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

An ILC approach to feed-forward friction compensation ⋆

Mikael Norrlöf *,** Svante Gunnarsson **

* *Robotics and Discrete Automation, ABB AB*
** *Department of Electrical Engineering, Linköping University, 58183 Linköping, Sweden (e-mail: author@liu.se)*

Abstract: An iterative, learning based, feed-forward method for compensation of friction in industrial robots is studied. The method is put into an ILC framework by using a two step procedure proposed in literature. The friction compensation method is based on a black-box friction model which is learned from operational data, and this can be seen as the first step in the method. In the second step the learned model is used for compensation of the friction using the reference joint velocity as input. The approach is supported by simulation experiments.

Keywords: Friction, learning control, industrial robots, iterative methods

1. INTRODUCTION

Iterative learning control (ILC) is an established method in the control field, and the origin of the method goes back to Uchiyama (1978) and Arimoto et al. (1984). Numerous papers have been published, and excellent surveys of the topic are given in e.g. Bristow et al. (2006); Ahn et al. (2007). The method is based on the property that the system under consideration is carrying out a given task repeatedly. By using a measure of the performance in one iteration it is possible to modify the control action for the next iteration and thereby obtain an improvement of the performance. ILC has been studied in both continuous and discrete time settings, for linear as well as non-linear systems, and in both the time and frequency domain. A number of applications have been reported from various fields, with emphasis on industrial robots.

Originally ILC was proposed as an entirely model free method, but it turns out that this leads to rather restrictive conditions for convergence. However, it has also been found that considerable improvements of the control system performance can be obtained using fairly simple models of the system. ILC is very good at reducing the low frequency components of the control error, while it is more difficult to handle the high frequency parts. This can be seen in the convergence criteria and expressions for the remaining error which will be discussed in Section 2. One situation where, by the nature of the problem, the error contains high frequency components is systems with friction, and this phenomenon is difficult to handle with conventional ILC, and some alternative or additional method is needed.

The purpose of this paper is to put an iterative feed-forward friction compensation, presented in Johansson et al. (2018) and Johansson (2017), into an ILC framework. The friction compensation method in these references uses a black-box friction model based on e.g. B-splines or some other function approximation. The friction compensation method as such lies somewhat outside the traditional ILC approach, in that it involves learning a parameterized model of (a part of) the system under consideration. However, a more general ILC framework has been proposed in Steinhauser (2019), and using this framework the learning of the friction model fits in very well. Section 2.4 of Steinhauser (2019) presents a nonlinear ILC algorithm consisting of two steps, where the first step is denoted *model correction* and the second step is called *model inversion*. The two-step approach will be presented in some more detail below.

The paper is organized as follows. Section 2 gives a brief ILC background. It is followed by Section 3, which presents the type of friction compensation that is studied here. In Section 4 the two step approach to ILC, introduced in Steinhauser (2019), is presented briefly. Section 5 shows how the learning based feed-forward friction compensation can be put in the two-step ILC framework, and the approach is then illustrated in an example in Section 6. Finally Section 7 contains some conclusions.

2. ILC BACKGROUND

This section gives a brief introduction to ILC. As mentioned before there are several very good survey papers about ILC, and for details it is referred to them. Consider linear discrete-time SISO systems described by

$$y_k(t) = T_r(q)r(t) + T_u(q)u_k(t) \tag{1}$$

with reference $r(t)$, ILC input $u_k(t)$ and output $y_k(t)$ at iteration k. The shift operator is denoted q. Load and measurement disturbances are omitted for simplicity. All signals are defined on a finite time interval $t = nT_s$, $n \in [1, N_s]$ with N_s number of samples and sampling interval $T_s = 1$ if nothing else is stated. The system can have internal feed-back, so $T_u(q)$ and $T_r(q)$ contain the system to be controlled and the controller in operation. Alternatively the system can be described using a matrix

⋆ This work was sponsored by the VINNOVA Competence Center LINK-SIC.

approach (in the literature often denoted lifted approach) but due to space limitations this is omitted here.

Using the transfer operator framework the ILC algorithms are often described according to

$$u_{k+1}(t) = Q(q)\big(u_k(t) + L(q)e_k(t)\big) \tag{2}$$

$$e_k(t) = r(t) - y_k(t) \tag{3}$$

where $Q(q)$ and $L(q)$ may be non-causal filters. The condition for convergence when the algorithm (2) is applied to (1) is given by

$$| Q(e^{i\omega}) || 1 - L(e^{i\omega})T_u(e^{i\omega}) |< 1 \quad \forall\omega \tag{4}$$

and the resulting error, $E_\infty(e^{i\omega})$, after convergence is given by

$$E_\infty(e^{i\omega}) = \frac{1 - Q(e^{i\omega})}{1 - Q(e^{i\omega})(1 - L(e^{i\omega})T_u(e^{i\omega}))}R(e^{i\omega}) \tag{5}$$

Equation (5) illustrates the well known property that for the error to tend to zero it is necessary to choose $Q(q) = 1$. On the other hand, for robustness and disturbance suppression reasons it is normally necessary to choose $Q(q) \neq 1$, and the standard choice is to choose $Q(q)$ as a zero-phase low pass filter. When the magnitude of Q tends to zero for high frequencies equation (5) shows that $E_\infty(e^{i\omega})$ will tend to one, and it will hence not be possible to eliminate the high frequency components of the error. A detailed discussion about convergence aspects, can be found in e.g. Norrlöf and Gunnarsson (2002). This reference also presents conditions for convergence and monotone convergence for the matrix formulation via eigenvalue and singular value conditions. It is important to remember that the expressions for convergence and final error, (4) and (5), are based on the assumption that the system to be controlled and the ILC algorithm are linear. Since friction is a nonlinear phenomenon the expressions have to interpreted with care, but they can still give useful insight into the behavior of ILC applied to systems with friction.

There are several studies of ILC being applied to systems with friction. An early reference is Wang and Longman (1994), which shows convergence of an ILC algorithm also in presence of Coulomb friction. The results are however based on rather restrictive conditions since only Coulomb friction is considered and it is assumed that the friction coefficient is known. The result also requires that the reference signal has special properties. In Driessen and Sadegh (2004) the authors extend the results to the multivariable case and also consider the influence of bounded input signals. Some additional early observations are presented in Norrlöf and Gunnarsson (1997), where it is illustrated that conventional ILC is unable to reduce the high frequency components of the error, due to the friction. Fleischer (1997) studies a combination of ILC and friction compensation, and it is shown that a substantial improvement is obtained when the two approaches are combined. Another contribution is Tsai et al. (2006) where the effects of friction are compensated via a modification of the reference signal to the control system.

3. FRICTION MODELING AND COMPENSATION

Friction is a complex phenomenon and it has been studied in numerous publications. See for example Armstrong-Hélouvry (1991), Al-Bender and Swevers (2008), Harnoy et al. (2008), Stotsky (2007), and Bittencourt and Gunnarsson (2012). In robot manipulator applications where high precision is required, friction is known to cause problems, especially for low-velocity motions. Friction can appear distributed in the system but a major part can be related to the motors and gearboxes.

One model which has received a lot of attention is the so called LuGre model, which is presented and discussed in Åstrom and de Wit (2008). The LuGre model captures also the dynamic properties of the friction, but there is also a static version of the model capturing Coulomb and viscous friction and the Stribeck effect. The static LuGre model is given by

$$\tau_F(v) = (f_c + (f_s - f_c)e^{-(v/v_s)^2})sign(v) + f_v v \tag{6}$$

where τ_F denotes the friction force and v is the relative velocity between the surfaces. In the model it has also been assumed that the viscous friction is linear. The LuGre model is based on physical insight and the process to estimate the parameters f_c, f_s and f_v often requires data collection with specially selected input signals.

An alternative approach is to formulate a friction model of black-box type and learn the parameters from data. The approach in this paper is inspired by the work presented in Johansson et al. (2018) and Johansson (2017) where the friction is modeled using B-spline networks (BSN). The structure of a B-spline network can be seen in Figure 1. The network contains one hidden layer where the so called B-splines are defined. The input x can be seen mapped to each individual B-spline within the network, where N represents the number of B-splines. Together with each B-spline there is also a weight, and the outputs of the individual splines are then summed together in order to generate the output y. When presented with training data, the weights are adapted via a suitable minimization method.

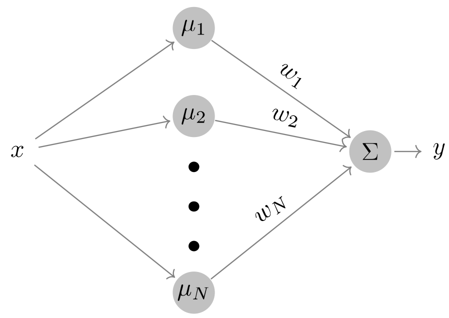

Fig. 1. B-spline network structure where x is the input to the N number of B-splines in the hidden layer. Each B-spline μ_i is assigned a weight w_i were the output is a linear combination of B-splines and weights.

The output of the function can be expressed as

$$y(x) = \sum_{i=1}^{N} \mu_i(x) \cdot w_i \tag{7}$$

where the B-splines are given in the form of the function $\mu_i(x)$ given that x is the input. Using the B-spline approach involves choosing the number of splines and the location of the knots. These choices depend on the application, and some aspects of the choices in the friction modeling application are discussed in Johansson et al. (2018).

The model in (7) is a linear regression and an alternative way to express the model is

$$y(x) = \varphi^T(x)\theta \tag{8}$$

where θ contains the model coefficients (weights)

$$\theta = (w_1, \dots, w_N)^T \tag{9}$$

and the regression vector φ consists of the spline functions, i.e.

$$\varphi(x) = (\mu_1(x), \dots, \mu_N(x))^T \tag{10}$$

The representation in equation (7) is general and can be used also with other types of functions. For example, a very simple model of Coulomb friction with the same friction coefficient in both directions, i.e.

$$\tau_F(v) = f_c \mathrm{sign}(v) \tag{11}$$

is obtained by putting $N = 1$, $\theta = f_c$ and

$$\mu_1 = \mathrm{sign}(v) \tag{12}$$

This relatively simple approach for modeling of the friction will be evaluated in the simulation example in Section 6. The general case using B-splines to model the friction is discussed in Johansson et al. (2018).

The overall structure of the feed-forward friction compensation presented in Johansson et al. (2018) and Johansson (2017) is illustrated in Figure 2, where the variables z and v represent signals from the robot and the regulator used in the learning procedure.

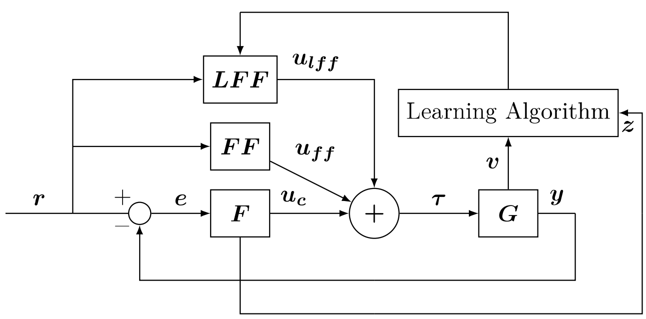

Fig. 2. Control system based on feed-forward (FF) and feed-back (F) control extended with learning-based feed-forward.

A key challenge when modeling and estimating friction is that the true friction force τ_F acting on the robot cannot be measured. Therefore it has to be replaced by an estimate. The approach here is to use parts of the control signal, generated by the robot controller, as estimate of the friction force. The variable u_p is defined as $u_p = u_c - u_I$, where u_c is the controller output, see Figure 2, and u_I is the integral part of the controller output. The argument is that if the feed-forward is based on a sufficiently accurate model of the robot, the task of the controller output is to counteract the friction force and to compensate for gravitational forces. By subtracting the integral part u_I handling the gravitational forces, the remaining part of the input, i.e. u_P represents an estimate of the friction force. This approach was applied with good results also in Längkvist (2009).

4. A TWO-STEP APPROACH TO ILC

A novel and very interesting approach to ILC is presented in Steinhauser (2019), in which nonlinear ILC is formulated as a two-step procedure, extending the work presented in Volckaert (2012). In the approach the ILC algorithm is divided into a model correction step and a model inversion step. To illustrate let the model of the relationship between input and output be described by

$$y_k(t) = \mathcal{T}(u_k(t), \hat{\theta}) \tag{13}$$

In the first step the model is updated (corrected) by using the input-output data collected in iteration k to compute a new estimate of the parameter vector θ. In Steinhauser (2019) this is done by minimizing a criterion of the type

$$\hat{\theta}_k = \arg\min_{\theta} \|y_k - \mathcal{T}(u_k, \theta)\|_V^2 + \|\theta\|_{W_1}^2 + \|\theta - \hat{\theta}_{k-1}\|_{W_2}^2 \tag{14}$$

where the first term minimizes the output error of the model, the second is a regularization terms penalizing the parameter vector itself, and the third term is used to affect the rate of change of the parameter vector. Various aspects of the choice of the weighting matrices V, W_1 and W_2 are discussed in Steinhauser (2019).

In the second step, denoted model inversion step, the new input vector is computed via a second minimization, where the input in iteration $k + 1$ is computed via

$$u_{k+1} = \arg\min_{u} \|y_d - \mathcal{T}(u, \hat{\theta}_k)\|_Q^2 + \|u\|_{R_1}^2 + \|u - u_k\|_{R_2}^2 \tag{15}$$

As pointed out in Steinhauser (2019) this step is essentially identical to the so called norm-optimal ILC formulation. The matrix Q is a weight on control error, and the matrices R_1 and R_2 put weights on the magnitude of the ILC input signal and the rate of change of the input signal respectively. Choosing $R_1 = 0$ leads to an update of the input without forgetting, or, in other words $Q = 1$ in ILC algorithm. Further aspects of the choice of the design matrices are discussed in Steinhauser (2019).

5. FRICTION COMPENSATION AS THE TWO-STEP APPROACH TO ILC

The estimation and update of the chosen friction model will now be formulated as the first step of the two-step approach. Consider the model in equation (8), and let x and y denote the velocity and friction force respectively. Assume that, at iteration k, there are N_s measurements of each variable. Let the measured friction values be collected in the vector Y_k, i.e.

$$Y_k = (y(1), \dots, y(N_s))^T \tag{16}$$

and the regressors for each measurement be collected in the matrix Φ_k, i.e.

$$\Phi_k = \begin{pmatrix} \varphi^T(x_1) \\ \vdots \\ \varphi^T(x_{N_s}) \end{pmatrix} \tag{17}$$

and from (10),

$$\varphi(x_i) = (\mu_1(x_i), \dots, \mu_N(x_i))^T \, . \tag{18}$$

Given that there is an estimate $\hat{\theta}_{k-1}$ of the parameter vector the task is to compute a new estimate of the parameter vector by minimizing (from (14))

$$\hat{\theta}_k = \arg\min_{\theta} \|Y_k - \Phi_k \theta\|_V^2 + \|\theta\|_{W_1}^2 + \|\theta - \hat{\theta}_{k-1}\|_{W_2}^2 \quad (19)$$

Straightforward minimization with respect to θ gives

$$\theta_k = Q_k(\theta_{k-1} + L_k E_k) \quad (20)$$

where

$$Q_k = (\Phi_k^T V \Phi_k + W_1 + W_2)^{-1}(W_2 + \Phi_k^T V \Phi_k) \quad (21)$$

$$L = (W_2 + \Phi_k^T V \Phi_k)^{-1} \Phi_k^T V \quad (22)$$

and

$$E_k = Y_k - \Phi \hat{\theta}_{k-1}. \quad (23)$$

To illustrate the approach consider the case when the friction is modeled as Coulomb friction with the same friction coefficient in both directions, as in (12). In this case θ is the scalar f_c and Φ_k consists of the function sign(v) for the N_s measurements. This gives that $\Phi^T \Phi = N_s$. Assume for simplicity that $V = 1$ and that $W_1 = \rho$ and $W_2 = \lambda$. Equations (21) and (22) give

$$Q = \frac{\lambda + N_s}{\rho + \lambda + N_s} \quad (24)$$

and

$$L_k = \frac{\Phi_k^T}{\lambda + N_s}. \quad (25)$$

In (24) one can see that choosing $\rho = 0$, i.e. no regularization, gives $Q = 1$ which means no forgetting in the update of the parameter estimate, and vice versa. With $\rho = 0$ the update equation becomes,

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \frac{\Phi_k^T}{\lambda + N_s} E_k. \quad (26)$$

Finally, setting $\lambda = 0$, i.e. no penalty on the change of the parameter estimate, the estimation of the friction coefficient becomes a one-step procedure, where the estimate is computed as

$$\hat{\theta}_k = \frac{1}{N_s} \Phi_k^T Y_k. \quad (27)$$

6. SIMULATION EXAMPLE

To illustrate the approach presented above and compare it with conventional ILC a simulation example will be presented, based on a simulation model taken from Norrlöf and Gunnarsson (1997). A one degree-of-freedom robot arm is described by,

$$J\ddot{x}(t) = u(t) - f_c sign(\dot{x}(t)) - f_v \dot{x}(t) \quad \dot{x}(t) \neq 0 \quad (28)$$

and

$$J\ddot{x}(t) = 0 \quad | u(t) | \leq f_c \quad \dot{x}(t) = 0. \quad (29)$$

The variable $x(t)$ represents the arm angle, which is the controlled output, while the measured output is

$$y(t) = x(t) + v(t), \quad (30)$$

where $v(t)$ is a measurement noise with variance $2.5 \cdot 10^{-5}$. The arm is subjected to both viscous and Coulomb friction. When the velocity is zero the input torque has to exceed a certain level in order for the arm to move. In the model J denotes the moment of inertia, f_v and f_c denote the friction coefficients of the viscous and Coulomb friction respectively. In the simulations the values $J = 0.0094$, $f_c = 0.2$, and $f_v = 0.01$ are used. The arm is controlled using a combination of feed-forward and feed-back control. The feed-back is a PD-controller with the discrete-time transfer function

$$F(z) = K_P + \frac{K_D}{T} \frac{(z-1)}{z} \quad (31)$$

with parameter values $K_P = 12.7$ and $K_D = 0.4$. The feed-forward controller is given by

$$F_f(z) = \frac{J^*(z-1)^2}{T^2 z^2} \quad (32)$$

where J^* is an estimate of the moment of inertia of the arm. The measurements of the arm angle contain measurement noise with variance $2.5 \cdot 10^{-5}$.

The simulation is carried out using the reference signal shown in Figure 3, causing the arm to move back and forth.

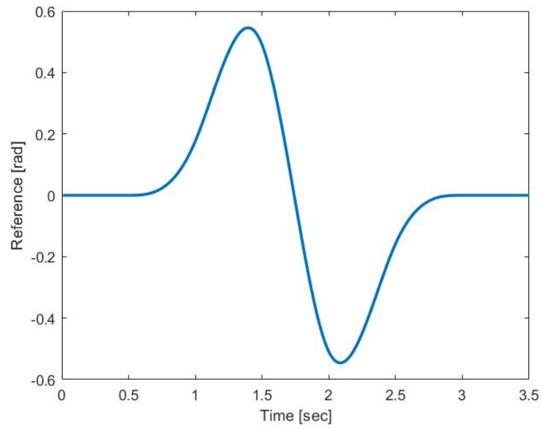

Fig. 3. Reference signal used in the simulation.

Three different tests are carried out to compare the performance of three different approaches:

I Conventional ILC using the ILC algorithm (2) with $Q(q)$ based on a second order Butterworth low pass filter with cut-off frequency $\omega_B = 15$ rad/s. Forward-backward filtering is used in order to obtain a zero phase filtering. The filer $L(q)$ consists of a gain and a time delay, which means

$$L(q) = \gamma q^{\delta} \quad (33)$$

In the simulations the values $\gamma = 0.5$ and $\delta = 2$ have been used.

II Parametric ILC, i.e. the first step in the two-step ILC algorithm as described above. Since the purpose is to illustrate the approach a very simple model structure for the feed-forward term is used. The friction is modeled as

$$\tau_F = f_c sign(\dot{y}) \quad (34)$$

and the friction model, i.e. the parameter f_c, is estimated according the procedure presented in Section 5.

III A combination of I and II.

Figure 4 shows the norm of the error between the reference signal and the controlled output, i.e.

$$\|r - x_k\|_2$$

This means that the error is evaluated using the controlled plant output, i.e. without measurement noise. The measured output, including the measurement noise, is used in the feed-back loop and in the different ILC algorithms. In Figure 5 the error including the measurement noise is shown in iteration 0 and 5 for the conventional ILC and the combined ILC using one estimation step and then conventional ILC. The error is significantly reduced after 5 iterations with both approaches. Clearly the combination of 1 iteration estimation and conventional ILC gives a smaller error compared to conventional ILC only which can be also be seen in Figure 4.

Two observations can be done from the results in Figure 4. First, the initial error reduction is significantly improved by using the parametric ILC and a small improvement can be seen by using two instead of only one estimation iterations, due to the measurement noise. Second, the final error using the conventional ILC is higher compared to the combined approach, using the parametric ILC and the conventional ILC in series. With approach II, the error is not reduced after the parameter estimation is ready and the feed-forward of the friction model is applied to the system. Combining parametric ILC and conventional ILC however, reduced the error faster than only using conventional ILC and the error also reaches a lower level. With measurement noise added to the measurements zero error will not be possible to achieve, even with $Q = 1$, and the choice of Q will also impact how the noise is accumulated in the ILC algorithm. The choice used in the simulation is a trade-off between noise level accumulation and reduction of the error.

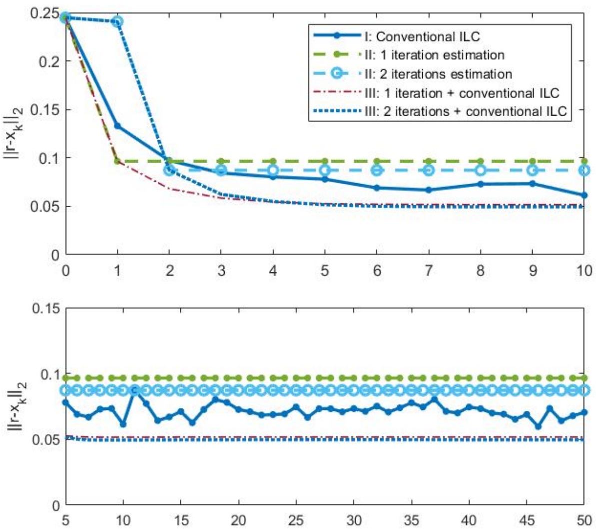

Fig. 4. Five different simulations are performed using the three approaches, conventional ILC (I), parametric ILC using one or two estimation iterations (II), finally a combined approach where a one and two iterations parametric ILC is used together with the conventional ILC algorithm (III). In the upper diagram the first 10 iterations are shown while in the lower the 5th until the 50th iteration are displayed to show the long term behavior of the different algorithms.

The results presented here represent initial results when the parametric approach is combined with conventional ILC, and future research will be needed to investigate the full potential of the approach.

7. CONCLUSION

An iterative, learning based, feed-forward method for compensation of friction in industrial robots has been studied. The method was put into an ILC framework by using the two step procedure proposed in Steinhauser (2019). The friction compensation method is based on a black-box friction model which is learned from operational data, and this can be seen as the first step in the method from Steinhauser (2019). In the second step the learned model is used for compensation of the friction using the reference joint velocity as input. The approach has been supported by simulation experiments in which the parametric approach is compared with and combined with conventional ILC.

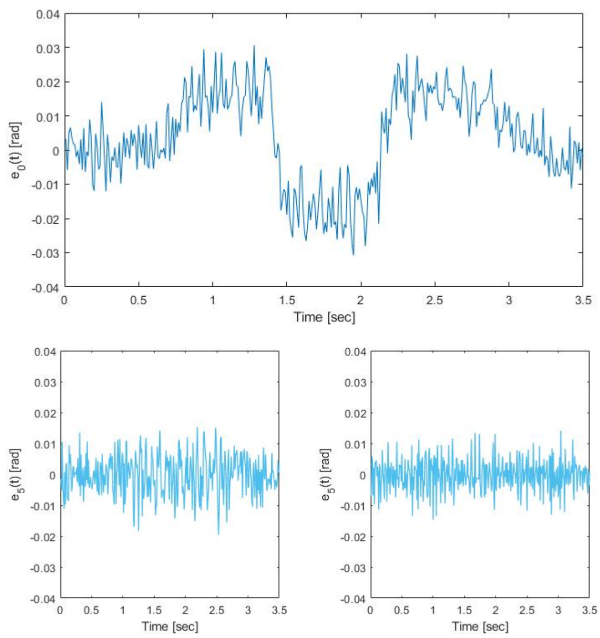

Fig. 5. Error $e_k(t) = r(t) - y_k(t)$ in iteration 0 (top) and iteration 5 for the conventional ILC (lower left) and the combined approach using 1 iteration estimation combined with conventional ILC (lower right). The error is reduced significantly after 5 iterations, especially considering the low frequency content.

REFERENCES

Ahn, H., Chen, Y., and Moore, K. (2007). Iterative learning control: Brief survey and categorization. *IEEE Transactions on System, Man, and Cybernetics - Part C: Applications and Reviews*, 37(6), 1099–1121.

Al-Bender, F. and Swevers, J. (2008). Characterization of friction force dynamics. *Control Systems Magazine*, 26(6).

Arimoto, S., Kawamura, S., and Miyazaki, F. (1984). Bettering Operation of Robots by Learning". *Journal of Robotic Systems*, 123–140.

Armstrong-Hélouvry, B. (1991). *Control of Machines with Friction*. Springer.

Bittencourt, A. and Gunnarsson, S. (2012). Static friction in a robot joint - modeling and identification of load and temperature effects. *Journal of Dynamic Systems, Measurements, and Control*, 135(4).

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

Driessen, B. and Sadegh, N. (2004). Convergence theory for multi-input discrete-time iterative learning control with Coulomb friction, continous outputs, and input bound. *International Journal of Adaptive Control and Signal Processing*, 18, 457–471.

Fleischer, M. (1997). Friktionsskattning i ett industrirobotsystem. In Swedish. Technical report, LiTH-ISY-EX-1777, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Harnoy, A., Friedland, B., and Cohn, S. (2008). Modeling and measuring friction effects. *IEEE Control Systems Magazine*, 28(6).

Johansson, V. (2017). Learning Local Models of Multi-axis Robot for Improved Feed-Forward Control. Technical report, Master's Thesis 2017: EX029, Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden.

Johansson, V., Moberg, S., Hedberg, E., Norrlöf, M., and Gunnarsson, S. (2018). A learning approach for feed-forward friction compensation. In *12th IFAC Symposium on Robot Control SYROCO*.

Längkvist, M. (2009). Online identification of friction coefficients in and indusstrial robot. Technical report, LiTH-ISY-EX-4259, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

Norrlöf, M. and Gunnarsson, S. (1997). Using iterative learning control to get better performance of robot control systems. Technical report, LiTH-ISY-R-1955, Department of Electrical Engineering, Linköping University, Linköping, Sweden.

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

Steinhauser, A. (2019). *Optimal iterative learning control for mechatronic systems. Theory, software, and application.* Ph.D. thesis, Faculty of Engineering Science, KU Leuven, Leuven, Belgium.

Stotsky, A. (2007). Adaptive estimation of the engine friction torque. *European Journal of Control*, 13, 618–624.

Åstrom, K.J. and de Wit, C.C. (2008). Revisiting the lugre friction model. *IEEE Control Systems*, 28(6), 101–114. doi:10.1109/MCS.2008.929425.

Tsai, M., Lin, M., and Yau, H. (2006). Development of command-based Iterative learning control algorithm with consideration of fricton, disturbance, and noise effects. *IEEE Transactions on Control Systems Technology*, 14(3), 511–518.

Uchiyama, M. (1978). Formation of high-speed motion pattern of a mechanical arm by trial. *Transaction of the Society of Instrument and Control Engineers*, 14, 706–712.

Volckaert, M. (2012). *Nonlinear iterative learning control through dynamic optimization.* Ph.D. thesis, Faculty of Engineering Science, KU Leuven, Leuven, Belgium.

Wang, Y. and Longman, R.W. (1994). Limit cycle behavior and convergence to zero error in learning control with stick-slip friction. In *IEEE Conference on Systems, Man, and Cybernetics*.