2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Design of an iterative learning control with a selective learning strategy for swinging up a pendulum

Jonas Beuchert[1], Jörg Raisch[1], and Thomas Seel[1]

Abstract— **Swinging up a pendulum on a cart is a well-known demonstration example for trajectory tracking in a nonlinear system. The standard realtime feedback control approach fails if the plant output is not available in real time, e.g. due to large or variable measurement delays. However, the task can be solved in multiple trials by applying feedforward inputs that are improved from trial to trial by Iterative Learning Control (ILC). Our examination demonstrates that an ILC can be used for trajectory tracking close to the singularities and the unstable equilibrium of a non-linear system. Specifically, we present an ILC algorithm for pendulum swing-up by angle trajectory tracking. The controller design is based on a modified plant inversion approach that restricts the learning process to trajectory segments with small tracking errors and sufficient input sensitivity. We show that these restrictions lead to improved learning progress in contrast to conventional learning from the complete trajectory. Controller performance is evaluated in an experimental testbed. The ILC starts from a zero-input trajectory and learns to swing up the pendulum within six iterations. Robustness is analyzed experimentally, and the performance is compared to literature results. The convergence is at least two orders of magnitude faster than the one achieved by other methods that avoid feedback and do not rely on a suitable initial input trajectory.**

I. INTRODUCTION

An electromechanical system consisting of a horizontal rail of finite length with a one-dimensionally movable cart thereon is provided. A pendulum rod is rotatably mounted on the cart. The aim is to transfer the rod from its downward position into its upright position by moving the cart on the rail. The cart acceleration is limited, and the finite length of the rail limits the cart movements.

The pendulum is a popular demonstration object in control engineering. There are already a large number of algorithms which solve the swing-up control task. The majority is based on real-time feedback control of the pendulum angle, e.g. on energy-based controllers [1]. However, these approaches are unfeasible if large variable time delays (jitters) occur during measurement data acquisition, as may be the case in wireless information transmission.

These problems can be avoided by considering a controller that learns to swing up the pendulum during several iterations. During each iteration, it operates as a feedforward controller. Subsequently, an optimized input trajectory for the following iteration is determined from the recorded measurement values.

Several previous contributions considered learning algorithms for swinging up a pendulum. The proposed design strategies can basically be partitioned into two groups. There are, on the one hand, algorithms from the field of machine learning and game theory [2], [3], [4], [5], [6] and, on the other hand, iterative learning controllers (ILC) [7], [8], [9], which can be clearly assigned to the discipline of control engineering.

In the area of machine learning, the most frequently used approach is reinforcement learning. It has got a significant advantage: No or just a little plant knowledge is necessary for the algorithms. This leads to great robustness with respect to model uncertainties. However, the number of required iterations is usually in the three- to six-digit [3] range and each step is computationally expensive in comparison to an ILC update step. Moreover, these algorithms are based on stochastics and therefore, do not always lead to success (e.g. 80 % in Guez et al. [4]).

In the field of iterative learning control, different design methodologies have been proposed for the inverted pendulum. On the one hand, Sathiyavathi et al. [7] and Amann et al. [8] present hybrid approaches consisting of an ILC and a real-time feedback controller, which lead to a swing-up within a small single-digit number of iterations but imply the above mentioned limitations of immediate feedback control. Schöllig and D'Andrea [9], on the other hand, rely on an ILC without realtime feedback. They propose a combination of conventional filtering and optimization using convex programming. However, this approach is computationally expensive and requires an initial input trajectory that is close enough to the ideal successful one.

The goal of the present contribution is to propose an ILC method that requires neither a good initial input trajectory nor computationally expensive calculations. Nevertheless, the pendulum rod should pass the singular points of the system, i.e. the points at which the input has no effect on the output, and reach its unstable upper equlibrium within a small number of iterations. We refrain from using realtime feedback control of the pendulum angle and aim at good robustness against parameter uncertainties.

II. ITERATIVE LEARNING CONTROL

ILC has been deeply studied in both theory and application during the last two decades [10], [11], [12], [13], [14]. It has proven to be a useful method for applications in electromechanical and process engineering engineering [7], [8], [9], [15], as well as in biomedical engineering [16], [17], [18], [19], [20].

In the current contribution, we formulate the ILC problem in the discrete-time systems framework, as explained for

[1]all authors are with Control Systems Group, Technische Universität Berlin, 10587 Berlin, Germany, Seel@control.TU-Berlin.de

example in [21]. The plant to be controlled is a single-input single-output system with input signal $u_j[k] \in \mathbb{R}$ and output signal $y_j[k] \in \mathbb{R}$, where $k = 1, 2, ..., n$ is the time index for every trial $j = 1, 2, ...$

A desired trajectory $y_\mathrm{d}[k], k = 1, ..., n$, is given as a reference, and an initial input trajectory $u_1[k], k = 1, ..., n$, is chosen. This input trajectory is applied to the system as a feedforward control signal and the output $y_1[k], k = 1, ..., n$, is measured. Between the first and the second trial, the tracking error

$$e_1[k] = y_\mathrm{d}[k] - y_1[k], k = 1, ..., n,$$

is determined and used to calculate the updated input signal $u_2[k]$, which is applied as a feedforward input during the second trial, and so on. Whether the tracking error decreases in some sense depends on the learning law that is used to update the input. A commonly used learning law [21] is the following:

$$u_{j+1}[k] = Q(q^{-1}) \left(u_j[k] + L(q^{-1})e_j[k]\right), \tag{1}$$

where $k = 1, ..., n$, and $L(q^{-1})$ and $Q(q^{-1})$ are potentially non-causal discrete-time filters, and q^{-1} is the back-shift operator.

Generally, for the L-filter, many approaches are possible, e.g. simple proportional gains (P-type), PD-type controllers, or plant inversion. In any case, it executes the main task of the learning process. The Q-filter is often set to identity, in which case (1) simplifies to $u_{j+1}[k] = \left(u_j[k] + L(q^{-1})e_j[k]\right)$. In most other cases, $Q(q^{-1})$ is chosen to be a low-pass to improve robustness of the closed-loop system by preventing learning on high frequencies.

The basic theory of ILC relies on the assumption that the system dynamics do not change from trial to trial and that the system is reset to the same initial condition prior to every trial. Moreover, every trial must have the same duration, and neither saturation nor measurement noise should occur [21]. In recent contributions, substantial efforts have been made towards relaxing these fundamental assumptions of ILC theory. In particular, adaptations for iteration-variant initial conditions have been proposed [22], and an extended framework for systems with variable trial duration has been developed [23] and validated in practice [24], [25], [17].

III. SYSTEM REPRESENTATION

The experimental setup is schematically shown in Figure 1. Measurements of the horizontal cart position $u \in \mathbb{R}$, the angle of the pendulum rod $y \in \mathbb{R}$, and the associated angular velocity $\dot{y} \in \mathbb{R}$ are available. The cart is moved by a conveyor belt and a powerful motor with an underlying very fast control loop that allows us to set the angular position of the motor in realtime. Hence, the input signal of the system to be controlled is the cart position $u \in \mathbb{R}$.

The cart position u, velocity $\dot{u}$, and acceleration $\ddot{u}$ of the system are bounded:

$$|u| \leqslant u_{\max}, \quad |\dot{u}| \leqslant \dot{u}_{\max}, \quad |\ddot{u}| \leqslant \ddot{u}_{\max}. \tag{2}$$

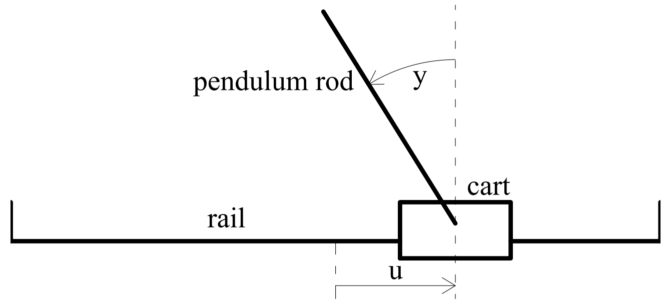

Fig. 1. Sketch of the hardware setup consisting of a pendulum attached to a cart that can move horizontally on a rail. The pendulum angle is defined to be zero when the pendulum rod is in the upper (unstable) equilibrium.

The dynamic relation between cart acceleration $\ddot{u}$ and pendulum angle y can be expressed by the nonlinear second-order differential equation

$$\ddot{y} = c_1(c_2(\ddot{u} \cdot \cos(y) + g \cdot \sin(y)) - c \cdot \dot{y}), \tag{3}$$
$$c_1 = \frac{1}{J_S + m \cdot a^2}, \quad c_2 = m \cdot a,$$

with parameter values given in Table I.

TABLE I

PARAMETERS OF THE PENDULUM SYSTEM TO BE CONTROLLED

Mass pendulum rod	m	0.3475 kg
Distance axis of rotation - center of gravity	a	0.459 m
Coefficient of friction	c	0.007 Nm s
Moment of inertia w.r.t. center of gravity	J_S	0.034 Nm s^2
Maximum cart position	$u_{\max}$	0.75 m
Maximum cart velocity	$\dot{u}_{\max}$	3.5 m/s
Maximum cart acceleration	$\ddot{u}_{\max}$	40 m/s^2

IV. CONTROLLER DESIGN

The pendulum angle should track a predefined angle trajectory y_d, which transfers the pendulum rod from the vertically downwards position at the start ($y[k = 1] = \pi$) to the upright position at the end ($y[k = n] \mod 2\pi = 0$). We assume that such a reference trajectory is known and that it is feasible, i.e. there exists an unknown input trajectory that yields the output trajectory $y = y_\mathrm{d}$.

The nonlinear differential equation (3) is linearized at a general linearization point $(y_S, \dot{y}_S, \ddot{u}_S) \in \mathbb{R}^3$ (which is not necessarily an equilibrium), i.e.

$$0 = c_1(c_2(\ddot{u}_S \cdot \cos(y_S) + g \cdot \sin(y_S)) - c \cdot \dot{y}_S). \tag{4}$$

This results in a linear differential equation with coefficients $\alpha_0, \alpha_1, \beta_2 \in \mathbb{R}$ that depend on the linearization point:

$$\ddot{y} = -\alpha_0(y - y_S) - \alpha_1(\dot{y} - \dot{y}_S) + \beta_2(\ddot{u} - \ddot{u}_S), \tag{5}$$

with

$$\alpha_0 = -c_1c_2(-\ddot{u}_S \cdot \sin(y_S) + g \cdot \cos(y_S)), \tag{6}$$
$$\alpha_1 = c_1c, \tag{7}$$
$$\beta_2 = c_1c_2 \cdot \cos(y_S). \tag{8}$$

These continuous-time dynamics are discretized at a finite number of time instants $k = 1, 2, \ldots, n$ along an input trajectory $u[k]$ and a corresponding output trajectory $y[k]$. The sampling period $\Delta = 0.002\,\mathrm{s}$ is chosen very small to assure that changes of $\alpha_0, \alpha_1, \beta_2$ from one discrete time instant to the next (and second-next) are negligible. More specifically, the Tustin approximation (bilinear transformation) without frequency prewarping is applied to establish the following discrete-time plant model of the input-output dynamics:

$$\begin{aligned}&(1 + a_{1,k} \cdot q^{-1} + a_{2,k} \cdot q^{-2}) y[k] \\ &\qquad = (b_{0,k} + b_{1,k} \cdot q^{-1} + b_{2,k} \cdot q^{-2}) u[k], \end{aligned} \tag{9}$$

where

$$a_{1,k} = \frac{-8 + 2 \cdot \alpha_{0,k} \cdot \Delta^2}{4 + 2 \cdot \alpha_{1,k} \cdot \Delta + \alpha_{0,k} \cdot \Delta^2}, \tag{10}$$

$$a_{2,k} = \frac{4 - 2 \cdot \alpha_{1,k} \cdot \Delta + \alpha_{0,k} \cdot \Delta^2}{4 + 2 \cdot \alpha_{1,k} \cdot \Delta + \alpha_{0,k} \cdot \Delta^2}, \tag{11}$$

$$b_{0,k} = b_{2,k} = \frac{4 \cdot \beta_{2,k}}{4 + 2 \cdot \alpha_{1,k} \cdot \Delta + \alpha_{0,k} \cdot \Delta^2}, \tag{12}$$

$$b_{1,k} = \frac{-8 \cdot \beta_{2,k}}{4 + 2 \cdot \alpha_{1,k} \cdot \Delta + \alpha_{0,k} \cdot \Delta^2}, \tag{13}$$

where the additional index k in $\alpha_{0,k}$, $\alpha_{1,k}$, and $\beta_{2,k}$ indicates that (6), (7), and (8) are evaluated at $u[k]$ and $y[k]$ for each k and that the back-shift operator in (9) acts only on $u[k]$ and $y[k]$.

After every ILC trial j, the measured input and output trajectory samples $u_j[k], y_j[k]$, $k = 1, 2, \ldots, n$, are used as linearization points to determine the trial- and time-dependent parameter values $a_{2,k,j}, a_{1,k,j}, b_{2,k,j}, b_{1,k,j}, b_{0,k,j}$. The L-filter is chosen such that it inverts the dynamics (9) to calculate the input trajectory for the $(j+1)$th trial based on the measured tracking error of the jth trial:

$$\begin{aligned}&(1 + a_{1,k} \cdot q^{-1} + a_{2,k} \cdot q^{-2}) e_j[k] \\ &\quad = (b_{0,k} + b_{1,k} \cdot q^{-1} + b_{2,k} \cdot q^{-2})(u_{j+1}[k] - u_j[k]),\end{aligned}$$

which leads to the input update law

$$\begin{aligned} u_{j+1}[k] = u_j[k] &+ A_{j,k} \cdot (u_j[k-1] - u_{j+1}[k-1]) \\ &+ B_{j,k} \cdot (u_j[k-2] - u_{j+1}[k-2]) \\ &+ C_{j,k} \cdot e_j[k] + D_{j,k} \cdot e_j[k-1] \\ &+ E_{j,k} \cdot e_j[k-2], \end{aligned} \tag{14}$$

with

$$A_{j,k} = \frac{b_{1,k,j}}{b_{0,k,j}}, \quad B_{j,k} = \frac{b_{2,k,j}}{b_{0,k,j}}, \quad C_{j,k} = \frac{1}{b_{0,k,j}}, \tag{15}$$

$$D_{j,k} = \frac{a_{1,k,j}}{b_{0,k,j}}, \quad E_{j,k} = \frac{a_{2,k,j}}{b_{0,k,j}}. \tag{16}$$

At the beginning of each trial, the cart is placed in the middle of the rail and the pendulum is resting in its lower equilibrium, i.e. $u_j[-1], u_j[0], e_j[-1], e_j[0], u_{j+1}[-1]$, and $u_{j+1}[0]$ are all zero.

A. *Overcoming singularities in plant inversion*

For a linear system and under ideal circumstances, the inversion-based approach leads to one-step convergence [21]. During the second iteration, the desired trajectory should be reached, i.e. $y_2[k] = y_\mathrm{d}[k] \ \forall\, k \in [1, n]$, regardless of the initial input sequence u_1. However, in the present case, there is a major limitation: When the pendulum rod is close to horizontal ($y_j[k] \bmod \pi \approx \frac{\pi}{2}$), the input (cart velocity) hardly influences the output (pendulum angle) of the system. Therefore, the coefficients C, D, and E become arbitrarily large. Since large changes in the cart velocity lead to very small changes in the pendulum angle, the inversion-based approach reacts to small tracking errors with very large input signal changes.

To avoid this problem, we would like to prevent learning from these trajectory segments by forcing the coefficients C, D, and E of the learning law to zero near the singularities. This is achieved approximately by using the following heuristic modification of the coefficients in the input update law (14):

$$\tilde{C}_{j,k} = \operatorname{sgn}\left(\frac{1}{b_{0,k,j}}\right) \cdot \min\left(\gamma v_C, \left|\frac{1}{b_{0,k,j}}\right|\right), \tag{17}$$

$$\tilde{D}_{j,k} = \operatorname{sgn}\left(\frac{a_{1,k,j}}{b_{0,k,j}}\right) \cdot \min\left(\gamma v_D, \left|\frac{a_{1,k,j}}{b_{0,k,j}}\right|\right), \tag{18}$$

$$\tilde{E}_{j,k} = \operatorname{sgn}\left(\frac{a_{2,k,j}}{b_{0,k,j}}\right) \cdot \min\left(\gamma v_E, \left|\frac{a_{2,k,j}}{b_{0,k,j}}\right|\right), \tag{19}$$

$$\gamma := \left(\frac{|(y_j[k] \bmod \pi) - \frac{\pi}{2}|}{d}\right)^p, \tag{20}$$

where $v_C, v_D, v_E, d, p \in \mathbb{R}^+$ are positive constants. The parameter d determines how far away from the singular points the coefficients are attenuated. Parameters v_C, v_D, v_E and p influence slope and shape of the transitions between the limited and the unmodified segments. We set $v_C = v_E = 20\,\mathrm{m/rad}$, $v_D = 40\,\mathrm{m/rad}$, $d = \pi/4\,\mathrm{rad}$, $p = 9$. Figure 2 shows an example for the coefficient $\tilde{C}$.

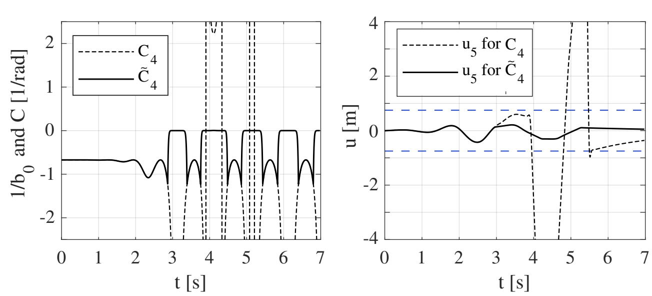

Fig. 2. *Left:* Error amplification factor for the fourth iteration of an experiment. Without singularity treatment, C_4 becomes very large. *Right:* These large values lead to an excessively changing input signal u_5 that exceeds $\pm u_{\max}$ (– – –). This is not the case if the modified $\tilde{C}_4$ is used.

B. *Learning only from small tracking errors*

The proposed learning law is based on linearization of the highly nonlinear plant dynamics. If an observed tracking error is small, the ILC should intuitively react with a small

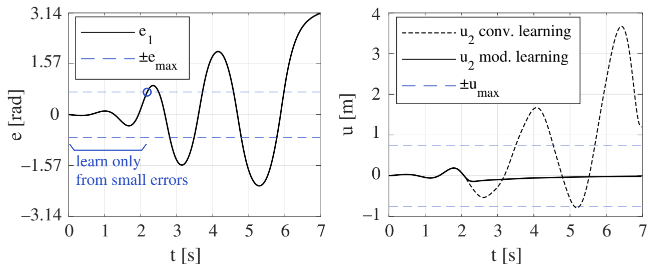

Fig. 3. *Left:* Measured error $e_1[k]$ during first iteration with $u_1[k] = 0\,\forall k$. *Right:* Input trajectory $u_2[k]$ as computed with conventional learning (considering all measurements) and with the modified learning approach that considers only small tracking errors. If the learning is confined to the first $2.2\,\mathrm{s}$, the input signal remains within its constraint $u_{\max}$.

change of the input trajectory that compensates the measured error. However, if the tracking error and the input changes are large, the linearization is no longer assured to be a good approximation of the true system dynamics. Learning from large tracking errors might therefore result in further increase of these tracking errors and to unnecessarily large cart velocities and positions.

Therefore, we restrict the learning process to learn only from samples with small tracking error. More specifically, for each trial j, we determine the smallest index k at which the absolute value $|e_j[k]|$ of the tracking error exceeds a threshold $e_{\max} > 0$. Denote this index by $k_{e,j}$. After every trial, the input update (1) is performed only for input indices $k \leqslant k_{e,j}$. For all larger time indices, the cart position u_{j+1} is set to decline to zero exponentially with the time constant $\tau \in \mathbb{R}_{>0}$:

$$u_{j+1}[k] = u_{j+1}[k_{e,j}] \cdot \exp\left(-\frac{(k-k_{e,j})\Delta}{\tau}\right) \forall k > k_{e,j}. \tag{21}$$

Note that the tracking error is zero at the beginning of each trial. Therefore, $k_{e,j}$ is nonzero for all trials. Assuming that an update improves the tracking at least on the first $k_{e,j}$ samples, we may expect that $k_{e,j}$ becomes larger and larger until the tracking error is smaller than $e_{\max} > 0$ for the entire trial.

These coupled dynamics of trial duration and tracking error have been studied before in [24]. There, however, the variable trial duration was a result of output constraint violations, whereas in the current contribution we deliberately choose to limit the learning to trajectory segments with small tracking errors. Another remarkable difference between the previous and the current study is that the pendulum swingup requires a trajectory that passes singular points and approaches an unstable equilibrium, both of which is not the case in the crane application.

Figure 3 illustrates the selective learning strategy by experimental results for a scenario with zero initial input and output signal, i.e. $u_1[k] = 0$ and $y_1[k] = \pi \forall k$. The large tracking errors of the first trial lead to a large input trajectory u_2 that exceeds the saturation limits, unless the proposed learning strategy is employed.

However, even in the theoretical case of an unlimited rail, it is advantageous to learn only from small tracking errors. This is demonstrated by experimental results in Figure 4. The conventional learning law that considers the complete measurement information leads to input trajectories with very large values. In contrast, the proposed strategy avoids large input changes and yields very small tracking errors in less than eight iterations. The root-mean-square of the tracking error reaches the level of $0.1\,\mathrm{rad}$ quickly and remains that low.

Whether the modified learning approach leads to successful pendulum swing-up in the experimental testbed will be investigated in Section V. Prior to this, we shall note that unmodeled higher frequencies might appear in the measurements and accumulate from trial to trial. Therefore, we introduce a Q-filter into the learning law (1). A second-order Butterworth lowpass filter is discretized using the Tustin approximation with frequency prewarping at the cutoff frequency f_0. It is applied to a unit-impulse sequence ($s_i(0) = 1, s_i(k) = 0 \ \forall k \neq 0$) first in forward direction and then backward in time to obtain a smoothed impulse response $r_i(k)$ with zero phase shift. The sample values $r_i(k), k = -n, \ldots, n$, of this response are the coefficients of the Q-filter polynomial (cf. [26]):

$$Q(q^{-1}) = r_i(-n)q^{-n} + \ldots + r_i(n)q^n. \tag{22}$$

This completes the controller design.

V. EXPERIMENTAL EVALUATION

The proposed iterative learning control consisting of (7), (8), (9), (10), (11), and the Q-filter is now evaluated in an experimental testbed. The ILC is used to update the input trajectories that are repeatedly applied to the cart-and-pendulum system as feedforward inputs. Control design parameters are chosen as follows:

$$\begin{aligned} &e_{\max} = 0.7\,\mathrm{rad}, \quad \tau = 2\,\mathrm{s}, \quad v_C = v_E = 20\,\mathrm{m/rad}, \\ &v_D = 40\,\mathrm{m/rad}, \quad d = \pi/4\,\mathrm{rad}, \quad p = 9, \quad f_0 = 2\,\mathrm{Hz}. \end{aligned}$$

A desired output trajectory is chosen that leads to swing-up in about seven seconds. The initial input and output signal are constantly zero, i.e. the ILC must learn the entire swing-up from scratch. See [27] for an animated visualisation.

Figure 5 shows input and output signals for the first six trials. It is obvious to see that the first input update (yielding u_2) used only the first $2.5\,\mathrm{s}$ of the measured error, while the second update (yielding u_3) was based on the first $3\,\mathrm{s}$, and so on. After five learning steps, the measured curve of the angle y is close to the desired y_{d} for all times. The pendulum swing-up is successful.

Figure 6 shows the development of the root-mean-square error e_{RMS} over the iteration index for two experiments with identical initial conditions. Note that the tracking error remains small even after trial 6.

These results demonstrate that the proposed iterative learning control scheme leads to successful swing-up after very few learning steps if the system parameters and dynamics

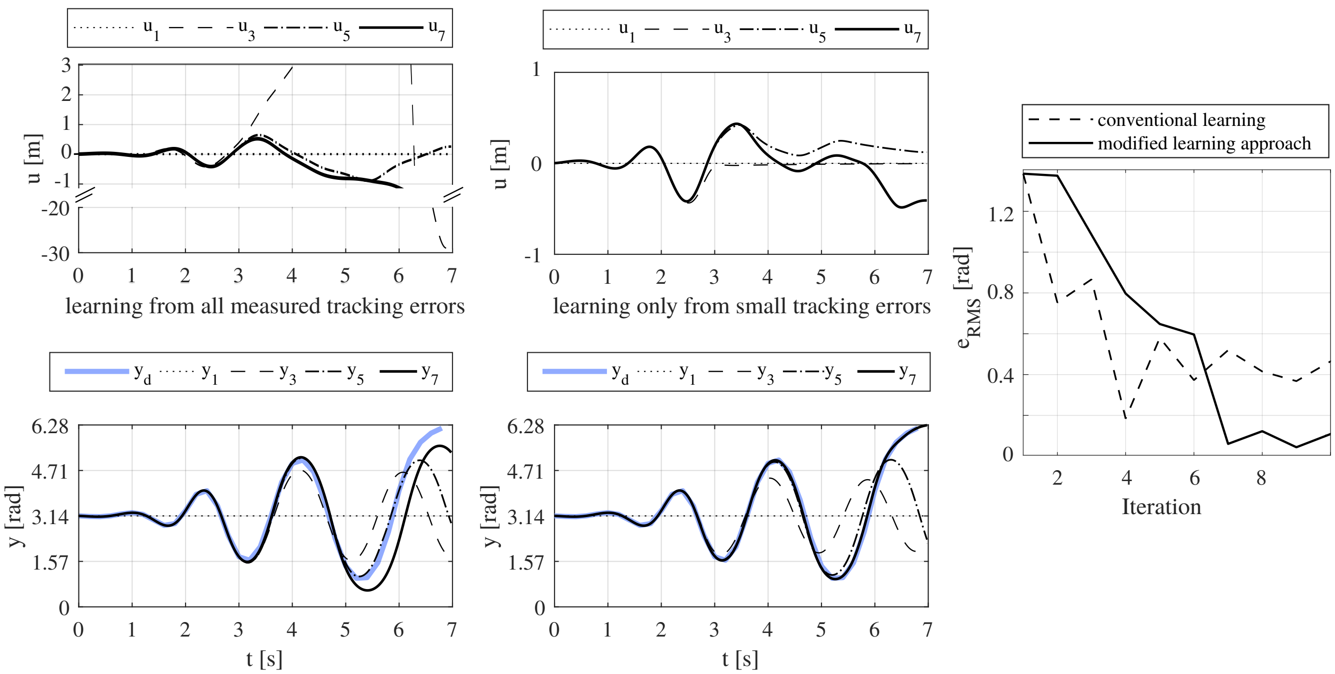

Fig. 4. Simulations of ILC with and without selective learning strategy. By learning only from small errors, the ILC is able to maintain the input within its bounds and to achieve smaller tracking errors.

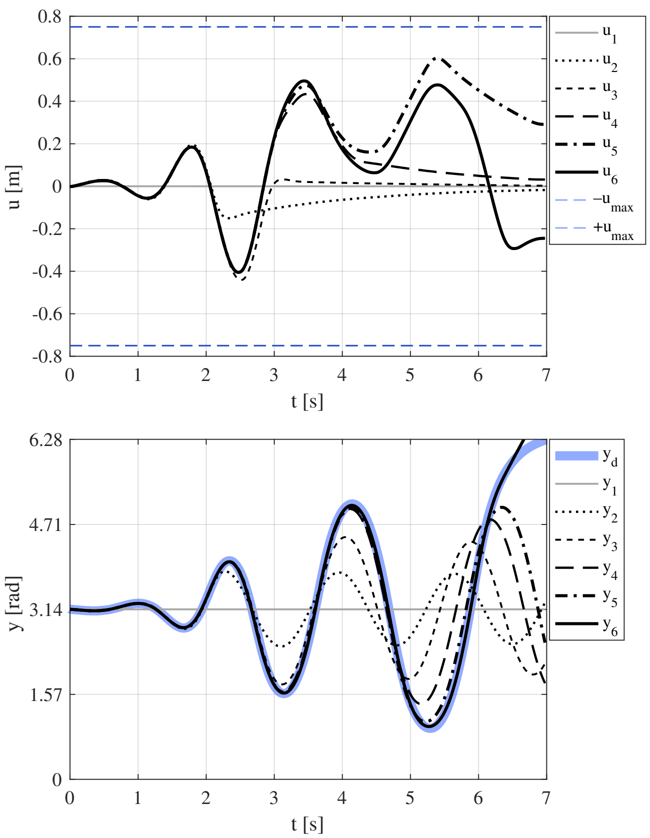

Fig. 5. Applied input trajectories (top) and desired and measured output trajectories (bottom) during iterations one to six.

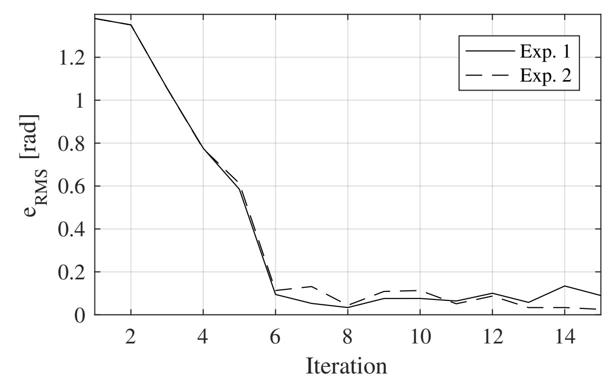

Fig. 6. Development of the root mean square error for two experiments.

are exactly known. To investigate the robustness of the algorithm, the same experimental testbed is used again, but the algorithm is parameterized with approximately 50 different parameter sets that differ from the true parameter values.

An experiment is called successful if the root mean square tracking error falls below $0.1\,\mathrm{rad} \approx 5.73°$ during the first 15 iterations. Table II indicates the extent to which a single parameter can be varied independently of the others in relation to its nominal value (cf. Table I) in order to enable a successful swing-up process. The smallest range of robustness is found for the distance between the axis of rotation and the center of gravity, which may be half of the true value but not more than 10% larger than the true value. All other model parameters may differ from their true values

TABLE II

ROBUSTNESS RANGES OF MODEL PARAMETERS IN THE ILC DESIGN.

Parameter		Nominal	Tolerance
Mass pendulum rod	m	$0.3475\,\mathrm{kg}$	50–300 %
Distance rotation axis to c. o. g.	a	$0.459\,\mathrm{m}$	50–110 %
Coefficient of friction	c	$0.007\,\mathrm{N\,m\,s}$	0–2000 %
Moment of inertia	J_S	$0.034\,\mathrm{N\,m\,s^2}$	20–200 %

by at least 50% into either direction.

Although the proposed approach is based on plant inversion, it is not sensitive to parameter uncertainties. All model parameters may vary by at least a factor 2 (i.e. between 50% and 200 %) from the true value, except for the length of the pendulum, which should be between 50 and 110 % of the true value.

VI. CONCLUSION

We proposed an iterative learning control method for the swing-up of a pendulum rod. The desired trajectory goes from the stable to the unstable equilibrium and passes singular points, in which the input's effect on the output vanishes, repeatedly. The standard inversion-based approach was modified by restricting the learning to small errors and to angles that are far from the singularities. We demonstrated that it can be advantageous to confine the learning process to small tracking errors if the controller design is based on linearization along a trajectory of a nonlinear system.

Unlike previous machine learning or ILC approaches, the proposed algorithm starts with a zero input and learns to swing up the pendulum in very few trials, while it neither requires realtime feedback of the pendulum angle nor computationally expensive optimizations. For the given experimental testbed, the proposed ILC algorithm achieves successful swing-up in five learning steps for the nominal parameter values and in no more than fourteen learning steps for severely modified parameter values.

However, a few limitations remain. As soon as the cart reaches an end of the rail, successful learning cannot be guaranteed anymore. Future research may aim at assuring convergence despite input saturation or constraint violation. Beyond that, we will theoretically analyze the influence of the parameter choices (especially $e_{\max} > 0$) on the tracking error and aim at establishing robust convergence criteria.

REFERENCES

[1] K. J. Åström and K. Furuta, Swinging Up a Pendulum by Energy Control, *13th International Federation of Automatic Control (IFAC) World Congress*, San Francisco, 1996.

[2] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra, Continuous control with deep reinforcement learning, *International Conference on Learning Representations*, San Juan, 2016.

[3] M. Watter, J. T. Springenberg, J. Boedecker and M. Riedmiller, Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images, *Computing Research Repository*, 2015.

[4] A. Guez, N. Heess, D. Silver and P. Dayan, Bayes-Adaptive Simulation-based Search with Value Function Approximation, *Advances in Neural Information Processing Systems*, Montral, 2014.

[5] N. Heess, J. J. Hunt, T. P. Lillicrap and D. Silver, Memory-based control with recurrent neural networks, *Computing Research Repository*, 2015.

[6] Y. Pan, K. Bakshi and E. A. Theodorou, A Cooperative Game Theoretic Approach to Trajectory Optimization, *Reinforcement Learning and Decision Making*, Edmonton, 2015.

[7] S. Sathiyavathi, K. Krishnamurthy, P. K. Bhaba and S. Somasundaram, Hybrid iterative learning control design: Application to inverted pendulum, *International Journal of Simulation - Systems, Science and Technology*, 2013.

[8] N. Amann, D. H. Owens and E. Rogers, Iterative learning control using optimal feedback and feedforward actions, *International Journal of Contro*l, pp. 277-293, 1996.

[9] A. Schöllig and R. D'Andrea, Optimization-Based Iterative Learning Control for Trajectory Tracking, *Proceedings of the European Control Conference*, Budapest, 2009.

[10] E. Rogers, K. Galkowski, D.H. Owens, Control Systems Theory and Applications for Linear Repetitive Processes, *Springer (Lecture Notes in Control and Information Sciences, 349)*, 2007.

[11] W. Paszke, E. Rogers, K. Galkowski, On the Design of ILC Schemes for Finite Frequency Range Tracking Specifications, *Proceedings of th IEEE Conference on Decision & Control*, 2010.

[12] K.L. Moore, M. Johnson, M.J. Grimble, Iterative Learning Control for Deterministic Systems, *Springer (Advaces in Industrial Control)*, 1993.

[13] M. Norrlöf, S. Gunnarsson, Time and frequency domain convergence properties in iterative learning control, *International Journal of Control*, 75(14):1114–1126, 2002.

[14] W. Paszke, E. Rogers, K. Galkowski, Experimentally verified generalized KYP Lemma based iterative learning control design, *Control Engineering Practice*, 53:57–67, 2016.

[15] H.S. Ahn, Y.Q. Chen, K.L. Moore, Iterative learning control: brief survey and categorization 1998–2004, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1099–1121, 2007.

[16] C.T. Freeman, A.-M. Hughes, J.H. Burridge, P.H. Chappell, P.L. Lewin, E. Rogers, Iterative learning control of FES applied to the upper extremity for rehabilitation, *Control Engineering Practice*, 17(3):368-381, 2009.

[17] T. Seel, C. Werner, T. Schauer, The Adaptive Drop Foot Stimulator – Multivariable Learning Control of Foot Pitch and Roll Motion in Paretic Gait, *Medical Engineering Physics*, 38(11):1205–1213,2016.

[18] Y. Wang, E. Dassau, F.J. Doyle III, Closed-loop control of artificial pancreatic-cell in type 1 diabetes mellitus using model predictive iterative learning control, *IEEE Transactions on Biomedical Engineering*, 57(2):211-219, 2010.

[19] T. Seel, S. Weber, K. Affeld, T. Schauer, Iterative Learning Cascade Control of Continuous Noninvasive Blood Pressure Measurement, *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2207–2212, 2013.

[20] P. Müller, C. Balligand, T. Seel, T. Schauer, Iterative Learning Control and System Identification of the Antagonistic Knee Muscle Complex During Gait Using Functional Electrical Stimulation, *IFAC-PapersOnLine*, 50(1):8786–8791, 2017.

[21] D. A. Bristow, M. Tharayil and A. G. Alleyne, A Survey of Iterative Learning Control, *IEEE Control Systems Magazine*, pp. 96-114, June 2006.

[22] K.H. Park, Z. Bien, A generalized iterative learning controller against initial state error, *International Journal of Control*, 73(10):871–881, 2000.

[23] T. Seel, T. Schauer, J. Raisch, Monotonic Convergence of Iterative Learning Control with Variable Pass Length, *International Journal of Control*, 90(3):409–422, 2016.

[24] M. Guth, T. Seel, J. Raisch, Iterative Learning Control with Variable Pass Length Applied to Trajectory Tracking on a Crane with Output Constraints, *Proceedings of the 52nd IEEE Conference on Decision and Control*, pp. 6676–6681, 2013.

[25] T. Seel, C. Werner, J. Raisch, T. Schauer. Iterative Learning Control of a Drop Foot Neuroprosthesis Generating Physiological Foot Motion in Paretic Gait by Automatic Feedback Control. Control Engineering Practice, 48 (1):8797, 2016.

[26] H. Elci, R.W. Longman, M.Q. Phan, J.N. Juang, R. Ugoletti, Simple Learning Control Made Practical by Zero-Phase Filtering: Applications to Robotics, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 49(6):753–767, 2002.

[27] Animation video at https://www.control.tu-berlin.de/ILC_pendulum