Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Adaptive Iterative Learning Control of an Industrial Robot during Neuromuscular Training

M. Ketelhut * **G. M. Brügge** * **F. Göll** ** **B. Braunstein** **
K. Albracht **,*** **D. Abel** *

* *Institute of Automatic Control, RWTH Aachen University, Aachen, Germany (e-mail: m.ketelhut@irt.rwth-aachen.de).*
** *Institute of Biomechanics and Orthopaedics, German Sport University Cologne, Cologne, Germany (e-mail: f.goell@dshs-koeln.de)*
*** *Faculty of Medical Engineering and Technomathematics, University of Applied Sciences, Aachen, Germany (e-mail: albracht@dshs-koeln.de)*

Abstract - To prevent the reduction of muscle mass and loss of strength coming along with the human aging process, regular training with e.g. a leg press is suitable. However, the risk of training-induced injuries requires the continuous monitoring and controlling of the forces applied to the musculoskeletal system as well as the velocity along the motion trajectory and the range of motion. In this paper, an adaptive norm-optimal iterative learning control algorithm to minimize the knee joint loadings during the leg extension training with an industrial robot is proposed. The response of the algorithm is tested in simulation for patients with varus, normal and valgus alignment of the knee and compared to the results of a higher-order iterative learning control algorithm, a robust iterative learning control and a recently proposed conventional norm-optimal iterative learning control algorithm. Although significant improvements in performance are made compared to the conventional norm-optimal iterative learning control algorithm with a small learning factor, for the developed approach as well as the robust iterative learning control algorithm small steady state errors occur.

Keywords: Iterative learning control, Robotic Rehabilitation, Adaptive control

1. INTRODUCTION

Doherty (2003) states that the human aging process results in a significant decline of neuromuscular function evidenced by a reduction in muscle mass and loss of strength, which occurs even in the healthy elderly. For this reason, regular training is essential for the prevention and therapy of age and wealth-related diseases. To reduce the risk of training-induced injuries, the forces applied to the musculoskeletal system as well as the velocity along the motion trajectory and the range of motion have to be monitored and controlled continuously. To address this issue, an experimental research platform, consisting of an industrial robot, a force plate and a motion capture system has been set up at the German Sport University in Cologne (Kolditz et al., 2015). The replacement of the force plate mounted at the robot's end effector allows the execution of different training scenarios. However, in this case the setup is used as a leg press training device, because the leg extension/flexion during leg press training is a multi-articular movement allowing flexibility of muscle activation and control. As high external knee adduction moments reflect high loadings of the medial compartment influencing the development and progression of e.g. osteoarthritis (Kutzner et al., 2013; Trepczynski et al., 2014), high external knee adduction moments shall be prevented. Kolditz et al. (2016) proved in previous work that the foot position and orientation can be used as manipulated variables to control knee joint loadings. Based on these findings, Ketelhut et al. (2019) present a norm-optimal iterative learning control (NOILC) algorithm addressing the control of knee joint loadings during the leg extension training with an industrial robot. There exist various other control mechanisms of lower limb rehabilitation robots (Meng et al., 2015), but these only obtain asymptotic convergence, which guarantees tracking performance at the steady state, when the time horizon goes to infinity.

NOILC algorithms use the repetitive nature of systems to adapt the subsequent control signal sequence based on the previous one. Thus, excellent tracking performance can be obtained from the very beginning of a cycle as stated by Xu and Tan (2003). Apart from this, NOILC schemes are robust against repetitive disturbances while no exact model of the system's dynamics is required. The results illustrate that the algorithm is able to minimize the external knee joint loadings for patients with different alignments of the knee (Ketelhut et al., 2019). However, even with a badly-modeled plant, the required performance can be effectively achieved as the trail number increases (Moore et al., 2005). The reason for this is, that the NOILC is assumed to be inherently robust (Donkers et al., 2008) due to e.g. the learning factor. The learning factor is used

to scale the optimal control signal sequence changes and thus compensate for model uncertainties, while yielding an adequate level of convergence. If the prediction model perfectly fits the system's dynamics, the learning factor can be chosen equal to one. In this case, the algorithm should converge within one cycle. Otherwise high learning factors result in oscillations or may even destabilize the system. Whereas if the system's dynamics are known exactly, no learning-based methods are required and rather direct methods could be used (Bristow et al., 2006). As this is usually not the case in e.g. medical applications and Son et al. (2013) explain that model uncertainty degrades convergence and performance, accounting for model uncertainty in the NOILC design is crucial. Apart from the imprecise model knowledge, ILC algorithms are usually applied to systems with nonlinear and parasitic dynamics. Furthermore, disturbances that vary from trial to trial and initialization errors (Chen et al., 1996) occur. (Owens and Feng, 2003)

The aim of this paper is to design an adaptive iterative learning control algorithm based on a robust ILC scheme to minimize the knee joint loadings during the leg extension training with an industrial robot. An ILC is considered to be robust if, despite the presence of model uncertainties and disturbances, convergence can be obtained and no steady-state error occurs. NOILC algorithms are only simple feed-forward control schemes in time domain. Feedback is only incorporated in the iteration domain. Thus, for the rejection of instantaneous disturbances, the ILC can be combined with e.g. a feedback controller as suggested by Rüschen et al. (2017) to address model uncertainties (Doh, 1999) as well as non-repetitive disturbances (Chin et al., 2004). Furthermore, there exist several different attempts by researchers such as Harte et al. (2005), Longman (2000) and Son et al. (2013) to derive convergence conditions for inverse model-based ILC, iterative learning and repetitive control and gradient-based ILC schemes, respectively. Apart from this, in the literature there are many ILC designs presented, which explicitly account for model uncertainties to improve robustness and convergence. Bristow and Alleyne (2005) present a time-varying low-pass Q-filter in the learning algorithm to enhance robustness of a simple P-type ILC scheme. Nguyen and Banjerdpongchai (2011) investigate a robust iterative learning controller design with a parametric uncertainty model to address time-varying uncertainties. In a higher order ILC (HOILC), the information of several previous cycles instead of just the last one is used to determine the control signal sequence for the subsequent cycle. This is especially beneficial if disturbances occur that vary from trial to trial or the system's dynamics are not consistent. For examples of HOILC algorithms refer to (Moore et al., 2005). Moreover, there exist several different H_∞ ILC algorithms (de Roover, 1996), combinations with disturbance observers (Kim et al., 1996) and stochastic-control-based ILC schemes (Saab, 2001). While the stochastic approach is mainly focused on measurement noise, the H_∞ approach can be used to determine the optimal filter as well as weighting matrices of the quadratic optimization problem (Van De Wijdeven et al., 2011). As stated by Ge et al. (2016), the ideal design of weighting matrices ensures monotonic convergence while maximizing convergence speed and minimizing steady-state errors. Today's methods to determine these weighting matrices however, either lead to conservative performance or require manual tuning. While Donkers et al. (2008) only consider additive uncertainty, the approach can be expanded to take also multiplicative uncertainty into account (Van de Wijdeven et al., 2009). Taking the finite time interval and existence of analytical solutions into consideration, Van De Wijdeven et al. (2011) present a robust worst-case ILC approach based on H_∞, which also allows for noncausal solutions. Son et al. (2013) propose a similar approach formulated as min-max problem to determine the worst-case value of the cost function under model uncertainty, thus yielding a dual minimization problem. This approach also accounts for time-varying uncertainties and can be formulated as convex optimization problem, yielding a global optimal solution.

In this paper, an adaptive optimal ILC algorithm based on (Son et al., 2013) is proposed. The aim of the algorithm is to minimize the knee joint loadings during the leg extension training with an industrial robot. The response of the algorithm is tested in simulation for different patient models and compared to the results of a HOILC, a NOILC and the ROILC algorithm presented by Son et al. (2013).

2. METHODS

In previous work, Ketelhut et al. (2019) proved that a NOILC algorithm is able to minimize knee joint loadings during the leg extension training with an industrial robot. As in the previous work, here the controller aims at minimizing the knee joint moment M, while the knee joint flexion angle Θ moves smoothly along a predefined trajectory to prevent jerks. Plate angle α and Cartesian position in y-direction y serve as controller outputs. To estimate the joint loadings, a setup consisting of the previously described experimental research platform, a motion capturing system and a musculoskeletal lower extremity model, implemented in OpenSim (Delp et al., 2007) has been found suitable (Kolditz et al., 2016). For further details regarding the musculoskeletal lower extremity model refer to (Ketelhut et al., 2019). In this paper however, the algorithm is only tested in simulation to obtain a proof of concept.

2.1 Iterative Learning Control

In an ILC algorithm, the control signal sequence for the subsequent cycle

$$\mathbf{u}_{j+1} = \mathbf{u}_j + \Delta\mathbf{u}_{j+1} \tag{1}$$

is calculated as the sum of the control signal sequence of the previous cycle plus the vector of control signal sequence changes $\Delta\mathbf{u}_{j+1} \in \mathbb{R}^{l(N+1)}$. In this paper, controller inputs and outputs during cycle j

$$\mathbf{y}_j = [M_j(0), \Theta_j(0), ... \quad M_j(N), \Theta_j(N)]^{\mathrm{T}} \in \mathbb{R}^{m(N+1)} \tag{2}$$

$$\mathbf{u}_j = [\alpha_j(0), y_j(0), ... \quad \alpha_j(N), y_j(N)]^{\mathrm{T}} \in \mathbb{R}^{l(N+1)} \tag{3}$$

are sequences with $N+1$ samples per m controller outputs or l inputs and the time index $k \in \{0, 1, .., N\}$, which wraps back to zero at the beginning of a new cycle.

Controller inputs are right external knee joint moment $M(k)$ and knee joint flexion angle $\Theta(k)$, while the plate angle $\alpha(k)$ and Cartesian position in y-direction $y(k)$ serve as corresponding outputs. For further information regarding ILC algorithms in general refer to the work of Xu and Tan (2003).

2.2 Higher Order Iterative Learning Control

In a simple P-type ILC scheme, the control signal sequence changes

$$\Delta \mathbf{u}_{j+1} = \mathbf{\Gamma}\, \mathbf{e}_j, \tag{4}$$

are calculated by the diagonal learning gain matrix $\mathbf{\Gamma} = \mathrm{diag}(\eta) \in \mathbb{R}^{m(N+1)\times m(N+1)}$ with the learning factor $\eta = [0, 1]$ times the tracking error

$$\mathbf{e}_j = \mathbf{r} - \mathbf{y}_j, \tag{5}$$

with the reference

$$\begin{aligned}\mathbf{r} =&[\mathrm{r_M}(0), \mathrm{r_\Theta}(0), ... \\ &\mathrm{r_M}(N), \mathrm{r_\Theta}(N)]^{\mathrm{T}} \in \mathbb{R}^{m(N+1)}.\end{aligned} \tag{6}$$

In this case, the external knee joint moment shall be minimized. Thus, the corresponding reference $\mathbf{r}_\mathrm{M} = [\mathrm{r_M}(0), \mathrm{r_M}(1), ... \ \mathrm{r_M}(N)]^{\mathrm{T}} = \mathbf{0}$ is set to zero for all cycles. The knee joint flexion angle on the other hand shall only change smoothly and uncomfortable jerks shall be prevented. To obtain this performance, the corresponding reference is defined as fourth-order polynomial

$$\mathrm{r_\Theta}(t) = c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0 \tag{7}$$

with $c_4 = 0.3332 \frac{^\circ}{s^4}$, $c_3 = -4.6647 \frac{^\circ}{s^3}$, $c_2 = 16.3265 \frac{^\circ}{s^2}$, $c_1 = 0 \frac{^\circ}{s}$ and $c_0 = -70°$ and the time t in seconds. To improve the robustness of the algorithm described in eq. (4), the information of several previous cycles instead of just the last one can be used to determine the control signal sequence for the subsequent cycle, yielding

$$\Delta \mathbf{u}_{j+1} = \sum_{p=j-g}^{j} \mathbf{\Gamma}_{p-g}\, \mathbf{e}_p \tag{8}$$

with the diagonal learning gain matrices $\mathbf{\Gamma}_p = \mathrm{diag}(\eta_{p-g}) \in \mathbb{R}^{m(N+1)\times m(N+1)}$ and the learning factors $\eta_{p-g} = [0, 1]$ for $p = j - g$, where g denotes the number of additional past cycles used. As mentioned before, the usage of several cycles from the past is beneficial if disturbances occur that vary from trial to trial or the system's dynamics change, thus improving the ILC's robustness. In this work the last three cycles are used to calculate the subsequent control signal sequence, thus $g = 2$.

2.3 Adaptive Optimal Iterative Learning Control

The following explanation of the adaptive optimal iterative learning algorithm closely follows the description of Son et al. (2013) and Son et al. (2014). The designed AOILC scheme considers a linear time-invariant multi-input multi-output (MIMO) system subject to additive uncertainty in this case. In the AOILC approach, the vector of control signal sequence changes $\Delta \mathbf{u}_{j+1} = \Delta \mathbf{u}_{\mathrm{opt},j+1}$ is the result of a quadratic optimization problem

$$\min_{\Delta \mathbf{u}_{\mathrm{opt},j+1}, \lambda_{j+1}} \quad J_{\mathrm{dual}}(\Delta \mathbf{u}_{\mathrm{opt},j+1}, \lambda_{j+1}) \tag{9a}$$

$$\text{s.t.} \quad \lambda_{j+1}\mathbf{I} - \mathbf{Q}_e \geq 0 \tag{9b}$$

$$\mathbf{Q}_e \hat{\mathbf{e}}_{j+1} \in \mathcal{R}(\mathbf{Q}_e - \lambda_{j+1}\mathbf{I}) \tag{9c}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}_{j+1} \leq \mathbf{u}_{\max} \tag{9d}$$

$$\delta \mathbf{u}_{\min} \leq \mathbf{M}_\delta \mathbf{u}_{j+1} \leq \delta \mathbf{u}_{\max} \tag{9e}$$

where J_{dual} denotes the nonlinear dual cost function, which reads

$$J_{dual}(\Delta \mathbf{u}_{j+1}, \lambda_{j+1}) = \|\hat{\mathbf{e}}_{j+1}\|^2_{\mathbf{Q}_{j+1}} + \|\Delta \mathbf{u}_{j+1}\|^2_{\mathbf{R}_{j+1}} \tag{10}$$

with the weighting matrices

$$\mathbf{Q}_{j+1}(\lambda_{j+1}) = (\mathbf{Q}_e^{-1} - \lambda_{j+1}^{-1}\mathbf{I})^\dagger \tag{11}$$

$$\mathbf{R}_{j+1}(\lambda_{j+1}) = \lambda_{j+1}\mathbf{W}_{j+1}^T \mathbf{W}_{j+1} + \mathbf{Q}_\delta, \tag{12}$$

where $(\mathbf{Q}_e^{-1} - \lambda_{j+1}^{-1}\mathbf{I})^\dagger$ is the pseudoinverse of $\mathbf{Q}_e^{-1} - \lambda_{j+1}^{-1}\mathbf{I}$. The first cost function term in eq. (10) penalizes the predicted tracking error without uncertainty

$$\hat{\mathbf{e}}_{j+1} = \mathbf{e}_j - \mathbf{G}\Delta \mathbf{u}_{\mathrm{opt},j+1}, \tag{13}$$

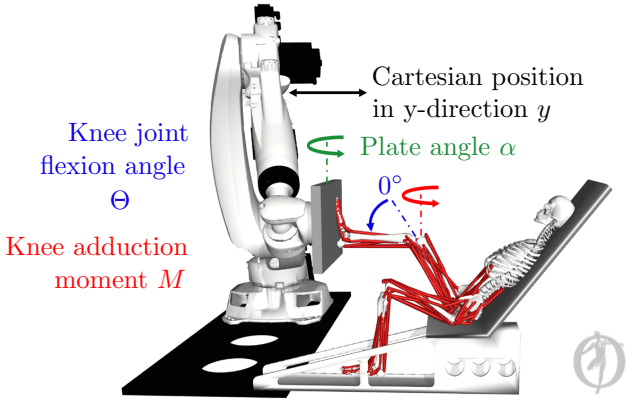
while the second one prevents severe changes of the control signal sequence between two consecutive cycles. The impulse response matrix $\mathbf{G}$ for linear time-invariant systems is of the form

$$\begin{aligned}\mathbf{G} = &\begin{pmatrix} 0 & 0 & \dots & 0 \\ \mathbf{C}\mathbf{B}_d & 0 & \dots & 0 \\ \mathbf{C}\mathbf{A}_d\mathbf{B}_d & \mathbf{C}\mathbf{B}_d & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}\mathbf{A}_d^{N-2}\mathbf{B}_d & \mathbf{C}\mathbf{A}_d^{N-3}\mathbf{B}_d & \dots & 0 \end{pmatrix} \\ &\in \mathbb{R}^{m(N+1)\times l(N+1)},\end{aligned} \tag{14}$$

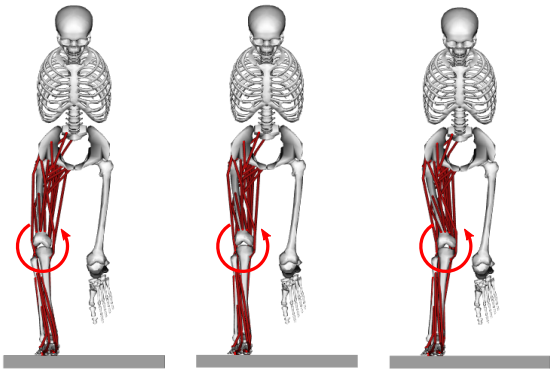
where $\mathbf{A}_d \in \mathbb{R}^{n\times n}$, $\mathbf{B}_d \in \mathbb{R}^{n\times l}$ and $\mathbf{C}_d \in \mathbb{R}^{m\times n}$ refer to the discrete system, input and output matrix with n states, respectively. In this case, G approximates the relation between plate angle $\alpha(t)$ and external knee joint moment $M(t)$ and between Cartesian coordinate in y-direction $y(t)$ and knee joint flexion angle $\Theta(t)$ as first-order systems equivalent to Barton and Alleyne (2010) and Ketelhut et al. (2019). The parameters are determined by parameter estimation using data from simulative step response experiments with the normal knee alignment patient model. According to Son et al. (2013) the proposed ROILC problem can be interpreted as classical NOILC algorithm with trial-by-trial updated weighting matrices based on the additive uncertainty weight $\mathbf{W}_{j+1} \in \mathbb{R}^{m(N+1)\times l(N+1)}$ and the scalar λ_{j+1} times the identity matrix $\mathbf{I} \in \mathbb{R}^{m(N+1)\times m(N+1)}$. The additive uncertainty weight $\mathbf{W}_{j+1}$ can be identified as transfer function that is an upper bound on the difference between the measured data and the system models. This is done once previously for the ROILC to account for all three patient models and iteratively after each iteration for the AOILC algorithm to improve the performance of the control algorithm. The diagonal weighting matrices $\mathbf{Q}_e \in \mathbb{R}^{m(N+1)\times m(N+1)}$ and $\mathbf{Q}_\delta \in \mathbb{R}^{l(N+1)\times l(N+1)}$ represent the (initial) weighting of the cost function terms without uncertainty. The notation $\|\mathbf{p}\|^2_{\mathbf{Q}_i} = \mathbf{p}^T\mathbf{Q}_i\mathbf{p}$ is the squared euclidean norm of $\mathbf{p}$ weighted with the positive semidefinite matrix

$$\mathbf{Q}_i = diag \begin{bmatrix} q_{\alpha,i} & 0 \\ 0 & q_{y,i} \end{bmatrix} \tag{15}$$

and the tuning parameter $q_{\alpha,i}$ and $q_{y,i}$ for $i = e,\delta$.



(a) Robot and lower extremity musculoskeletal model with knee adduction moment M (red), knee joint flexion angle Θ (blue), plate angle α (green) and Cartesian position in y-direction y (black) in OpenSim.



(b) Patient models with varus, normal and valgus alignment of the knee with knee adduction moments (red).

Fig. 3. Lower extremity musculoskeletal models to simulate patients implemented in OpenSim.

The plate angle $\alpha(t)$ as well as the Cartesian coordinate in y-direction $y(t)$ are limited by a lower

$$\mathbf{u}_{\min} = [u_{\alpha,\min}, u_{y,\min}, ..., u_{\alpha,\min}, u_{y,\min}]^{\mathrm{T}} \in \mathbb{R}^{l(N+1)} \quad (16)$$

and an upper

$$\mathbf{u}_{\max} = [u_{\alpha,\max}, u_{y,\max}, ... \quad u_{\alpha,\max}, u_{y,\max}]^{\mathrm{T}} \in \mathbb{R}^{l(N+1)} \quad (17)$$

boundary with the constraints $u_{\alpha,\min}$, $u_{y,\min}$ and $u_{\alpha,\max}$, $u_{y,\max}$. Moreover, the control signal sequence changes within an iteration are restricted by

$$\delta\mathbf{u}_{\min} = [\delta u_{\alpha,\min}, \delta u_{y,\min}, ... \quad \delta u_{\alpha,\min}, \delta u_{y,\min}]^{\mathrm{T}} \in \mathbb{R}^{l(N+1)} \quad (18)$$

and

$$\delta\mathbf{u}_{\max} = [\delta u_{\alpha,\max}, \delta u_{y,\max}, ... \quad \delta u_{\alpha,\max}, \delta u_{y,\max}]^{\mathrm{T}} \in \mathbb{R}^{l(N+1)} \quad (19)$$

to prevent severe changes of the control signal sequence. The numerical gradient of $\mathbf{u}_{j+1}$ is obtained through the multiplication with

$$\mathbf{M}_{\delta} = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ -1 & 1 & \ddots & \vdots & \vdots \\ 0 & -1 & \ddots & 0 & \vdots \\ \vdots & 0 & \ddots & 1 & 0 \\ 0 & 0 & \dots & -1 & 1 \end{bmatrix} \in \mathbb{R}^{l(N+1)\times l(N+1)}. \quad (20)$$

For the detailed derivation of the optimization problem refer to Appendix A or to (Son et al., 2013).

3. RESULTS

3.1 Simulation Setup

To test the response of the designed adaptive ROILC algorithm described in Section 2.3, the musculoskeletal lower extremity model described in (Kolditz et al., 2015) is implemented in OpenSim (Delp et al., 2007) and patient models with a varus, normal and valgus alignment of the knee are created, see Fig. 3. Usually people have a femoral-tibial angle about 175°, while it is much smaller for people with valgus (<−3°) and higher for people with varus knees (>6°) as stated by Schnüke et al. (2009). In this case, femoral-tibial angles with deviations of ±10° are applied to simulate the patient with varus and valgus alignment. The adaptive ROILC algorithm is implemented in *MATLAB* and the ipopt solver in CasADi is used for the nonlinear optimization. CasADi is an open source software framework for numerical optimization (Andersson et al., 2019). During the simulation, the magnitude of the reaction force, which acts orthogonally to the force plate, and the corresponding center of pressure are assumed to remain constant (Force = 1000 N). In four simulations, the tracking performance and disturbance rejection of the proposed adaptive ROILC algorithm is tested and compared to the results of the conventional NOILC algorithm presented by Ketelhut et al. (2019), the HOILC algorithm described in Section 2.2 and the ROILC algorithm with a constant uncertainty weighting $\mathbf{W}$ by Son et al. (2013). For the control parameters used in the simulations refer to Tab. B.1.

3.2 Simulation Results

Fig. 4 shows the external knee joint moment M as well as the corresponding plate α and knee angle Θ and the Cartesian position in y-direction of a patient with normal knees as representative of all three patient models, see Fig. 3. The AOILC response in cycle $1, 3, 5$ and 10 is illustrated as solid blue, dashed red, yellow and dotted purple line, respectively. Moreover, the knee angle reference $\mathbf{r}_{\Theta}$ is displayed as dash-dotted black line. While the external knee joint moment reaches a maximal value of 8.4 Nm during the first cycle, the AOILC algorithm is able to reduce it significantly after 2 iterations. The adaptive ROILC requires 2 iterations to reduce the maximal external knee joint moment deviation below 1 Nm and 3 iterations until it finally converges (root mean square variation of the plate angle below 0.01°). The maximum absolute errors amount to 8.4 Nm and 12.3° knee angle. The corresponding root mean square errors (RMSE) for all three patient models (varus, normal and valgus) are shown in Fig. 5. The RMSE of the external knee joint moment for the patient with varus, normal and valgus alignment are displayed on the top, in the middle and in

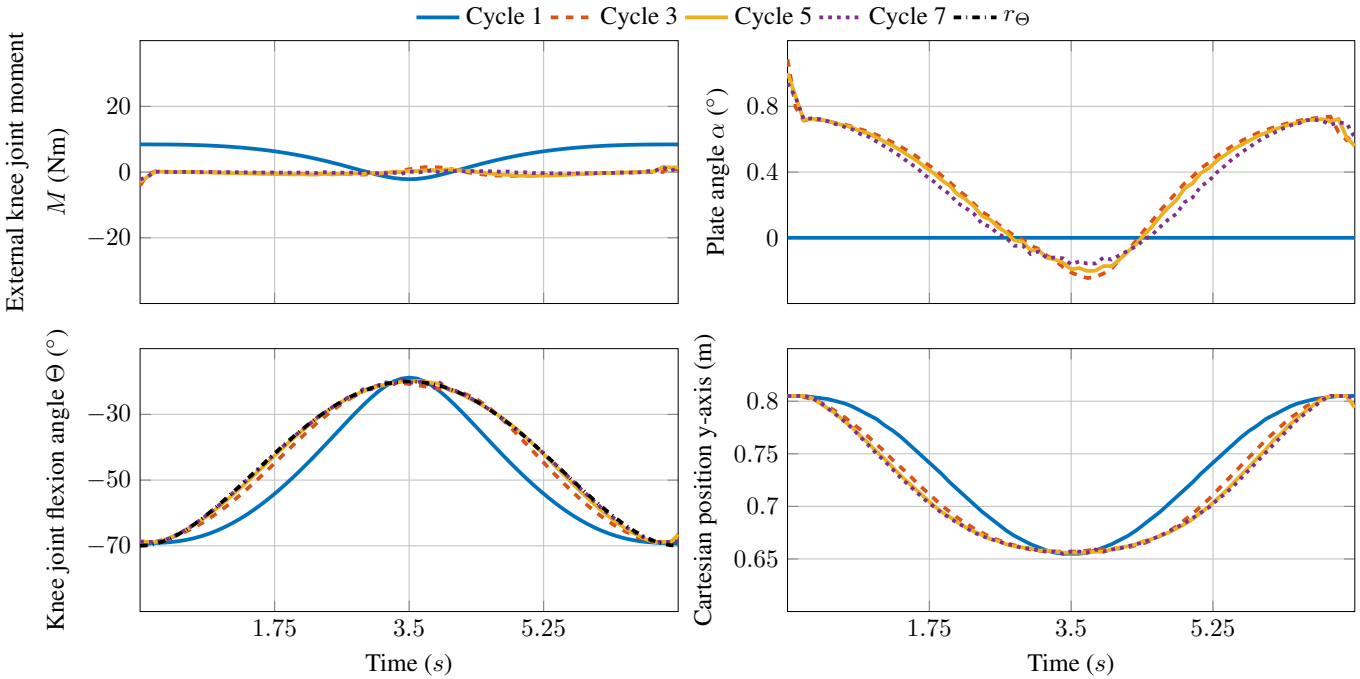


Fig. 4. External knee joint moment M (top left), corresponding plate angle α (top right), knee angle Θ (bottom left) and Cartesian position in y-direction (bottom right) of a patient with normal knees illustrated in Fig. 3 with the AOILC algorithm over time for cycle $1, 3, 5$ and 7.

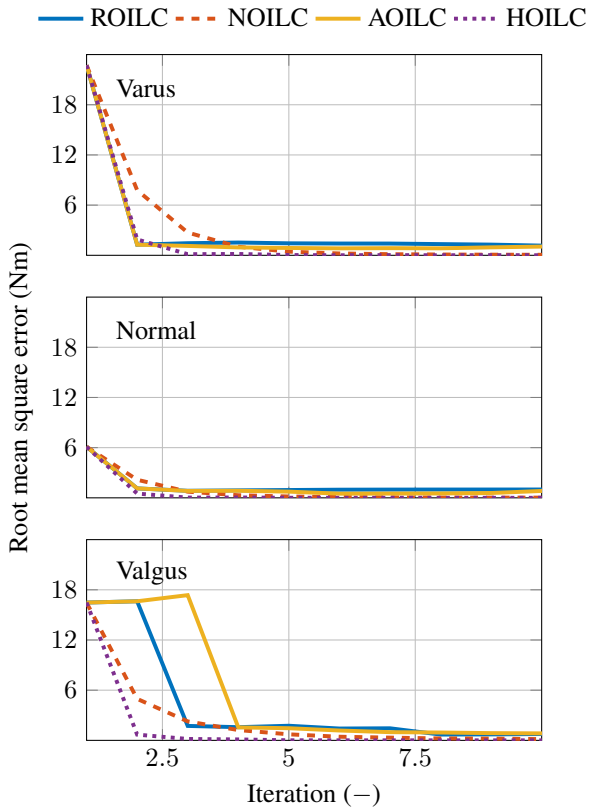


Fig. 5. Root mean square error of patient with varus (top), normal (middle) and valgus (bottom) knee alignment illustrated in Fig. 3 controlled by NOILC, ROILC, HOILC and adaptive ROILC algorithm over multiple iterations.

the bottom, respectively. The results obtained with the ROILC algorithm, the conventional NOILC, the AOILC and the HOILC are illustrated as solid blue, dashed red, solid yellow and dotted purple line, respectively. The results show that compared to the patient model with the normal knee alignment, the initial RMSE of the patient model with varus and valgus alignment is much higher. Nevertheless, all four algorithms only require few iterations to reduce the RMSE significantly. The AOILC as well as the conventional ROILC algorithm though, are $2-3$ iterations faster compared to the conventional NOILC scheme for the patient with varus and normal knee alignment due to the learning factor. The HOILC compared to the NOILC scheme also takes less iterations to converge, but no boundary constraints and cost function can be defined. In this case, additional limitations are required to ensure the safety of the patients. Nevertheless, the NOILC and HOILC scheme for the patient with valgus knee alignment are $1-2$ iterations faster than the the AOILC and ROILC algorithms even though this is the patient model with the highest model uncertainties. Apart from this, compared to the HOILC and NOILC algorithm small steady-state errors occur, as $\lambda_{j+1} \to \infty$ for $j \to \infty$ (Son et al., 2014). Consequently, $\|\mathbf{R}_{j+1}(\lambda_{j+1})\|$ increases to very large values and thus changes of the optimal control signal sequence are severely punished.

4. CONCLUSION

In this paper an AOILC algorithm is proposed to minimize the knee joint loadings during the leg extension training with an industrial robot. The algorithm inherently accounts for additive model uncertainty by adjusting the weighting matrices in each iteration. For this purpose, a worst-case performance index under additive variable

weighted uncertainty is minimized yielding a nonlinear dual optimization problem. For the simulative validation of the presented control algorithm three different patient models with varus, normal and valgus alignment of the knee are used. The results illustrate that the AOILC algorithm compared to the NOILC and the ROILC algorithm with constant uncertainty weighting yield improved performance. However, in contrast to the NOILC algorithm a minimal steady-state error occurs. To validate the findings, further investigations with real patients training at the experimental research platform are required. Moreover, the accuracy of the uncertainty weighting could also be improved based on experimental data. So far, the direct measurement of knee joint forces is only possible in patients receiving total knee arthroplasty. Thus, a model mismatch between the OpenSim model and the real patient may result in an incorrect estimation of the external knee joint moment. To incorporate the AOILC algorithm in all training scenarios realized on the experimental research platform, the definition of the system's dynamics as function of the way along the path instead of time might be favorable.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the funding by the Federal Ministry of Education and Research received within the research project "Lernende roboterassistierte Systeme für das neuromuskuläre Training - RoSylerNT" (16SV7853/16SV7854) as part of the program "Forschungsprogramm zur Mensch-Technik-Interaktion: Technik zum Menschen bringen".

LITERATUR

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36. doi: 10.1007/s12532-018-0139-4.

Barton, K.L. and Alleyne, A.G. (2010). A norm optimal approach to time-varying ilc with application to a multi-axis robotic testbed. *IEEE Transactions on Control Systems Technology*, 19(1), 166–180.

Bristow, D.A. and Alleyne, A.G. (2005). Monotonic convergence of iterative learning control for uncertain systems using a time-varying q-filter. In *Proceedings of the 2005, American Control Conference, 2005.*, 171–177. IEEE.

Bristow, D.A., Tharayil, M., and Alleyne, A.G. (2006). A survey of iterative learning control. *IEEE Control Systems*, 26(3), 96–114.

Chen, Y., Wen, C., Xu, J.X., and Sun, M. (1996). An initial state learning method for iterative learning control of uncertain time-varying systems. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 4, 3996–4001. IEEE.

Chin, I., Qin, S.J., Lee, K.S., and Cho, M. (2004). A two-stage iterative learning control technique combined with real-time feedback for independent disturbance rejection. *Automatica*, 40(11), 1913–1922.

de Roover, D. (1996). Synthesis of a robust iterative learning controller using an h/sub/spl infin//approach. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 3, 3044–3049. IEEE.

Delp, S.L., Anderson, F.C., Arnold, A.S., Loan, P., Habib, A., John, C.T., Guendelman, E., and Thelen, D.G. (2007). Opensim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering*, 54(11), 1940–1950.

Doh, T.Y. (1999). Robust iterative learning control with current feedback for uncertain linear systems. *International Journal of Systems Science*, 30(1), 39–47.

Doherty, T.J. (2003). Invited review: aging and sarcopenia. *Journal of applied physiology*, 95(4), 1717–1727.

Donkers, T., van de Wijdeven, J., and Bosgra, O. (2008). Robustness against model uncertainties of norm optimal iterative learning control. In *2008 American Control Conference*, 4561–4566. IEEE.

Ge, X., Stein, J.L., and Ersal, T. (2016). Optimization based weighting matrices design for norm optimal iterative learning control. In *ASME 2016 Dynamic Systems and Control Conference*, V002T28A002–V002T28A002. American Society of Mechanical Engineers.

Harte, T., Hätönen, J., and Owens*, D. (2005). Discrete-time inverse model-based iterative learning control: stability, monotonicity and robustness. *International Journal of Control*, 78(8), 577–586.

Ketelhut, M., Göll, F., Braunstein, B., Albracht, K., and Abel, D. (2019). Iterative learning control of an industrial robot for neuromuscular training. In *2019 IEEE Conference on Control Technology and Applications (CCTA)*, 926–932. IEEE.

Kim, B., Chung, W., and Youm, Y. (1996). Robust learning control for robot manipulators based on disturbance observer. In *Proceedings of the 1996 IEEE IECON. 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, volume 2, 1276–1282. IEEE.

Kolditz, M., Albin, T., Abel, D., Fasse, A., Brüggemann, G.P., and Albracht, K. (2015). Simulative analysis of joint loading during leg press exercise for control applications. *IFAC-PapersOnLine*, 48(20), 435–440.

Kolditz, M., Albin, T., Albracht, K., Brüggemann, G.P., and Abel, D. (2016). Isokinematic leg extension training with an industrial robot. In *Biomedical Robotics and Biomechatronics (BioRob), 2016 6th IEEE International Conference on*, 950–955. IEEE.

Kutzner, I., Trepczynski, A., Heller, M.O., and Bergmann, G. (2013). Knee adduction moment and medial contact force–facts about their correlation during gait. *PLoS One*, 8(12), e81036.

Longman, R.W. (2000). Iterative learning control and repetitive control for engineering practice. *International journal of control*, 73(10), 930–954.

Meng, W., Liu, Q., Zhou, Z., Ai, Q., Sheng, B., and Xie, S.S. (2015). Recent development of mechanisms and control strategies for robot-assisted lower limb rehabilitation. *Mechatronics*, 31, 132–145.

Moore, K.L., Chen, Y., and Ahn, H.S. (2005). Algebraic *h_infty* design of higher-order iterative learning controllers. In *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, 1207–1212. IEEE.

Nguyen, D.H. and Banjerdpongchai, D. (2011). A convex optimization approach to robust iterative learning control for linear systems with time-varying parametric

uncertainties. *Automatica*, 47(9), 2039–2043.
Owens, D.H. and Feng, K. (2003). Parameter optimization in iterative learning control. *International Journal of Control*, 76(11), 1059–1069.
Rüschen, D., Prochazka, F., Amacher, R., Bergmann, L., Leonhardt, S., and Walter, M. (2017). Minimizing left ventricular stroke work with iterative learning flow profile control of rotary blood pumps. *Biomedical Signal Processing and Control*, 31(Supplement C), 444–451.
Saab, S.S. (2001). A discrete-time stochastic learning control algorithm. *IEEE Transactions on Automatic Control*, 46(6), 877–887.
Schnüke, M., Schulte, E., and Schumacher, U. (2009). Prometheus lernatlas der anatomie. allgemeine anatomie und bewegungssystem.
Son, T.D., Pipeleers, G., and Swevers, J. (2013). Robust optimal iterative learning control with model uncertainty. In *52nd IEEE Conference on Decision and Control*, 7522–7527. IEEE.
Son, T.D., Pipeleers, G., and Swevers, J. (2014). Experimental validation of robust iterative learning control on an overhead crane test setup. *IFAC Proceedings Volumes*, 47(3), 5981–5986.
Trepczynski, A., Kutzner, I., Bergmann, G., Taylor, W.R., and Heller, M.O. (2014). Modulation of the relationship between external knee adduction moments and medial joint contact forces across subjects and activities. *Arthritis & Rheumatology*, 66(5), 1218–1227.
Van de Wijdeven, J., Donkers, T., and Bosgra, O. (2009). Iterative learning control for uncertain systems: robust monotonic convergence analysis. *Automatica*, 45(10), 2383–2391.
Van De Wijdeven, J., Donkers, M., and Bosgra, O. (2011). Iterative learning control for uncertain systems: Noncausal finite time interval robust control design. *International Journal of Robust and Nonlinear Control*, 21(14), 1645–1666.
Xu, J.X. and Tan, Y. (2003). *Linear and nonlinear iterative learning control*, volume 291. Springer.

Appendix A. ADAPTIVE ITERATIVE LEARNING CONTROL

The following derivation of the adaptive optimal iterative learning algorithm closely follows the description of Son et al. (2013) and Son et al. (2014). Consider an AOILC approach, where the vector of control signal sequence changes $\Delta\mathbf{u}_{j+1} = \Delta\mathbf{u}_{\text{opt},j+1}$ is the result of the quadratic optimization problem

$$\min_{\Delta\mathbf{u}_{\text{opt},j+1}} \max_{\Delta} J(\Delta\mathbf{u}_{\text{opt},j+1}, \boldsymbol{\Delta}), \tag{A.1}$$

with the cost function

$$J(\Delta\mathbf{u}_{j+1}, \boldsymbol{\Delta}) = \|\mathbf{r} - \hat{\mathbf{y}}_{j+1}\|^2_{\mathbf{Q}_e} + \|\Delta\mathbf{u}_{\text{opt},j+1}\|^2_{\mathbf{Q}_\delta}.$$

The first cost function term penalizes the predicted tracking error considering uncertainty, while the second one prevents severe changes of the control signal sequence between two consecutive cycles. The controlled variable sequence of the subsequent cycle

$$\hat{\mathbf{y}}_{j+1} = \mathbf{y}_j + (\mathbf{G} + \boldsymbol{\Delta}\mathbf{W}_{j+1})\Delta\mathbf{u}_{\text{opt},j+1} \tag{A.2}$$

is predicted as the sum of the measured controlled variable sequence during the previous cycle $\mathbf{y}_j$ and the impulse response matrix $\mathbf{G}$. To account for model uncertainties, an additive uncertainty $\boldsymbol{\Delta} \in \mathbb{R}^{m(N+1)\times m(N+1)}$, where $\|\boldsymbol{\Delta}\| \leq 1$, with the weight $\mathbf{W}_{j+1}$ is added to (A.2).

To find the worst case cost function (A.1) with respect to $\boldsymbol{\Delta}$, the maximization problem

$$\max_{\mathbf{v}_{j+1}} \|\underbrace{\hat{\mathbf{e}}_{j+1} - \mathbf{v}_{j+1}}_{\mathbf{r}-\hat{\mathbf{y}}_{j+1}}\|^2_{\mathbf{Q_e}} \tag{A.3}$$

subject to $\|\boldsymbol{\Delta}\| \leq 1$ can be solved, with

$$\mathbf{v}_{j+1} = \boldsymbol{\Delta}\mathbf{W}_{j+1}\Delta\mathbf{u}_{\text{opt},j+1} \tag{A.4}$$

similar to (Son et al., 2013). The other cost function term in (A.1) does not rely on $\boldsymbol{\Delta}$. Reformulation of the constraint $\|\boldsymbol{\Delta}\| \leq 1$ as

$$\|\mathbf{v}_{j+1}\|^2 \leq \|\mathbf{W}_{j+1}\Delta\mathbf{u}_{\text{opt},j+1}\|^2 \tag{A.5}$$

and introducing the Lagrangian multiplier λ_{j+1} yields the Lagrangian function

$$\begin{aligned} L(\mathbf{v}_{j+1}, \lambda_{j+1}) &= \|\hat{\mathbf{e}}_{j+1} - \mathbf{v}_{j+1}\|^2_{\mathbf{Q}_e} \\ &\quad + \lambda_{j+1}(\|\mathbf{W}_{j+1}\Delta\mathbf{u}_{\text{opt},j+1}\|^2 - \|\mathbf{v}_{j+1}\|^2). \end{aligned} \tag{A.6}$$

The solution corresponding to the original constrained optimization is the maximum of the Lagrangian function. Thus, the partial derivatives of (A.6) with respect to $\mathbf{v}_{j+1}$ have to fulfill

$$\frac{\partial L}{\partial \mathbf{v}_{j+1}} = -2\hat{\mathbf{e}}_{j+1}\mathbf{Q}_e + 2(\mathbf{Q}_e - \lambda_{j+1}\mathbf{I})\mathbf{v}^*_{j+1} = 0 \tag{A.7}$$

and

$$\frac{\partial^2 L}{\partial \mathbf{v}_{j+1}\partial \mathbf{v}^*_{j+1}} = (\mathbf{Q}_e - \lambda_{j+1}\mathbf{I}) \leq 0 \tag{A.8}$$

yielding

$$\mathbf{v}^*_{j+1} = (\mathbf{Q}_e - \lambda_{j+1}\mathbf{I})^\dagger\mathbf{Q}_e\hat{\mathbf{e}}_{j+1} \tag{A.9}$$

with $\lambda_{j+1}\mathbf{I} - \mathbf{Q}_e \geq 0$ due to the second order partial derivative (A.8) and $\mathbf{Q}_e\hat{\mathbf{e}}_{j+1} \in \mathcal{R}(\mathbf{Q}_e - \lambda_{j+1}\mathbf{I})$ to solve the equation problem, where $\mathcal{R}(\mathbf{Q}_e - \lambda_{j+1}\mathbf{I})$ is the range of $\mathbf{Q}_e - \lambda_{j+1}\mathbf{I}$. According to this, the Lagrangian dual function

$$g(\lambda_{j+1}) = \begin{cases} L(\mathbf{v}^*_{j+1}, \lambda_{j+1}) & \lambda_{j+1}\mathbf{I} - \mathbf{Q}_e \geq 0, \\ & \mathbf{Q}_e\hat{\mathbf{e}}_{j+1} \in \mathcal{R}(\mathbf{Q}_e - \lambda_{j+1}\mathbf{I}) \\ +\infty & \text{otherwise} \end{cases} \tag{A.10}$$

is derived, where

$$\begin{aligned} L(\mathbf{v}^*_{j+1}, \lambda_{j+1}) =& \hat{\mathbf{e}}^T_{j+1}(\mathbf{Q}_e^{-1} - \lambda_{j+1}^{-1}\mathbf{I})^\dagger\hat{\mathbf{e}}_{j+1} \\ &+ \lambda_{j+1}\|\mathbf{W}_{j+1}\Delta\mathbf{u}_{\text{opt},j+1}\|^2 \end{aligned} \tag{A.11}$$

and $(\mathbf{Q}_e - \lambda_{j+1}\mathbf{I})^\dagger$ is the pseudoinverse of $\mathbf{Q}_e - \lambda_{j+1}\mathbf{I}$. As a result, the maximization problem in (A.3) can also be defined as

$$\min_{\lambda_{j+1}} \quad g(\lambda_{j+1}) \tag{A.12a}$$
$$\text{s.t.} \quad \lambda_{j+1}\mathbf{I} - \mathbf{Q}_e \geq 0 \tag{A.12b}$$
$$\mathbf{Q}_e\hat{\mathbf{e}}_{j+1} \in \mathcal{R}(\mathbf{Q}_e - \lambda_{j+1}\mathbf{I}) \tag{A.12c}$$

and combined with the minimization problem in (A.1) to

$$\min_{\Delta\mathbf{u}_{\text{opt},j+1}, \lambda_{j+1}} \quad J_{\text{dual}}(\Delta\mathbf{u}_{\text{opt},j+1}, \lambda_{j+1}) \tag{A.13a}$$
$$\text{s.t.} \quad \lambda_{j+1}\mathbf{I} - \mathbf{Q}_e \geq 0 \tag{A.13b}$$
$$\mathbf{Q}_e\hat{\mathbf{e}}_{j+1} \in \mathcal{R}(\mathbf{Q}_e - \lambda_{j+1}\mathbf{I}) \tag{A.13c}$$

where J_{dual} denotes the nonlinear dual cost function, which reads

$$\begin{aligned} J_{dual}(\Delta\mathbf{u}_{j+1}, \lambda_{j+1}) &= \hat{\mathbf{e}}^T_{j+1}(\mathbf{Q}_e^{-1} - \lambda_{j+1}^{-1}\mathbf{I})^\dagger\hat{\mathbf{e}}_{j+1} \\ &+\lambda_{j+1}\|\mathbf{W}_{j+1}\Delta\mathbf{u}_{\text{opt},j+1}\|^2 + \|\Delta\mathbf{u}_{\text{opt},j+1}\|^2_{\mathbf{Q}_\delta} \end{aligned} \tag{A.14}$$

Rearranging of (A.14) yields the dual cost function described in Section 2.3

$$J_{dual}(\Delta\mathbf{u}_{j+1}, \lambda_{j+1}) = \|\hat{\mathbf{e}}_{j+1}\|^2_{\mathbf{Q}_{j+1}} \tag{A.15}$$

$$+ \|\Delta\mathbf{u}_{\mathrm{opt},j+1}\|^2_{\mathbf{R}_{j+1}} \tag{A.16}$$

with the weighting matrices

$$\mathbf{Q}_{j+1}(\lambda_{j+1}) = (\mathbf{Q}_e^{-1} - \lambda_{j+1}^{-1}\mathbf{I})^{\dagger}, \tag{A.17}$$

$$\mathbf{R}_{j+1}(\lambda_{j+1}) = \lambda_{j+1}\mathbf{W}^T\mathbf{W} + \mathbf{Q}_{\delta}. \tag{A.18}$$

Appendix B. CONTROL PARAMETERS

Tab. B.1 includes the parameters of the HOILC and AOILC algorithm described in Section 2.2 and Section 2.3 as well as the parameters of the conventional NOILC algorithm presented by Ketelhut et al. (2019).

Tab. B.1. Parameters of different control algorithms used for simulation.

Description	Symbol	Value	Unit
Cycle duration	T_c	7	s
Sample Time	T_s	0.1	s
Lower boundary α	$u_{\alpha,\min}$	-2.5	$^\circ$
Upper boundary α	$u_{\alpha,\max}$	2.5	$^\circ$
Proportional coefficient $\mathbf{G}_\Theta$	$K_{\Theta,y}$	-600	$\frac{^\circ}{\mathrm{m}}$
Time constant $\mathbf{G}_\Theta$	$\tau_{\Theta,y}$	0.01	s
Lower boundary y	$u_{y,\min}$	0.655	m
Upper boundary y	$u_{y,\max}$	0.8	m
Proportional coefficient $\mathbf{G}_M$	$K_{M,\alpha}$	-11.45	$\frac{\mathrm{Nm}}{^\circ}$
Time constant $\mathbf{G}_M$	$\tau_{M,\alpha}$	0.1	s
NOILC			
Learning factor	η	0.7	-
Deviation weight	q_e	10	-
Gradient weight	q_δ	1000	-
ROILC			
Deviation weight α	$q_{\alpha,e}$	1	-
Learning weight α	$q_{\alpha,\delta}$	10^-3	-
Deviation weight y	$q_{y,e}$	1	-
Learning weight y	$q_{y,\delta}$	10^-3	-
Lower boundary change of α during cycle	$\delta u_{\alpha,\min}$	-0.3	$\frac{^\circ}{\mathrm{s}}$
Upper boundary change of α during cycle	$\delta u_{\alpha,\max}$	0.3	$\frac{^\circ}{\mathrm{s}}$
Lower boundary change of y during cycle	$\delta u_{y,\min}$	-0.02	$\frac{^\circ}{\mathrm{s}}$
Upper boundary change of y during cycle	$\delta u_{y,\max}$	0.02	$\frac{^\circ}{\mathrm{s}}$
HOILC			
Number of additional past cycles used	g	2	-
Learning factor last cycle	η_1	0.08	-
Learning factor second-last cycle	η_2	0.001	-
Learning factor third-last cycle	η_3	0.0005	-