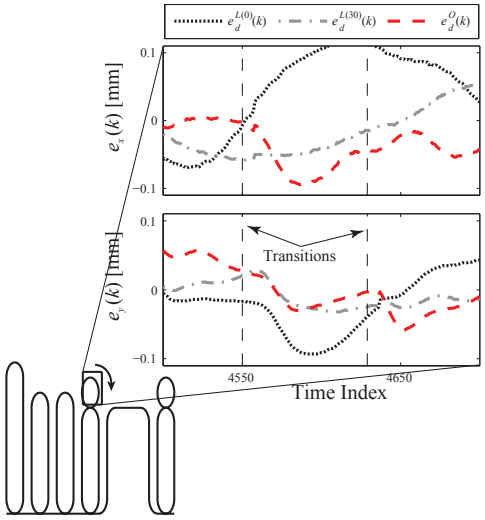

Fig. 3. *Axes error signals during a task transition without set equivalence. After basis task transition points, operation set performance, $e_d^O(k)$, transiently degrades, approaching feedback control performance, $e_d^{L(0)}(k)$.*

other factors and that *basis task* sequence affects transition states, x_n^O, there will be *basis signals*, $u_n^O(k)$, that are not appropriate for the instantaneous state conditions at the transitions. Consequently, performance will degrade.

Here, we borrow the concept of bumpless transfer from the feedback control community [11]. Systems with a switched control scheme will have transient performance degradation after a controller switch because the two controllers have different objectives and therefore different input magnitudes at the transition time. The idea in bumpless transfer is that transition performance can be improved by forcing the latent control signal to emulate the active signal at the instant of transition. After transition, control action converges to that of the new controller at a rate given by the bumpless transfer weighting filters.

We have modified the bumpless transfer algorithm for use on open-loop signals, Fig. 4. Prior to a *basis task* transition, each *basis signal* in the *operation set* is applied to a cascade of filters where the latent signal, $u_l(k)$, tracks the filtered input signal of the i^{th} *basis signal*. At the transition, switches s_2 and s_3 flip. The tunable parameter is the digital filter $b(q)/a(q)$ where the polynomials $a(q)$ and $b(q)$ are designed to reject a constant $u_{i+1}(0)$ and pass the signal $u(k)$ before the transition index Δ_{i+1}.

$$u_l(k) =$$

$$\begin{cases} \overbrace{\frac{a(q)}{a(q)+b(q)}}^{\text{reject}} u_{i+1}(0) + \overbrace{\frac{1}{a(q)+b(q)}}^{\text{pass}} u(k) \text{ for } k < \Delta_{i+1} \\ \frac{a(q)+b(q)}{a(q)+b(q)} u_\Delta(k) \text{ for } k \geq \Delta_{i+1} \end{cases} \quad (11)$$

where

$$u_\Delta(k) = \{u(\Delta_{i+1}-\delta), u(\Delta_{i+1}-\delta+1), \ldots, u(\Delta_{i+1}-1), u_{i+1}(0), \ldots\} \quad (12)$$

and δ is the order of the polynomial $a(q)+b(q)$.

Stability is guaranteed when the roots of polynomial $a(q)+b(q)$ are contained within the unit disk. Critically, this filter does not modify $u_{i+1}(k)$ at time indices away from Δ_{i+1}. That is

$$u(k) \to u_{i+1}(k) \text{ as } k \to (\Delta_{i+1} + K_{i+1}) . \quad (13)$$

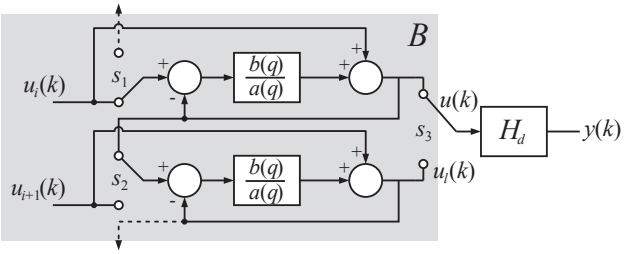

Fig. 4. *Section of a block diagram of the bumpless transfer filter $B(u(k))$ in a cascade of subfilters.*

IV. EXPERIMENTAL SETUP

A. System Description

BTILC with and without bumpless transfer is tested experimentally on a serial positioning system. The positioning system, shown in Fig. 2, is used to position a micro-extrusion system for fabricating micro-sized structures [3]. Dynamic models of each axes were identified in [12]. Parameter values for the axes plant models, H_d, and stabilizing feedback controllers, k_d, can be found in the Appendix. All results were acquired with a sampling rate, $1/t_s$, of $1kHz$.

$$H_d(z) = \frac{K(z+\alpha_1)(z^2+\alpha_2 z+\alpha_3)(z^2+\alpha_4 z+\alpha_5)}{(z+\beta_1)(z-1)(z^2+\beta_2 z+\beta_3)(z^2+\beta_4 z+\beta_5)} \quad (14)$$

$$k_d(z) = \frac{K(z^2+\alpha_1 z+\alpha_2)}{(z-1)(z+\beta_1)} \quad (15)$$

B. Training Set

The *training set* is designed to include linear motion primitives and circular motion (clockwise (CW) and counterclockwise (CCW)) primitives. Fig. 5 shows the designed training routine. The *training set* trajectory consists of six circuits around a perimeter in the same plane, $r_z(k) = 0$, with each circuit designed to identify different task information.

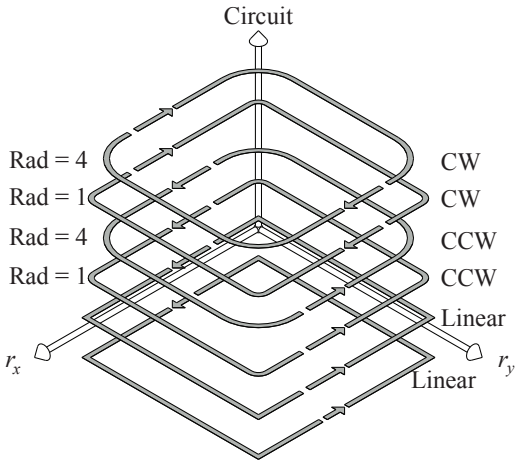

Fig. 5. *Training set. All six circuits are in the same plane with each circuit designed to identify different basis task information. Training set contains the basis tasks {linear, linear start, linear stop, CCW, CW, and dwell}.*

Descriptively, there are six *basis tasks* types in the *operation space*:

$$R^O = \{linear, linear\ start, linear\ stop, CCW, CW, dwell\} \quad (16)$$

where each type has classifiers, giving 49 unique *basis tasks*. Classifiers are given in Table I. The adjacency matrix for the *training set* is given in the Appendix, (20).

TABLE I
BASIS TASKS CLASSIFIERS IN TRAINING SET

Circuit	Type	Vel $[mm/s]$	Rad $[mm]$	Dir/Quad
1	Linear	10	0	+x,+y,-x,-y
2	Linear	20	0	+x,+y,-x,-y
3	CCW	20	1	3,4,1,2
4	CCW	20	4	3,4,1,2
5	CW	20	1	3,4,1,2
6	CW	20	4	3,4,1,2

ILC is applied to the systems given in (14) and (15) with the reference trajectory in Fig. 5 and with a standard update law, (17) [1].

$$u^{T(j+1)}(k) = Q(q)\left[u^{T(j)}(k) + L(q)e^{T(j)}(k+1)\right] \quad (17)$$

In (17), $Q(q)$ is a filter for robustness termed the Q-filter and $L(q)$ is the learning filter. $Q(q)$ is chosen to be a Gaussian filter and $L(q)$ a standard PD-type learning law, however, other filter choices are feasible. Filter details are provided in (21) and (22) in the Appendix. ILC is run for 30 iterations. The identified *basis signals* are considered to be the best input to achieve their respective *basis tasks*. Each *basis signal* is stored in the *basis signal library* for application in the *operation set*.

C. Operation Set

Four different *operation set* trajectories are tested. The two in the top row of Fig. 6, have an *operation set* that is equivalent to the *training set*: $\mathbf{T} + \mathbf{O} = \mathbf{T}$. The

two in the bottom row of Fig. 6, violate set equivalence, satisfying only the weakened constraint in (10). Namely, these trajectories contain direct $CCW \leftrightarrow CW$, $CCW \leftrightarrow CCW$, and $CW \leftrightarrow CW$ *basis task* transitions. Therefore, $\mathbf{O}(25:32, 25:32) \neq \mathbf{0}_{16\times16}$ and $\mathbf{T}+\mathbf{O} \neq \mathbf{T}$. A three way comparison is made for each *operation set* trajectory: (a) trajectory tracking for a typical ILC algorithm run to 30 iterations, $y^{L(30)}(k)$, (b) BTILC, $y^{O}(k)$, (c) and BTILC with bumpless transfer, $B\left(y^{O}(k)\right)$. The bumpless transfer filter is chosen heuristically

$$\frac{b(z)}{a(z)} = \frac{0.0117z + 0.0195}{z^2 - 1.819z + 0.8187}. \tag{18}$$

Three of the trajectories $\{(1,1),(1,2),(2,1)\}$ are motivated by the raster patterns used in micro-scale fabrication processes [4] or material interrogation processes [5]. $(2,2)$ is arbitrarily chosen to explore the flexibility limits of BTILC.

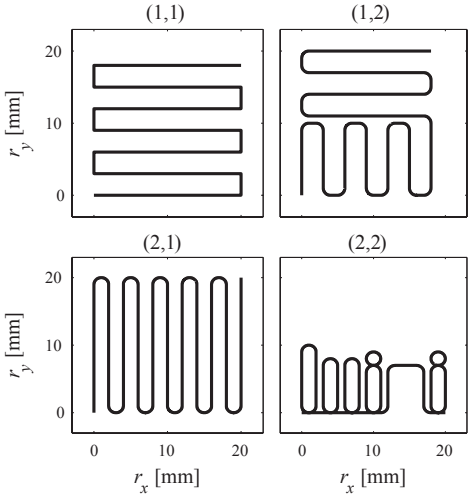

Fig. 6. *Operation set trajectories to test in simulation. Operation sets in the top row have set equivalence,* $\mathbf{T}+\mathbf{O}=\mathbf{T}$. *Operation sets in the bottom row do not have set equivalence,* $\mathbf{T}+\mathbf{O}\neq\mathbf{T}$.

V. EXPERIMENTAL RESULTS

The y-axis is capable of more accurately tracking reference trajectories than the x-axis because the x-axis has relatively more inertia. Consequently, error signals in the y-axis are smaller in magnitude and have higher signal-to-noise ratios. ILC is most effective in decreasing error in the x-axis and therefore this section will focus on x-axis results. y-axis results follow the same trends, however to a lesser degree. All BTILC data is an average of five trials with the same input, $u_d^O(k)$ or $B\left(u_d^O(k)\right)$.

A. Operation Set: $\mathbf{T}+\mathbf{O}=\mathbf{T}$

For *operations sets* chosen such that $\mathbf{T}+\mathbf{O}=\mathbf{T}$, the BTILC input signal, $u^O(k)$, is almost identical to the input signal from a typical implementation of ILC, $u^{L(30)}(k)$, Fig. 7. This is because the transition sequencing has been maintained and therefore the state dissimilarities at the transitions, δx_n in (8), are minimal. Fig. 10 displays that the RMS, (19), of the entire error signal degrades less than 21% in the x-axis

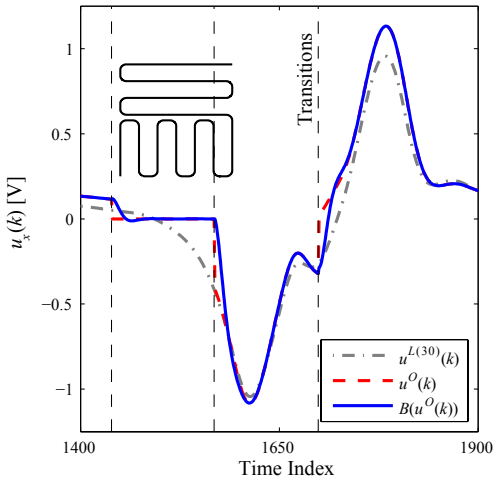

Fig. 7. *Operation set basis task sequences that have set equivalence have smooth transitions. Bumpless transfer filtering has only a small affect at transitions.*

by using BTILC for control. Details are found in Table II. In fact, BTILC has a RMS error tracking performance better than RMS $\left(e^{L(10)}(k)\right)$ on average for the x-axis.

$$\text{RMS}\left(e(k)\right) = \sqrt{\sum_K e^2(k)/K} \tag{19}$$

Since the state dissimilarities are small, (8), the transitions between *basis signals* are smooth and therefore bumpless transfer modifies the input signal minimally. Fig. 7 demonstrates that bumpless transfer input signal, $B\left(u^O(k)\right)$, is almost identical to the *operation set* input signal, $u^O(k)$. Bumpless transfer reduces RMS of the x-axis error by 4%, Table III.

B. Operation Set: $\mathbf{T}+\mathbf{O}\neq\mathbf{T}$

When there is not set equivalence, $\mathbf{T}+\mathbf{O}\neq\mathbf{T}$, the state dissimilarity, (8), is large. Therefore, the identified *basis signals* are not properly designed for the given *basis task* states, x_n^O. This result is demonstrated in Fig. 8 in which the ideal ILC signal, $u^{L(30)}(k)$, has a considerably different signal shape than the concatenated *basis signals*, $u^O(k)$. Bumpless transfer helps bridge the gap between *basis signals*, better approximating $u^{L(30)}(k)$ at the transition points. The influence of bumpless transfer can be seen in Fig. 9 where the tracking error for $B\left(e^O(k)\right)$ is comparatively less than $e^O(k)$; whereas BTILC approaches feedback performance following transition indices, BTILC with bumpless transfer performs significantly better than feedback. Tracking performance does degrade, as given by Table II and Fig. 10, however bumpless transfer does regain some performance losses. Applying bumpless transfer to BTILC gains a performance improvement of over 12% on average in the x-axis, Table III, in terms of the RMS error. Even with $\mathbf{T}+\mathbf{O}\neq\mathbf{T}$, Fig. 10 shows that bumpless transfer provides tracking performance that is better than RMS $\left(e^{L(14)}(k)\right)$ on average, in the x-axis.

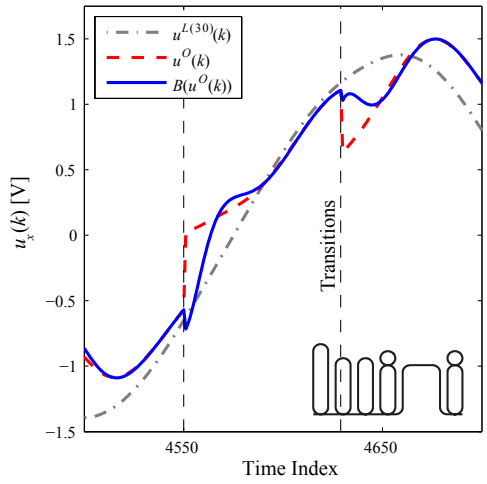

Fig. 8. *Operation set basis task sequences without set equivalence,* $\mathbf{T}+\mathbf{O} \neq \mathbf{T}$*, have discontinuous transitions. Bumpless transfer filtering modifies transitions, smoothing out input signal and better approximating the ILC input signal,* $u^{L(30)}(k)$.

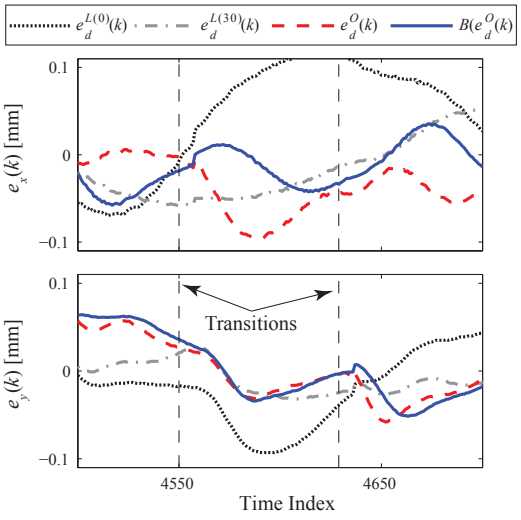

Fig. 9. *Error signal comparison. Compared to BTILC,* $e_d^O(k)$*, applying bumpless transfer for BTILC reduces error transients at the task transitions.*

TABLE II

COMPARISON: $\frac{\text{RMS}\left(B\left(e_d^O(k)\right)\right)-\text{RMS}\left(e_d^{L(30)}(k)\right)}{\text{RMS}\left(e_d^{L(30)}(k)\right)} \times 100\%$

	$\mathbf{T}+\mathbf{O}=\mathbf{T}$		$\mathbf{T}+\mathbf{O}\neq\mathbf{T}$	
Axis	(1,1)	(1,2)	(2,1)	(2,2)
x	19.34	22.15	57.83	30.58
y	79.55	41.61	-9.52	48.46

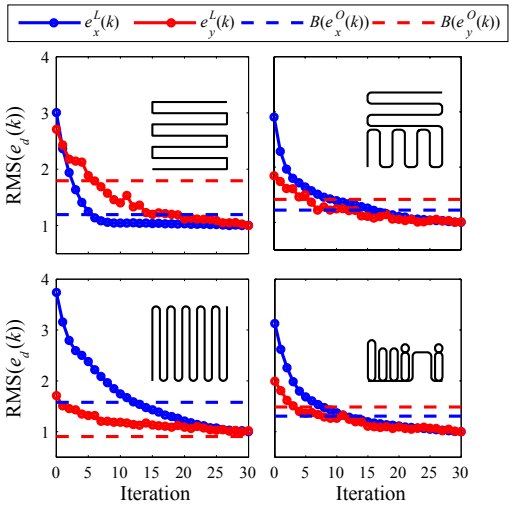

Fig. 10. *RMS error for each operation set simulated. All data is normalized such that* $\min_j \left(\text{RMS}\left(e_d^{L(j)}(k)\right)\right) = 1$*.* $B\left(e_d^O(k)\right)$ *data represents mean performance for 5 trials. This data spans the Iteration axis for the sake of comparison.*

TABLE III

COMPARISON: $\frac{\text{RMS}\left(e_d^O(k)\right)-\text{RMS}\left(B\left(e_d^O(k)\right)\right)}{\text{RMS}\left(e_d^O(k)\right)} \times 100\%$

	$\mathbf{T}+\mathbf{O}=\mathbf{T}$		$\mathbf{T}+\mathbf{O}\neq\mathbf{T}$	
Axis	(1,1)	(1,2)	(2,1)	(2,2)
x	0.37	7.64	5.03	19.59
y	2.82	1.24	2.69	-3.84

C. Discussion

The results display a marked improvement over feedback control. Depending on the application performance requirements, BTILC presents a viable option for control improvement for operation sets that have weakened trajectory constraints. Furthermore, performance can be improved at instances where there is a *basis task* transition that is not equivalent to the *training set* by applying bumpless transfer. The results display how BTILC is capable of significant tracking performance improvements over feedback with a wide variety of *operation set* reference signals without retraining the system for a new trajectory. In both axes, $\text{RMS}\left(B\left(e_d^O(k)\right)\right) < \text{RMS}\left(e_d^{L(11)}(k)\right)$ on average; therefore 11 iterations of tracking performance is achievable just by intelligently using information already on-hand.

The design of the bumpless transfer filter $b(z)/a(z)$ presents a tradeoff between a fast response with minimal signal attenuation and a slow response so that large signal discontinuities can be bridged. A key example of this tradeoff can be seen in Figs. 7 and 8. The fast filter response ensures a smooth transition with quick convergence to the subsequent *basis signal* when $\mathbf{T}+\mathbf{O}=\mathbf{T}$. However, the system reacts too quickly when $\mathbf{T}+\mathbf{O}\neq\mathbf{T}$ and does not fully attenuate the transient. Future work will investigate this tradeoff and filters designed to account for transition characteristics.

VI. CONCLUSION

Previous work has shown that the *basis task* approach to ILC is an effective method of improving tracking performance over typical feedback schemes, while critically alleviating the trajectory invariance constraint of ILC. This works builds on this effort, further enhancing trajectory flexibility by regaining performance with bumpless signal transfer. Current results show that bumpless transfer modifies

the supplementary input signal so that it better approximates signals identified by standard ILC algorithms. However, there is a trade-off between achieving seamless *basis signal* transitions and attenuating important signal characteristics. Future work will explore more advanced bumpless transfer algorithms that better balance the trade-offs of the filter.

APPENDIX

A. Plant and Controller Parameters

Plant	Parameter				
Den	β_1	β_2	β_3	β_4	β_5
H_x	-0.9994	-1.978	0.9894	-1.738	0.8672
H_y	-0.9994	-1.983	0.9911	-1.87	0.9539
k_x	-0.7408				
k_y	-0.7408				
Num	α_1	α_2	α_3	α_4	α_5
H_x	0.9604	-1.981	0.9918	-1.874	0.9747
H_y	1	-1.983	0.9912	-1.873	0.9547
k_x	-1.941	0.9423			
k_y	-1.949	0.9506			
Gain	K				
H_x	8.3315×10^{-4}				
H_y	1.8506×10^{-3}				
k_x	38				
k_y	27.375				

B. Adjacency Matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{8\times 8} & \mathbf{0} & \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{4\times 4} & \mathbf{I}_{4\times 4} \end{bmatrix} & \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{4\times 4}^{\perp} & \mathbf{I}_{4\times 4}^{\perp} \end{bmatrix} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{8\times 1} \\ \mathbf{I}_{8\times 8} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \begin{bmatrix} \mathbf{0} & \mathbf{I}_{4\times 4}^{\perp} \\ \mathbf{0} & \mathbf{I}_{4\times 4}^{\perp} \end{bmatrix} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \begin{bmatrix} \mathbf{0} & \mathbf{I}_{4\times 4} \\ \mathbf{0} & \mathbf{I}_{4\times 4} \end{bmatrix} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{1\times 8} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (20)$$

where $\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{\perp} = \begin{bmatrix} b & a \\ d & c \end{bmatrix}$

C. ILC Learning Filters

$$L_d\left(e_d(k)\right) = k_{P,d} e_d(k) + k_{D,d}\left(e_d(k) - e_d(k-1)\right) \quad (21)$$

$$Q_d\left(x(k), k^*\right) = \frac{1}{\sum_{i=0}^{L} e^{-\frac{(t_s(k^*-i))^2}{2\sigma^2}}} \sum_{i=0}^{L} x(k^*-i) e^{-\frac{(t_s(k^*-i))^2}{2\sigma^2}} \quad (22)$$

d	k_P	k_D	σ
x-axis	1.2	25	1.02×10^{-2}
y-axis	1	10	7.36×10^{-3}

REFERENCES

[1] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of Iterative Learning Control," *Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[2] H.-S. Ahn, Y. Chen, and K. L. Moore, "Iterative Learning Control: Brief survey and catagorization," *IEEE Transactions on Systems, Man, and Cyberkinetics - Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1099–1121, 2007.

[3] D. Hoelzle, A. Alleyne, and A. Wagoner Johnson, "Basis task approach to Iterative Learning Control with applications to micro-Robotic Deposition," *IEEE Transactions on Control Systems Technology*, In Press, DOI: 10.1109/TCST.2010.2063030, 2011.

[4] S. Mishra, K. Barton, A. Alleyne, P. Ferreira, and J. Rogers, "High speed drop-on-demand printing with a pulsed electrohydrodynamic jet," *Journal of Micromechanics Microengineering*, vol. 20, no. 9, pp. 1–8, 2010.

[5] Y. Wu and Q. Zou, "An iterative based feedforward-feedback control approach to high-speed afm imaging," in *Proc. of the IEEE American Control Conference*, St. Louis, MO, June 2009, pp. 1658–1663.

[6] R. Horowitz, "Learning control of robot manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 115, pp. 402–411, 1993.

[7] S. Kawamura and N. Sakagami, "Analysis of dynamics of underwater robot manipulators basing on Iterative Learning Control and time-scale transformation," in *IEEE International Conference on Robotics & Automation*, Washington, DC, USA, May 2002, pp. 1088–1094.

[8] C. T. Freeman, Z. Cai, P. L. Lewin, and E. Rogers, "Objective-driven ILC for point-to-point movement tasks," in *Proc. of the IEEE American Control Conference*, St. Louis, MO, USA, June 2009, pp. 252–257.

[9] M. Sekimoto, S. Kawamura, and T. Ishitsubo, "Basis-motion torque composition approach: Generation of motions with different velocity profiles among joints," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 18-22 2010, pp. 670–676.

[10] D. Gorinevsky, D. E. Torfs, and A. Goldenberg, "Learning approximation of feedfoward control dependence on the task parameters with application to direct-drive manipulator tracking," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 4, pp. 567–581, 1997.

[11] S. Graebe and A. L. Ahlén, "Dynamic transfer among alternative controllers and its relation to antiwindup controller design," *IEEE Transactions on Control Systems Technology*, vol. 4, no. 1, pp. 92–99, 1996.

[12] D. Bristow and A. Alleyne, "A high precision motion control system with application to microscale robotic deposition," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1008–1020, 2006.