2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Estimation-based ILC using Particle Filter with Application to Industrial Manipulators

Patrik Axelsson, Rickard Karlsson, and Mikael Norrlöf

***Abstract*— An estimation-based *iterative learning control* (ILC) algorithm is applied to a realistic industrial manipulator model. By measuring the acceleration of the end-effector, the arm angular position accuracy is improved when the measurements are fused with motor angle observations. The estimation problem is formulated in a Bayesian estimation framework where three solutions are proposed: one using the *extended Kalman filter* (EKF), one using the *unscented Kalman filter* (UKF), and one using the *particle filter* (PF). The estimates are used in an ILC method to improve the accuracy for following a given reference trajectory. Since the ILC algorithm is repetitive no computational restrictions on the methods apply explicitly. In an extensive Monte Carlo simulation study it is shown that the PF method outperforms the other methods and that the ILC control law is substantially improved using the PF estimate.**

I. Introduction

A typical control configuration in modern industrial robots is to use measurements only from the motor angles of the manipulator in the controller. As a result of the development of cost efficient manipulators the mechanical structure has become less rigid. One component that contribute significantly to the flexibility of the manipulator is the gearbox, were the joint position deviates from the actuator position, hence the controller using only actuator positions is insufficient. To achieve better performance, the joint positions or tool position have to be measured directly or estimated. Here, the focus is on estimation of the joint positions.

Traditionally, many non-linear estimation problems are solved using the *extended Kalman filter* (EKF) [2]. In [14] an EKF is used to improve the trajectory tracking for a rigid 2-*degree-of-freedom* (DOF) robot. The robot dynamics are non-linear and the measurement noise is not always Gaussian. Hence, linearised models may not always be a good approach. The *particle filter* (PF) [9], [10] provides a general solution to many problems where linearisation and Gaussian approximations are intractable or would yield too low performance. The PF method is also motivated since it provides the possibility to design control laws and perform diagnosis in a much more advanced way. The use of Bayesian recursive estimation methods such as EKF and PF has been evaluated in simulations in e.g. [18], [22]. The recent publications [5], [6] present results based on real data from an ABB industrial robot where actuator positions and acceleration of the tool are fused using an EKF and PF.

Here we focus on the next step, i.e., to use the sensor fusion framework and the estimated joint position or the estimated tool position for control. In this contribution it is shown how the improved estimates can be used in an *iterative learning control* (ILC) method, [3], [21]. An early approach to estimation-based ILC is presented in [11], which presents a simulation study where measurements of the actuator position are combined with the arm acceleration of a single link robot. The corresponding approach is implemented and evaluated experimentally in [12]. For realistic robot structures the state estimation problems becomes a challenging non-linear filtering problem. The combination of non-linear state estimation and ILC is evaluated with promising results in [25] using a realistic 2-DOF simulation model containing mechanical elasticities. In [25] only an EKF was used in the estimation, it was also assumed that the measurement noise was Gaussian. Here both EKF, UKF, and PF are used, together with non-Gaussian noise distributions. For simplicity, here a 1-DOF model is used, but extension to several DOFs is straightforward.

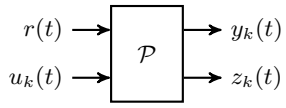

Fig. 1. System $\mathcal{P}$ with reference $r(t)$, ILC input $u_k(t)$, measured variable $y_k(t)$ and controlled variable $z_k(t)$ at ILC iteration k and time t.

II. Estimation-based ILC

Iterative learning control (ILC) is a way to improve the performance of systems that perform the same task repeatedly. Here it is assumed that the controlled variable cannot be directly measured. The situation is schematically described in Fig 1, where $r(t)$ and $u_k(t)$ denote the reference signal and the ILC input at iteration k, respectively. The system $\mathcal{P}$ can represent an open loop system as well as a closed loop system with internal feedback. Further details are given in [26]. There are two types of output signals from the system, $z_k(t)$ denotes the controlled variable and $y_k(t)$ the measured variable at ILC iteration k. In general the tool position is not directly measurable, therefore the ILC algorithm has to rely on estimates of the controlled variable based on measurements of related quantities. The measured variables are collected in the variable $y_k(t)$. In Fig 2 the actual control structure used for the ILC algorithm presented

*This work was supported by the Vinnova Excellence Center LINK-SIC.
P. Axelsson and M. Norrlöf are with Division of Automatic Control, Department of Electrical Engineering, Linköping University, SE-58183 Linköping, Sweden, {axelsson, mino}@isy.liu.se.
R. Karlsson is with Division of Automatic Control, Department of Electrical Engineering, Linköping University, SE-58183 Linköping, Sweden and Nira Dynamics, Teknikringen 6, SE-583 30 Linköping, Sweden, rickard.karlsson@niradynamics.se

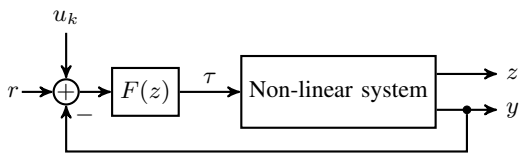

Fig. 2. ILC control structure used in the example in the paper.

in this paper is shown. The control input u_k from the ILC algorithm is added to the reference, which in turn drives the feedback loop with the controller F and the non-linear system.

The ILC algorithm used in this paper is given by the discrete-time filter

$$u_{k+1}(t) = \mathcal{Q}(q)\big(u_k(t) + \mathcal{L}(q)\epsilon_k(t)\big), \tag{1}$$

where q is the time-shift operator, t is time and k is the ILC iteration index. The filters $\mathcal{Q}(q)$ and $\mathcal{L}(q)$ are possibly non-causal. If the filter $\mathcal{Q}(q)$ is omitted and the filter $\mathcal{L}(q)$ is chosen as the inverse of the system, then the error converges to zero in one iteration. However, inverting the system will be very sensitive to model errors and may result in a very complicated ILC filter. Instead, the choice $\mathcal{L}(q) = \gamma q^{\delta}$ is often used, where $0 < \gamma < 1$ and δ a positive integer are the design variables. Moreover, the filter $\mathcal{Q}(q)$ is introduced in order to restrict the high frequency influence from the error and also reduce the influence of measurement noise. Including $\mathcal{Q}(q)$ makes the ILC algorithm converging slower and to a non zero error. Other choices of ILC updating laws can be found in [1], [7], [21]. The error used in the ILC algorithm is the difference between the reference r and the estimate $\hat{z}_k$ of the controlled variable at iteration k,

$$\epsilon_k(t) = r(t) - \hat{z}_k(t). \tag{2}$$

The estimation procedure is described in detail in Section III. For every ILC iteration k, the system is simulated using a new noise realisation for both the process and measurement noise. For evaluation of the performance of the ILC algorithm, the true error is used,

$$\varepsilon_k(t) = r(t) - z_k(t). \tag{3}$$

III. Bayesian Estimation

Consider the discrete-time state-space model

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t), \tag{4a}$$
$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{e}_t, \tag{4b}$$

with state variables $\mathbf{x}_t \in \mathbb{R}^n$, input signal $\mathbf{u}_t$ and measurements $\mathbb{Y}_t = \{\mathbf{y}_i\}_{i=1}^t$, with known *probability density functions* (PDFs) for the process noise, $p_w(\mathbf{w})$, and measurement noise $p_e(\mathbf{e})$. The notation $\mathbf{x}_t = \mathbf{x}(t)$ will be used alternately in the sequel of the paper to denote time dependency, not to be confused with the subscript k used to indicate the ILC iteration. Moreover, boldface lower-case variables indicate vectors and boldface upper-case variables indicate matrices.

The non-linear posterior prediction density $p(\mathbf{x}_{t+1}|\mathbb{Y}_t)$ and filtering density $p(\mathbf{x}_t|\mathbb{Y}_t)$ for the Bayesian inference [15] are given by

$$p(\mathbf{x}_{t+1}|\mathbb{Y}_t) = \int_{\mathbb{R}^n} p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\mathbb{Y}_t)d\mathbf{x}_t, \tag{5a}$$
$$p(\mathbf{x}_t|\mathbb{Y}_t) = \frac{p(\mathbf{y}_t|x_t)p(\mathbf{x}_t|\mathbb{Y}_{t-1})}{p(\mathbf{y}_t|\mathbb{Y}_{t-1})}. \tag{5b}$$

For the special case of linear-Gaussian dynamics and linear-Gaussian observations, the Kalman filter [17] is the optimal solution. For non-linear and non-Gaussian systems, the PDF cannot, in general, be expressed with a finite number of parameters. Instead approximative methods are used. This is usually done in two ways; either by approximating the system or by approximating the posterior PDF, see for instance, [4]. Here, three different approaches of solving the Bayesian equations are considered; EKF, UKF, and PF. The EKF will solve the problem using a linearisation of the system and assuming Gaussian noise. The UKF will also assume Gaussian noise but keeping the non-linearity as it is. The PF on the other hand will approximately solve the Bayesian equations by stochastic integration. Hence, no linearisation errors occur. The PF can also handle non-Gaussian noise models where the PDFs are known only up to a normalization constant. Also, hard constraints on the state variables can easily be incorporated in the estimation problem.

In the ILC application, due to the repetitive nature of the algorithm, no computational restrictions on the methods apply explicitly, hence in this paper only tracking performance is studied.

A. The Extended Kalman Filter (EKF)

For many non-linear problems, the noise assumptions and the non-linearity are such that a linearised solution will be a good approximation. This is the idea behind the EKF [2] where the model is linearised around the previous estimate. The time update and measurement update for the EKF are

$$\begin{cases} \hat{\mathbf{x}}_{t+1|t} = f(\hat{\mathbf{x}}_{t|t}, \mathbf{u}_t, 0), \\ \mathbf{P}_{t+1|t} = \mathbf{F}_t\mathbf{P}_{t|t}\mathbf{F}_t^T + \mathbf{G}_t\mathbf{Q}_t\mathbf{G}_t^T, \end{cases} \tag{6a}$$

$$\begin{cases} \hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - h(\hat{\mathbf{x}}_{t|t-1})), \\ \mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{H}_t\mathbf{P}_{t|t-1}, \\ \mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^T(\mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^T + \mathbf{R}_t)^{-1}, \end{cases} \tag{6b}$$

where the linearised matrices are given as

$$\mathbf{F}_t = \nabla_{\mathbf{x}} f(\mathbf{x}_t, \mathbf{u}_t, 0)|_{\mathbf{x}_t=\hat{\mathbf{x}}_{t|t}}, \ \mathbf{H}_t = \nabla_{\mathbf{x}} h(\mathbf{x}_t)|_{\mathbf{x}_t=\hat{\mathbf{x}}_{t|t-1}},$$
$$\mathbf{G}_t = \nabla_{\mathbf{w}} f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t)|_{\mathbf{x}_t=\hat{\mathbf{x}}_{t|t}}.$$

In (6), $\mathbf{P}$ denotes the covariance matrix for the estimation error, and the noise covariances are given as

$$\mathbf{Q}_t = \mathrm{Cov}\left(\mathbf{w}_t\right), \ \mathbf{R}_t = \mathrm{Cov}\left(\mathbf{e}_t\right), \tag{7}$$

where the process noise and measurement noise are assumed to be zero mean Gaussian distributions.

B. The Unscented Kalman Filter (UKF)

The UKF, [16], utilises the unscented transform and avoids linearisation by propagating samples, so-called sigma points, through the dynamic model and in a similar way evaluates these samples using the measurement relation. These are chosen deterministically as the mean value and values equally spaces determined by the covariance. Note that the UKF always approximates the underlying PDF with a Gaussian distribution. For details see [13], [16]. The advantage over the EKF is that no linearisation is needed.

C. The Particle Filter (PF)

The PF, [9], [10], provides an approximate solution to the Bayesian estimation problem formulated in (5). The PF approximates the density $p(\mathbf{x}_t|\mathbb{Y}_t)$ by a large set of N samples (particles), $\{\mathbf{x}_t^{(i)}\}_{i=1}^N$, where each particle has an assigned relative weight, $\omega_t^{(i)}$, chosen such that all weights sum to unity. The location and weight of each particle reflect the value of the density in the region of the state space. The PF updates the particle location in the state space and the corresponding weights recursively with each new observed measurement. Using the samples (particles) and the corresponding weights, the Bayesian equations can be approximately solved. To avoid divergence, a resampling step is introduced, [10]. There exist theoretical limits [9] that the approximated PDF converges to the true as the number of particles go to infinity.

The estimate for each time, t, is often chosen as the minimum mean square estimate, i.e.,

$$\hat{\mathbf{x}}_{t|t} = \mathbb{E}\left(\mathbf{x}_t\right) = \int_{\mathbb{R}^n} \mathbf{x}_t p(\mathbf{x}_t|\mathbb{Y}_t) d\mathbf{x}_t \approx \sum_{i=1}^{N} \omega_t^{(i)} \mathbf{x}_t^{(i)}, \quad (8)$$

but other choices are possible.

D. Cramér-Rao Lower Bound (CRLB)

When different estimators are used, it is fundamental to know the best possible achievable performance. As mentioned previously, the PF will approach the true PDF asymptotically. In practice only approximations are possible since the number of particles are finite. For other estimators, such as the EKF, it is important to know how much the linearisation or model structure used, will affect the performance. The *Cramér-Rao lower bound* (CRLB) is such a characteristic for the second order moment [8], [19]. The theoretical posterior CRLB for a general dynamic system was derived in [9], [23], [24]. In this paper we will utilize a parametric CRLB, which can be found as the EKF solution around the true known state trajectory for the case with Gaussian measurement noise and process noise. When the measurement noise is non-Gaussian this can be handled using the relative accuracy presented in detail in [13].

IV. Dynamic Models

In this section a continuous-time robot model is discussed. The model is transformed into discrete time, where it can be used by the EKF, UKF, and PF. The measurements are the angle measurements from the motors and the acceleration at the end-effector.

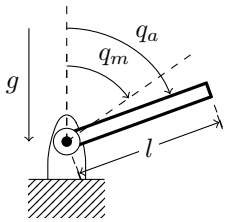

Fig. 3. Single flexible joint corresponding to joint two of a six-axes industrial manipulator.

A. Robot Model

The robot model is a flexible single joint model, see Fig 3. The model is based on the two-axes model in [20], where the parameters are presented. For the two-axes model the second joint has been positioned in $-\pi/2$ and rigidly attached to the first arm, giving a single joint model. The link is modelled as a rigid-body. The joint flexibility is modelled as a spring-damper pair with non-linear spring torque $\mathcal{T}(\,\cdot\,)$ and linear damping. The deflection in each joint is given by the arm angle q_a and the motor angle q_m. The motor characteristics are given by a non-linear friction torque $\mathcal{F}(\,\cdot\,)$. The dynamical relation for the joint is described by

$$M_a\ddot{q}_a + \mathcal{T}(q_a, q_m) + \mathcal{D}(\dot{q}_a - \dot{q}_m) = 0, \quad (9a)$$

$$M_m\ddot{q}_m - \mathcal{T}(q_a, q_m) - \mathcal{D}(\dot{q}_a - \dot{q}_m) + \mathcal{F}(\dot{q}_m) = \tau, \quad (9b)$$

where M_a and M_m are the inertia for the arm and motor, respectively, $\mathcal{D}$ is the damping coefficient, and τ is the applied motor torque. Using the state vector $\mathbf{x} = \begin{pmatrix} q_a & q_m & \dot{q}_a & \dot{q}_m \end{pmatrix}^T$ and (9) give a non-linear state space model according to

$$\dot{\mathbf{x}} = \underbrace{\begin{pmatrix} x_3 \\ x_4 \\ M_a^{-1}(-\mathcal{T}(x_1, x_2) - \mathcal{D}(x_3 - x_4)) \\ M_m^{-1}(\tau + \mathcal{T}(x_1, x_2) + \mathcal{D}(x_3 - x_4) - \mathcal{F}(x_4)) \end{pmatrix}}_{\tilde{f}(\mathbf{x},\tau)}. \quad (10)$$

B. Estimation Model

The estimation methods, described in Section III, need a discrete-time state space model according to (4). Using Euler sampling

$$\dot{\mathbf{x}} \approx \frac{\mathbf{x}_{t+1} - \mathbf{x}_t}{T_s}, \quad (11)$$

and assuming that the process noise enters the model in the same way as the motor torque, give the discrete-time model

$$\mathbf{x}_{t+1} = \mathbf{x}_t + T_s\tilde{f}(\mathbf{x}_t, \tau_t + w_t) = f(\mathbf{x}_t, \tau_t, w_t). \quad (12)$$

C. Observation Model

The observations used for estimation is the motor angle q_m and the rotational acceleration of the tool position $a_{\text{TCP}} = l\ddot{q}_a$, where l is the length of the arm and $\ddot{q}_a$ is given by the third row in the state space model (10). The observation relation can now be written as

$$h(\mathbf{x}_t) = \begin{pmatrix} q_m(t) + e_m(t) \\ a_{\text{TCP}}(t) + e_{a_{\text{TCP}}}(t) \end{pmatrix}, \quad (13)$$

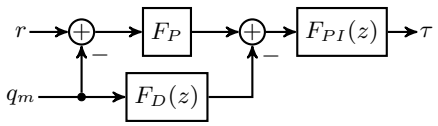

Fig. 4. PID controller in interacting form.

where $e_m(t)$ and $e_{a_{\text{TCP}}}(t)$ are the measurement noise, described in more details in Section V-A.

V. Results

In this section, the simulation results, both for the estimation performance and the ILC performance, are presented.

A. Simulation Setup

The model parameters given in [20] are used with a spring described by

$$k_1^{high} = 0.6 \cdot 10^6, \quad k_1^{low} = 0.2 \cdot 10^6, \quad \psi_1 = 20. \tag{14}$$

The robot model in Section IV is unstable, hence a feedback controller is required for simulation. The performance of the controller is not important here since the ILC control law will improve the performance. The robot model is stabilized using the feedback loop, shown in Fig 4, consisting of a PID controller in interacting form. The three parts of the PID controller are, in continuous time, chosen as

$$F_P = 10, \quad F_D(s) = \frac{s}{0.001s+1}, \quad F_{PI}(s) = \frac{0.9s+3}{0.3s},$$

and then discretized using Tustin's formula with a sample time of $T_s = 1\,\text{ms}$. Note that the feedback loop is not tuned to give good performance. The reference signal r is chosen as a step filtered through a FIR filter of length $n = \lfloor 10 \times 0.01/T_s \rfloor = 100$ with all coefficients equal to $1/n$. The filtering is performed in 4 consecutive steps to get a twice differentiable smooth reference signal. Moreover, process noise with the distribution $p_w(w) = \mathcal{N}(0, Q)$ is added during the simulation, where $Q = 10^6$.

The measurement noise e_m and $e_{a_{\text{TCP}}}$ are chosen to resemble quantization errors according to

$$e = \begin{cases} -\zeta, & \text{with probability } 1/3 \\ 0, & \text{with probability } 1/3 \\ \zeta, & \text{with probability } 1/3 \end{cases}, \tag{15}$$

where $\zeta = 0.1$ for e_m and $\zeta = 10$ for $e_{a_{\text{TCP}}}$. For the PF the distribution $p_e(e)$ is not exactly known. It is known that three peaks are present but the actual values are not known. It is assumed that the peaks are in the positions -0.8ζ, 0, and 0.8ζ. To handle expected modelling errors, a Gaussian kernel is placed at the positions -0.8ζ, 0, and 0.8ζ, giving the Gaussian mixture distribution

$$p_e(e) = \sum_{i=1}^{3} \frac{1}{3} \mathcal{N}(e|\mu_i, \sigma^2), \tag{16}$$

where $\mu_i \in \{-0.8\zeta,\, 0,\, 0.8\zeta\}$, and the standard deviations are chosen as $\sigma = 14 \cdot 10^{-3}$ for e_m and $\sigma = 1.4$ for $e_{a_{\text{TCP}}}$.

Fig 5 shows the true measurement noise as vertical bars together with the Gaussian mixture. The EKF, which cannot use the true distribution, uses a Gaussian approximation of the Gaussian mixture distribution giving the density $p_e(\mathbf{e}) = \mathcal{N}(\mathbf{0}, \mathbf{R})$ where $\mathbf{R} = \frac{128}{3}\text{diag}(10^{-4}, 1)$.

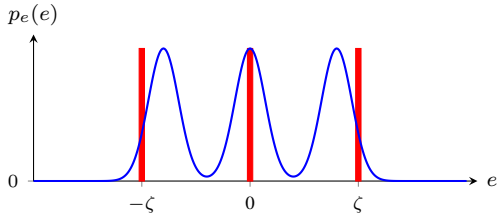

Fig. 5. Distribution for the measurement noise as vertical bars together with a Gaussian mixture which is used in the PF. Note that the PDFs are not correctly normalized, the figure shows only the concept.

B. Monte Carlo Estimations and Performance Evaluation

The performance of the EKF, UKF, and PF is evaluated using the *root mean square error* (RMSE)

$$\text{RMSE}(t) = \left(\frac{1}{N_{\text{MC}}} \sum_{j=1}^{N_{\text{MC}}} \|\mathbf{x}_t^{\text{TRUE}} - \hat{\mathbf{x}}_t^{(j)}\|_2^2 \right)^{1/2}, \tag{17}$$

over $N_{\text{MC}} = 1000$ *Monte Carlo* (MC) simulations. The RMSE is compared to the CRLB, described in Section III-D, using the following relationship

$$\text{RMSE}(t) \geq \sqrt{\text{tr}\, P_{\text{CRLB}}(\mathbf{x}_t^{\text{TRUE}})}, \tag{18}$$

where tr is the trace operator and $P_{\text{CRLB}}(\mathbf{x}_t^{\text{TRUE}})$ is the covariance matrix given from the CRLB calculations using the true state trajectory. The covariance matrix of the measurement noise used for the CRLB, denoted as R^{CRLB}, is calculated as $R^{\text{CRLB}} = \text{cov}(\mathbf{e})\Psi^{-1}$, where Ψ is the relative accuracy presented in [13]. Here numerical calculations give $\Psi \approx 35$ for both measurements where $p(\mathbf{e})$ is the Gaussian mixture in (16) using $\mu_i \in \{-\zeta,\, 0,\, \zeta\}$. Note that we are here only looking at the RMSE for q_a, hence only q_a^{TRUE} and $\hat{q}_a$ are used in (17) and only the (1,1)-element of P_{CRLB} is used in (18). Fig 6 shows the RMSE for the EKF, UKF, and PF. The CRLB is also included. It can be seen that the RMSE for the EKF and UKF are very similar. We also see that the RMSE for the PF is close to the CRLB whereas RMSE for EKF and UKF are significantly higher. The PF used $N = 10000$ particles too ensure that this should not effect the performance. In this paper we have not addressed the issue how to reduce the number of particles in order to achieve a less computational demanding algorithm since the estimation can be done off-line. The estimated arm angle for one of the 1000 MC simulations is shown in Fig 7 after the step has occurred. The motor angle q_m is also included to show how different it is to the arm angle q_a. The figure also shows the distribution of all particles visualised as the grey area. The particle distribution shows that the true q_a is almost covered all the time by the posterior distribution even if the mean estimate given by the PF does not give the true trajectory. This detail will be discussed more in Section VI.

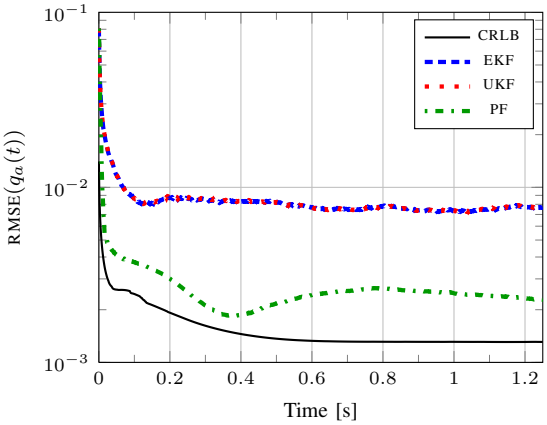

Fig. 6. RMSE over 1000 MC simulations. The performance for the EKF, UKF, and PF is compared to the CRLB.

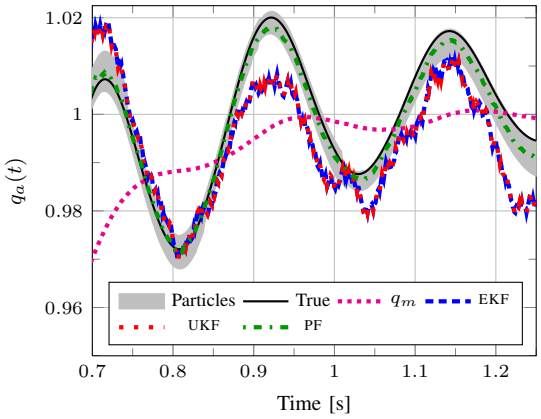

Fig. 7. Estimated q_a for EKF, UKF, and PF. The motor angle q_m is also included. The grey area shows the distribution of all the particles.

TABLE I

NUMERICAL VALUES FOR THE ILC FILTERS.

	γ	δ	f_c	n
q_m	0.95	90	5	2
q_a or $\hat{q}_a$	0.95	90	20	2

The similar performance of the EKF and UKF can be explained by the fact that both the EKF and UKF approximate the underlying distribution by a Gaussian distribution, together with that the linearisation errors in the EKF are small. Hence, the good performance of the PF comes from the fact that the posterior distribution is approximated by the particles, and not that no linearisation is performed.

C. Estimation-based ILC Simulations

In this section, the ILC update law from Section II will be used with five different ways to find $\hat{z}_k(t)$ in (2). The five different ways to find $\hat{z}_k(t)$ are

1) The motor angle q_m including measurement noise.
2) True arm angle q_a without any noise.
3) Estimated arm angle $\hat{q}_a$ using the EKF.
4) Estimated arm angle $\hat{q}_a$ using the UKF.
5) Estimated arm angle $\hat{q}_a$ using the PF.

Case 1 is included to show that a lot is gained if the ILC update law uses information from the variable that is to be controlled. For Case 1, the motor reference r_m is used in (2). It is here assumed that $r_m = \eta r_a$, where $\eta > 1$ is the gear ratio. Case 2 corresponds to the best solution that can be obtained with this framework.

The ILC update law described in Section II is used with $\mathcal{L} = \gamma q^\delta$ and $\mathcal{Q}$ as a nth order digital Butterworth filter with cut-off frequency f_c. The numerical values for the filters are presented in Table I for the two cases i) ILC using measured q_m, ii) ILC using q_a or an estimate of q_a. The numerical values for the ILC filters in Table I are chosen to give a satisfactory behaviour.

The performance of the five ILC algorithms is evaluated using the relative reduction error of the 2-norm in percentage with respect to the error when no ILC signal is applied, hence

$$\rho_k = 100 \frac{||\varepsilon_k(t)||_2}{||\varepsilon_0(t)||_2}. \tag{19}$$

The relative reduction error ρ_k averaged over 100 MC simulations is shown in Fig 8 for the five cases using 100 ILC iterations. It can be seen that ILC using the motor angles does not give a good result. Using the true q_a gives a lower bound of 0.088 %. The performance from $\hat{q}_a$ depends on the performance of the Bayesian filters, the PF gives a lower error than the EKF and UKF. The mean of the relative reduction, averaged over 100 MC simulations, from iteration 50 to iteration 100 for the five ILC algorithms are presented in Table II.

The final ILC signal at iteration $k = 100$, for one of the 100 MC simulations, is presented in Fig 9. We can see that the ILC signals for q_a and PF are very similar whereas the ILC signals for EKF and UKF are more oscillatory.

TABLE II

MEAN OF THE RELATIVE REDUCTION ERROR ρ_k, AVERAGED OVER 100 MC SIMULATIONS, FROM ITERATION $k = 50$ TO $k = 100$.

q_m [%]	q_a [%]	EKF [%]	UKF [%]	PF [%]
12.00	0.088	4.79	4.82	1.27

VI. EXTENSIONS

Due to non-linearities, non-Gaussian measurement noise, and hard constraints on the state vector, the posterior distribution is non-Gaussian. One extension to the ILC framework is to propagate the complete distribution, given by the particles and the corresponding weights, instead of only the mean estimate. In Fig 7 the particles cover the true trajectory most of the time even if the mean estimate differ from the true. It means that if the complete distribution is used, the ILC algorithm can converge faster and to a lower value. One of the difficulties is that the ILC control law is a filter, hence it is not possible to propagate the distribution at each time

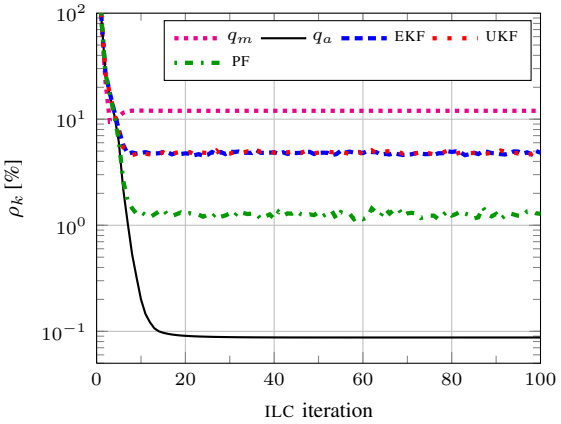

Fig. 8. The relative reduction error ρ_k in (19), averaged over 100 MC simulations, using 100 ILC iterations.

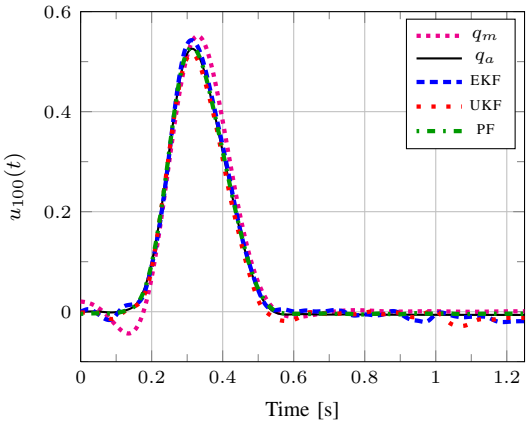

Fig. 9. The ILC signal u after 100 iterations for one of the 100 MC simulations. The signals for q_a and PF are very similar.

instance, instead complete trajectories are needed, which can be hard to get due to the degeneracy problem of the PF[1] [4].

Since the ILC estimations are off-line we have not considered the computational burden of the PF. The number of particles can be reduced and other resampling algorithms can be applied to speed up the computations. Also model errors effecting the robustness might be important but is outside the scope of this paper.

VII. Conclusions

In the paper we have presented Bayesian estimation techniques for improving the position accuracy of an industrial manipulator. The proposed PF method outperforms the EKF and the UKF when non-linearities and quantization noise are considered in an extensive Monte Carlo simulation. These estimates are used in an ILC control law, to improve the reference tracking. Furthermore, the proposed PF method enables extensions when hard constraints on the state vector are given. Also, propagation of the complete posterior distribution through the ILC update law can be considered.

[1] All or most of the particles at the end time will share a common ancestor, giving one single trajectory or a few trajectories at the beginning.

References

[1] D. A. Bristow, M. Tharayil, and A. G. Alleyne. A survey of iterative learning control - a learning-based method for high-performance tracking control. *IEEE Control Systems Magazine*, pages 96–114, June 2006.

[2] B. Anderson and J. B. Moore. *Optimal Filtering*. Prentice Hall, Englewood Cliffs, NJ, 1979.

[3] S. Arimoto, S. Kawamura, and F. Miyazaki. Bettering operation of robots by learning. *Journal of Robotic Systems*, 1(2):123–140, 1984.

[4] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Processing*, Feb. 2002.

[5] P. Axelsson. Evaluation of six different sensor fusion methods for an industrial robot using experimental data. In *Proc. of the 10th Int. IFAC Symp. on Robot Contr.*, pages 126–132, Dubrovnik, Croatia, Sep. 2012.

[6] P. Axelsson, R. Karlsson, and M. Norrlöf. Bayesian state estimation of a flexible industrial robot. *Control Engineering Practice*, 20(11):1220–1228, Nov. 2012. DOI: 10.1016/j.conengprac.2012.06.004.

[7] Z. Bien and J.-X. Xu. *Iterative Learning Control: Analysis, Design, Integration and Application*. Kluwer Academic Publishers, Boston, MA, 1998.

[8] H. Cramér. *Mathematical Methods of Statistics*. Princeton, NJ: Princeton University Press, 1946.

[9] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[10] N. J. Gordon, D. J. Salmond, and A. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.

[11] S. Gunnarsson and M. Norrlöf. Iterative learning control of a flexible mechanical system using accelerometers. In *IFAC 6th symposium on robot control, SYROCO*, Vienna, Austria, Sept. 2000.

[12] S. Gunnarsson, M. Norrlöf, E. Rahic, and M. Özbek. On the use of accelerometers in iterative learning control of a flexible robot arm. *International Journal of Control*, 80(3):363–373, Mar. 2007.

[13] G. Hendeby. *Performance and Implementaion Aspects of Nonlinear Filtering*. Linköping studies in science and technology. dissertations. no. 1161, Feb. 2008.

[14] R. Jassemi-Zargani and D. Necsulescu. Extended Kalman filter-based sensor fusion for operational space control of a robot arm. *IEEE Transactions on Instrumentation and Measurement*, 51(6):1279 – 1282, Dec 2002.

[15] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, 1970.

[16] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proc of the IEEE*, 92(3):401–422, Mar. 2004.

[17] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the AMSE–Journal of Basic Engineering*, 82:35–45, 1960.

[18] R. Karlsson and M. Norrlöf. Position estimation and modeling of a flexible industrial robot. In *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, July 2005.

[19] S. Kay. *Fundamentals of Statistical Signal Processing*. Prentice Hall, 1993.

[20] S. Moberg, J. Öhr, and S. Gunnarsson. A benchmark problem for robust control of a multivariable nonlinear flexible manipulator. In *Proc. of the 17th IFAC WC*, pages 1206–1211, Seoul, Korea, July 2008.

[21] K. L. Moore. *Iterative Learning Control for Deterministic Systems*. Advances in Industrial Control. Springer-Verlag, London, 1993.

[22] G. Rigatos. Particle filtering for state estimation in nonlinear industrial systems. *IEEE Transactions on Instrumentation and Measurement*, 58(11):3885–3900, Nov. 2009.

[23] P. Tichavsky, P. Muravchik, and A. Nehorai. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing*, 46(5):1386–1396, 1998.

[24] H. Van Trees. *Detection, Estimation and Modulation Theory*. Wiley, New York, 1968.

[25] J. Wallén, S. Gunnarsson, R. Henriksson, S. Moberg, and M. Norrlöf. ILC applied to a flexible two-link robot model using sensor-fusion-based estimates. In *Proc. IEEE Conf. Decision Control*, pages 458–463, Shanghai, China, dec 2009.

[26] J. Wallén, M. Norrlöf, and S. Gunnarsson. A framework for analysis of observer-based ILC. *Asian Journal Control*, 13(1):3–14, Jan. 2011.