

Bayesian Methods for Estimating Tool Position of an Industrial Manipulator

Patrik Axelsson, Rickard Karlsson, and Mikael Norrlöf

Abstract—State estimation of a flexible industrial manipulator is presented using experimental data. The problem is formulated in a Bayesian framework where the extended Kalman filter and particle filter are used. The filters use the joint positions on the motor side of the gearboxes as well as the acceleration at the end-effector as measurements and estimates the corresponding joint angles on the arm side of the gearboxes. The techniques are verified on a state of the art industrial robot, and it is shown that the use of the acceleration at the end-effector improves the estimates significantly.

I. INTRODUCTION

Modern industrial robot control is usually only based on measurements from the motor angles of the manipulator. However, the ultimate goal is to move the tool according to a predefined path. In [9] a method for improving the absolute accuracy of a standard industrial manipulator is described. The improved accuracy is achieved through identification of unknown or uncertain parameters in the robot system, and applying the *iterative learning control* (ILC) method, [2], [20], using additional sensors to measure the actual tool position. The aim of this paper is to evaluate Bayesian estimation techniques for sensor fusion and to improve the estimate of the tool position from measurements of the acceleration at the end-effector. The improved accuracy at the end-effector is needed in for instance laser cutting, where low cost sensors such as accelerometers are a feasible choice. The configuration of the system with the accelerometer is depicted in Fig 1.

Traditionally, many nonlinear Bayesian estimation problems are solved using the *extended Kalman filter* (EKF) [1], [13]. In [11] an EKF is used to improve the trajectory tracking for a rigid 2 *degree-of-freedom* (DOF) robot. The robot dynamics and measurements are highly nonlinear and the measurement noise is not always Gaussian. Hence, linearized models may not always be a good approach. The *particle filter* (PF) [7], [5] provides a general solution to many problems where linearizations 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. Bayesian sensor fusion techniques using particle filtering based on external sensors have so far been

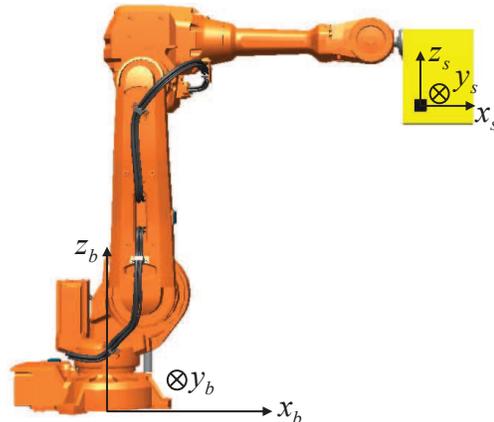


Fig. 1. The ABB IRB4600 robot with the accelerometer. The base coordinate system, (x_b, y_b, z_b) , and the coordinate system for the sensor (accelerometer), (x_s, y_s, z_s) , are also shown.

applied to very few industrial robotic applications, [15], [16], [21], focusing on evaluation and simulation. This paper extends the idea to experimental data, from a state of the art industrial robot, for evaluation of the EKF and the PF.

II. BAYESIAN ESTIMATION

Consider the discrete state-space model

$$x_{t+1} = f(x_t, u_t, w_t), \quad (1a)$$

$$y_t = h(x_t) + e_t, \quad (1b)$$

with state variables $x_t \in \mathbb{R}^n$, input signal u_t and measurements $\mathbb{Y}_t = \{y_i\}_{i=1}^t$, with known *probability density functions* (PDFs) for the process noise, $p_w(w)$, and measurement noise $p_e(e)$. The nonlinear prediction density $p(x_{t+1}|\mathbb{Y}_t)$ and filtering density $p(x_t|\mathbb{Y}_t)$ for the Bayesian inference [12] is given by

$$p(x_{t+1}|\mathbb{Y}_t) = \int_{\mathbb{R}^n} p(x_{t+1}|x_t)p(x_t|\mathbb{Y}_t) dx_t, \quad (2a)$$

$$p(x_t|\mathbb{Y}_t) = \frac{p(y_t|x_t)p(x_t|\mathbb{Y}_{t-1})}{p(y_t|\mathbb{Y}_{t-1})}. \quad (2b)$$

For the important special case of linear-Gaussian dynamics and linear-Gaussian observations the Kalman filter [14] will give the solution. For nonlinear and non-Gaussian systems, the PDF can in general not be expressed with a finite number of parameters. Instead approximative methods must be used, for instance the *extended Kalman filter* (EKF) and the *particle filter* (PF). The EKF will solve the problem using a linearization of the system assuming Gaussian noise,

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

where the PF on the other hand will approximately solve the Bayesian equations by stochastic integration. Hence, no linearizations errors occur. The PF can also handle non-Gaussian noise models and hard constraints on the state variables.

A. The Extended Kalman Filter (EKF)

For many nonlinear problems the noise assumptions and the nonlinearity are such that a linearized solution will be a good approximation. This is the idea behind the EKF [1], [13] where the model is linearized around the previous estimate. Here the time update and measurement update for the EKF are presented,

$$\begin{cases} \hat{x}_{t+1|t} = f(\hat{x}_{t|t}, u_t, 0), \\ P_{t+1|t} = F_t P_{t|t} F_t^T + G_t Q_t G_t^T, \end{cases} \quad (3a)$$

$$\begin{cases} \hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t (y_t - h(\hat{x}_{t|t-1})), \\ P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}, \\ K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1}, \end{cases} \quad (3b)$$

where the linearized matrices are given as

$$F_t = \nabla_x f(x_t, u_t, 0)|_{x_t = \hat{x}_{t|t}}, \quad (4a)$$

$$G_t = \nabla_w f(x_t, u_t, w_t)|_{x_t = \hat{x}_{t|t}}, \quad (4b)$$

$$H_t = \nabla_x h(x_t)|_{x_t = \hat{x}_{t|t-1}}, \quad (4c)$$

and the noise covariances are given as

$$Q_t = \text{Cov}(w_t), \quad R_t = \text{Cov}(e_t). \quad (5)$$

Note that the process noise and measurement noise are assumed zero mean processes.

B. The Particle Filter (PF)

The *particle filter* (PF), [7], [5], [22], provides an approximate solution to the discrete time Bayesian estimation problem formulated in (2) by updating an approximate description of the posterior filtering density. Let x_t denote the state of the observed system and $\mathbb{Y}_t = \{y_i\}_{i=1}^t$ be the set of observed measurements until present time. The PF approximates the density $p(x_t|\mathbb{Y}_t)$ by a large set of N samples (particles), $\{x_t^{(i)}\}_{i=1}^N$, where each particle has an assigned relative weight, $\gamma_t^{(i)}$, 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 and the corresponding weights recursively with each new observed measurement. Using the samples (particles) and the corresponding weights the Bayesian equations can be solved approximately. To avoid divergence a resampling step is introduced, [7]. The PF is summarized in Algorithm 1, where the proposal distribution $p^{\text{prop}}(x_{t+1}^{(i)}|x_t^{(i)}, y_{t+1})$ can be chosen arbitrary as long as it is possible to draw samples from it.

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

$$\hat{x}_{t|t} = \mathbb{E}(x_t) = \int_{\mathbb{R}^n} x_t p(x_t|\mathbb{Y}_t) dx_t \approx \sum_{i=1}^N \gamma_t^{(i)} x_t^{(i)}, \quad (6)$$

Algorithm 1 The Particle Filter

- 1: Generate N samples $\{x_0^{(i)}\}_{i=1}^N$ from $p(x_0)$.
- 2: Compute

$$\gamma_t^{(i)} = \gamma_{t-1}^{(i)} \cdot \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{p^{\text{prop}}(x_t^{(i)}|x_{t-1}^{(i)}, y_t)}$$

and normalize, i.e., $\bar{\gamma}_t^{(i)} = \gamma_t^{(i)} / \sum_{j=1}^N \gamma_t^{(j)}$, $i = 1, \dots, N$.

- 3: [Optional]. Generate a new set $\{x_t^{(i^*)}\}_{i=1}^N$ by resampling with replacement N times from $\{x_t^{(i)}\}_{i=1}^N$, with probability $\bar{\gamma}_t^{(i)} = Pr\{x_t^{(i^*)} = x_t^{(i)}\}$ and reset the weights to $1/N$.
 - 4: Generate predictions from the proposal distribution $x_{t+1}^{(i)} \sim p^{\text{prop}}(x_{t+1}^{(i)}|x_t^{(i^*)}, y_{t+1})$, $i = 1, \dots, N$.
 - 5: Increase t and continue to step 2.
-

but other choices, such as the ML-estimate, might be of interest. There exist theoretical limits [5] that the approximated PDF converges to the true as the number of particles tends to infinity.

III. DYNAMIC MODELS

In this section a continuous-time 2 DOF robot model is discussed. The model is simplified and transformed into discrete time, where it is used by the EKF and PF. The measurements are in both cases angle measurements from the motors, with acceleration information from the end-effector.

A. Robot Model

The robot model used in this work is a joint flexible two-axes model, see Fig 2. The model corresponds to axis 2 and 3 for a serial 6 DOF industrial robot. A common assumption of the dynamics of the robot is that the transmission can be approximated by two or three masses connected by springs and dampers. The coefficients in the resulting model can be estimated from an identification experiment, see for instance [17]. Here it will be assumed that the transmission can be described by a two mass system and that the links are rigid.

The dynamic model can be described by a torque balance for the motors and the arms. The equation describing the torque balance for the motor becomes

$$\begin{aligned} M_m \ddot{q}_m &= -f_m \dot{q}_m - r_g k (r_g q_m - q_a) \\ &\quad - r_g d (r_g \dot{q}_m - \dot{q}_a) + \tau, \end{aligned} \quad (7)$$

where M_m is the motor inertia matrix, q_m the motor angle, q_a the arm angle, $r_g < 1$ the gear ratio, f_m the motor friction, k the spring constant and d the damping coefficient. Input to the system is the motor torque, τ . The corresponding relation for the arm becomes a nonlinear equation

$$\begin{aligned} M_a(q_a) \ddot{q}_a &+ C(q_a, \dot{q}_a) \dot{q}_a + g(q_a) = \\ &k (r_g q_m - q_a) + d (r_g \dot{q}_m - \dot{q}_a), \end{aligned} \quad (8)$$

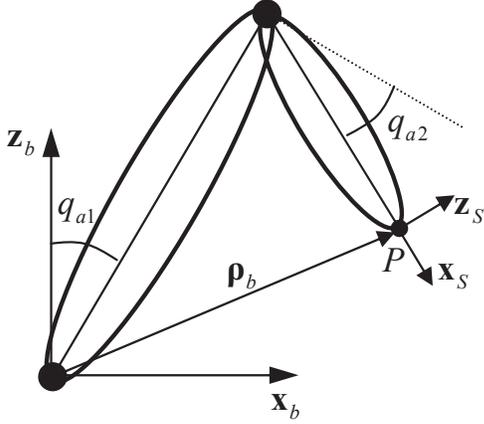


Fig. 2. A two degrees of freedom robot model. The links are assumed to be rigid and the joints are described by a two mass system connected by a spring damping pair.

where $M_a(\cdot)$ is the arm inertia matrix, $C(\cdot)$ the Coriolis and centrifugal terms and $g(\cdot)$ the gravitational torque. A more detailed model of the robot should include nonlinear friction such as Coulomb friction. An important extension would also be to model the nonlinear spring characteristics in the gear-boxes. In general the gear-box is less stiff for torques close to zero and more stiff when high torques are applied. An extended flexible joint model is proposed in [19, Paper A] which improves the control accuracy. The extended flexible joint model can also be used here.

B. Estimation Model

The estimation model has to reflect the dynamics of the true system. A straight forward choice of estimation model is the state space equivalent of (7) and (8), which gives a nonlinear dynamic model with 8 states (motor- and arm angular positions and velocities). Instead, the dynamics is reformulated and used in the measurement relation (see Section III-C) and we propose a linear state space model for the dynamics with arm angles, arm velocities and arm accelerations as state variables, together with bias terms compensating for model errors and sensor drift. The state vector is therefore given as

$$x_t = (q_{a,t}^T \quad \dot{q}_{a,t}^T \quad \ddot{q}_{a,t}^T \quad b_{m,t}^T \quad b_{\rho,t}^T)^T, \quad (9)$$

where $q_{a,t} = (q_{a,t}^1 \quad q_{a,t}^2)^T$ contains the arm angles from joint 2 and 3 in Fig 1, $\dot{q}_{a,t}$ is the angular velocity, $\ddot{q}_{a,t}$ is the angular acceleration, $b_{m,t} = (b_{m,t}^1 \quad b_{m,t}^2)^T$ is the bias term for the motor angles and $b_{\rho,t} = (b_{\rho,t}^1 \quad b_{\rho,t}^2)^T$ is the bias term for the acceleration at time t . The bias terms are used to handle model errors in the measurement equation but also to handle drift in the measured signals, especially in the acceleration signals. This yields the following state space model in discrete time

$$x_{t+1} = F_t x_t + G_{u,t} u_t + G_{w,t} w_t, \quad (10a)$$

$$y_t = h(x_t) + e_t, \quad (10b)$$

where

$$F_t = \begin{pmatrix} \mathcal{I} & T\mathcal{I} & T^2/2\mathcal{I} & \mathcal{O} & \mathcal{O} \\ \mathcal{O} & \mathcal{I} & T\mathcal{I} & \mathcal{O} & \mathcal{O} \\ \mathcal{O} & \mathcal{O} & \mathcal{I} & \mathcal{O} & \mathcal{O} \\ -\mathcal{O} & -\mathcal{O} & -\mathcal{O} & -\mathcal{I} & -\mathcal{O} \\ \mathcal{O} & \mathcal{O} & \mathcal{O} & \mathcal{O} & \mathcal{I} \end{pmatrix}, \quad (11a)$$

$$G_{w,t} = \begin{pmatrix} \frac{T^2}{2}\mathcal{I} & \mathcal{O} & \mathcal{O} \\ T\mathcal{I} & \mathcal{O} & \mathcal{O} \\ \mathcal{I} & \mathcal{O} & \mathcal{O} \\ -\mathcal{O} & -\mathcal{I} & -\mathcal{O} \\ \mathcal{O} & \mathcal{O} & \mathcal{I} \end{pmatrix}, \quad G_{u,t} = \begin{pmatrix} \frac{T^3}{6}\mathcal{I} \\ \frac{T^2}{2}\mathcal{I} \\ T\mathcal{I} \\ \mathcal{O} \\ \mathcal{O} \end{pmatrix}. \quad (11b)$$

The input signal, u_t , is ideally a known signal entering the model as described in (10). For the particular system it is not available directly. Instead it is assumed to represent the arm jerk reference, which is here given as the differentiated arm angular acceleration reference signal. The process noise, w_t and measurement noise e_t are considered Gaussian with zero mean and covariances, Q_t and R_t respectively. The sample time is denoted T and \mathcal{I} and \mathcal{O} are two by two unity and null matrices. The observation relation, (10b), is described in full detail in the next section.

C. Sensor Model

The observation relation is given by

$$h(x_t) = \begin{pmatrix} q_{m,t} + b_{m,t} \\ \ddot{\rho}_t + b_{\rho,t} \end{pmatrix}, \quad (12)$$

where $q_{m,t}$ is the motor angle and $\ddot{\rho}_t$ is the Cartesian acceleration vector in the accelerometer frame $\{s\}$, see Fig 2. With the simplified model described in Section III-A, the motor angle $q_{m,t}$ is computed from (8) according to

$$q_{m,t} = r_g^{-1} \left(q_{a,t} + k^{-1} (M_a(q_{a,t}) \ddot{q}_{a,t} + g(q_{a,t}) + C(q_{a,t}, \dot{q}_{a,t}) \dot{q}_{a,t} - d(r_g \dot{q}_{m,t} - \dot{q}_{a,t})) \right). \quad (13)$$

Here, the motor angular velocity \dot{q}_m can be seen as an input signal to the sensor model. However, the damping term $d(r_g \dot{q}_m - \dot{q}_a)$ is small compared to the other terms, and is therefore neglected.

The approach is similar to the one suggested in [8], which uses the relation given by (8) in a case when the system is scalar and linear. However, the results presented here are more general, since a multi-variable nonlinear system is considered.

The acceleration in frame $\{s\}$, Fig 2, measured by the accelerometer, is expressed as

$$\ddot{\rho}_t = R_s^b(q_{a,t}) (\ddot{\rho}_{b,t} + n_g), \quad (14)$$

where $R_s^b(q_{a,t})$ is the rotation matrix from $\{b\}$ to $\{s\}$, $n_g = (0 \quad 0 \quad g)^T$ is the gravity vector and $\ddot{\rho}_{b,t}$ is the second time derivative of the vector $\rho_{b,t}$, see Fig 2. The vector $\rho_{b,t}$ is described by the kinematics [23] which is a nonlinear mapping from joint angles to Cartesian coordinates, i.e.,

$$\rho_{b,t} = \begin{pmatrix} x_t^{\text{ACC}} \\ z_t^{\text{ACC}} \end{pmatrix} = \mathcal{T}_{\text{ACC}}(q_{a,t}), \quad (15)$$

where $\mathbf{x}_t^{\text{ACC}}$ and $\mathbf{z}_t^{\text{ACC}}$ are the position of the accelerometer expressed in frame $\{b\}$. Differentiation of $\rho_{b,t}$ twice, with respect to time, gives

$$\ddot{\rho}_{b,t} = J(q_{a,t})\ddot{q}_{a,t} + \left(\sum_{i=1}^2 \frac{\partial J(q_{a,t})}{\partial q_{a,t}^{(i)}} q_{a,t}^{(i)} \right) \dot{q}_{a,t}, \quad (16)$$

where $q_{a,t}^{(i)}$ is the i th element of $q_{a,t}$ and $J(q_{a,t})$ is the Jacobian of $\mathcal{T}_{\text{ACC}}(q_{a,t})$, i.e.,

$$J(q_a) = \nabla_{q_a} \mathcal{T}_{\text{ACC}}(q_a). \quad (17)$$

Remark: If the nonlinear dynamics (7) and (8), are used, see Section III-A, the relation in (13) becomes linear since $q_{m,t}$ is a state variable. However, the relation in (16) becomes more complex since $\ddot{q}_{a,t}$ is no longer a state, but has to be computed using (8).

IV. EXPERIMENTS ON AN ABB IRB4600 ROBOT

The experiments were performed on an ABB IRB4600, like the one seen in Fig 1. The accelerometer used in the experiments is a 3-axis accelerometer from Crossbow Technology, with a range of ± 2 g, and a sensitivity of 1 V/g [4]. In the next sections the experimental setup and results are given.

A. Experimental Setup

The orientation and position of the accelerometer were estimated using the method described in [3]. All measured signals, i.e., acceleration, motor angles and arm angular acceleration reference, are synchronous and sampled with a rate of 2 kHz. The accelerometer measurements are filtered with a LP-filter before any estimation method is applied to better reflect the tool movement. The path used in the evaluation is illustrated in Fig 3 and it is programmed such that only joint 2 and 3 are moved. Moreover, the wrist is configured such that the couplings to joint 2 and 3 are minimized. It is not possible to get measurements of the true state variables $\mathbf{x}_t^{\text{TRUE}}$, as is the case for the simulation, instead, the true trajectory of the end-effector, more precisely the *tool center point* (TCP), $\mathbf{x}_t^{\text{TCP}}$ and $\mathbf{z}_t^{\text{TCP}}$, is used for evaluation. The true trajectory is measured using a laser tracking system from Leica Geosystems. The tracking system has an accuracy of 0.01 mm per meter and a sample rate of 1 kHz [18]. However, the measured tool position is not synchronized with the other measured signals, i.e., a manual synchronization is therefore needed, which can introduce small errors. Another source of error is the accuracy of the programmed TCP in the control system of the robot. The estimated data is therefore aligned with the measured position to avoid any static errors. The alignment is performed using a least square fit between the estimated position and the measured position.

B. Experimental Results

The only measured quantity, to compare the estimates with, is the measured Cartesian tool position, as was mentioned in Section IV-A. Therefore, the estimated arm angles

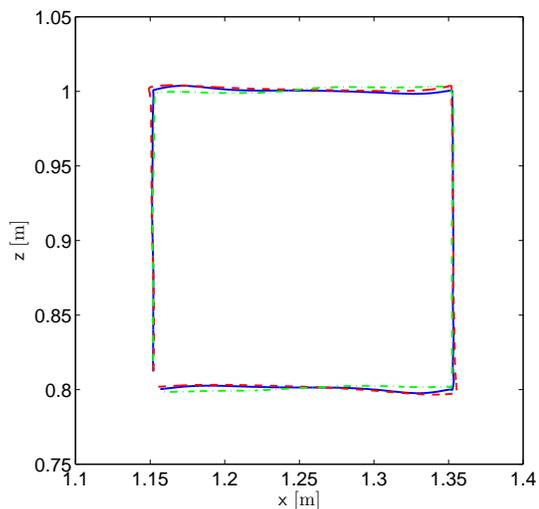


Fig. 3. The path start at the lower left corner and is counterclockwise. A laser tracking system from Leica Geosystems has been used to measure the true tool position (solid) and tool position estimates using the EKF (dashed) and $\mathcal{T}_{\text{TCP}}(q_{m,t})$ (dash-dot).

are used to compute an estimate of the TCP using the kinematic relation, i.e.,

$$\begin{pmatrix} \hat{\mathbf{x}}_t^{\text{TCP}} \\ \hat{\mathbf{z}}_t^{\text{TCP}} \end{pmatrix} = \mathcal{T}_{\text{TCP}}(\hat{q}_{a,t}), \quad (18)$$

where $\hat{q}_{a,t}$ is the result from the EKF or the PF. Another simple way to estimate the tool position is to use the forward kinematic applied to the motor angles¹, i.e., $\mathcal{T}_{\text{TCP}}(q_{m,t})$. In the evaluation study the estimates from the EKF, PF, and $\mathcal{T}_{\text{TCP}}(q_{m,t})$ are compared to measurements from the Leica system.

EKF. Fig 3 shows that the estimated paths follow the true path for the complete trajectory. The performance of the estimates are better showed in Fig 4 and 5, where the four sides are magnified. Notice that $\mathcal{T}_{\text{TCP}}(q_{m,t})$ cannot estimate the oscillations of the true path, which is not a surprise since the oscillations originates from the flexibilities in the gear boxes which are not taken care of in this straightforward way to estimate the TCP. The EKF is much better on capturing the dynamics and reducing the path error. Note however that the EKF estimate goes past the corners somewhat before it changes direction. The position *root mean square error* (RMSE) is presented in Fig 6, where the EKF acceleration method shows a significantly better performance. This is based on the single experimental trajectory, but the result is in accordance with previous simulation result as well as theoretical calculations in [15], [16]. The MATLAB implementation of the EKF is almost real-time, and without losing performance the measurements can be slightly decimated (to approximately 200 Hz), yielding faster than real-time calculations.

¹The motor angles are first transformed to the arm side of the gear box via the gear ratio.

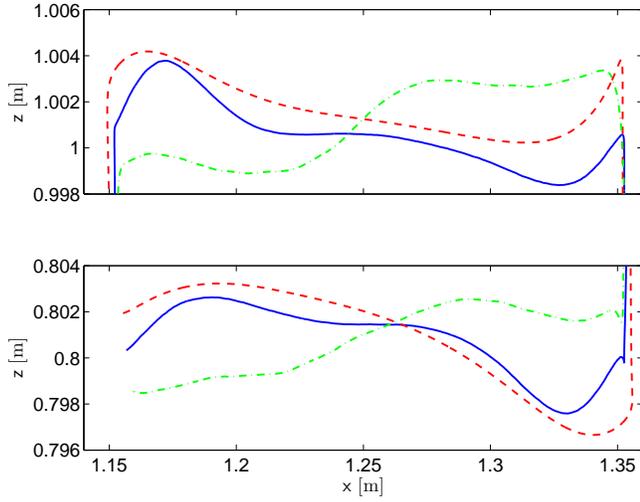


Fig. 4. Horizontal sides for the true tool position (solid) and tool position estimates using the EKF (dashed) and $\mathcal{T}_{TCP}(q_{m,t})$ (dash-dot).

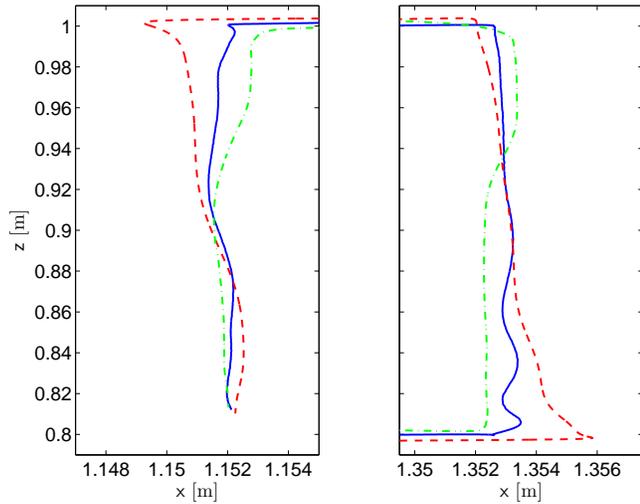


Fig. 5. Vertical sides for the true tool position (solid) and tool position estimates using the EKF (dashed) and $\mathcal{T}_{TCP}(q_{m,t})$ (dash-dot).

PF. The standard prior sampling proposal, $p^{\text{prop}}(\cdot) = p(x_{t+1}|x_t)$, that works well in simulation with no model errors did not work sufficiently well on the experimental data. Hence, it was necessary to use a better proposal. One could in principle use an optimal proposal density, [6], [10], but the problem is that it is difficult to sample from that. Instead the proposal density is approximated using an EKF, [6], [10]

$$p^{\text{prop}}(x_t|x_{t-1}, y_t) = \mathcal{N}(f(x_{t-1}^{(i)}) + K_t^{(i)}(y_t - \hat{y}_t^{(i)}), (H_t^{(i)}R_t^\dagger H_t^{(i)} + Q_{t-1}^\dagger)^\dagger), \quad (19)$$

where \dagger denotes the pseudo-inverse, and where the matrices are assumed to be evaluated for each particle state.

The result of the PF compared to the EKF are found in Fig 7 and Fig 8. The PF performs better in the corners, i.e., the estimated path does not go past the corners before it

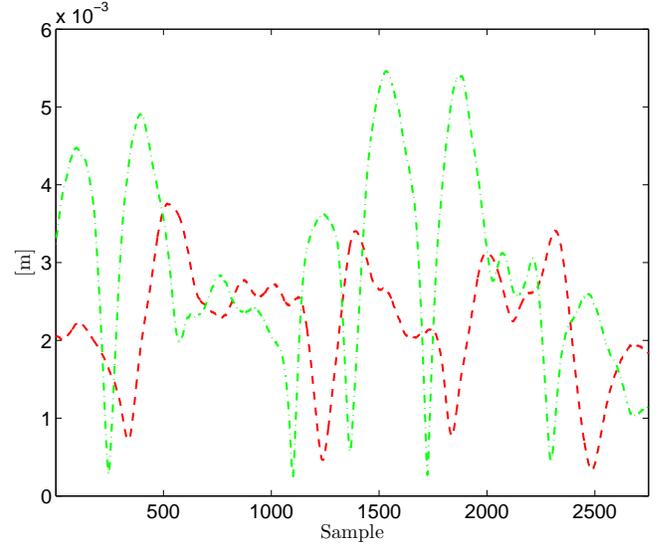


Fig. 6. Tool position RMSE for the EKF (dashed) and $\mathcal{T}_{TCP}(q_{m,t})$ (dash-dot). The 2-norm of the RMSE-signals are 0.1239 and 0.1666 for the EKF and $\mathcal{T}_{TCP}(q_{m,t})$, respectively. The RMSE is based on the single experimental trajectory, but the result is in accordance with previous simulation result as well as theoretical calculations in [15], [16].

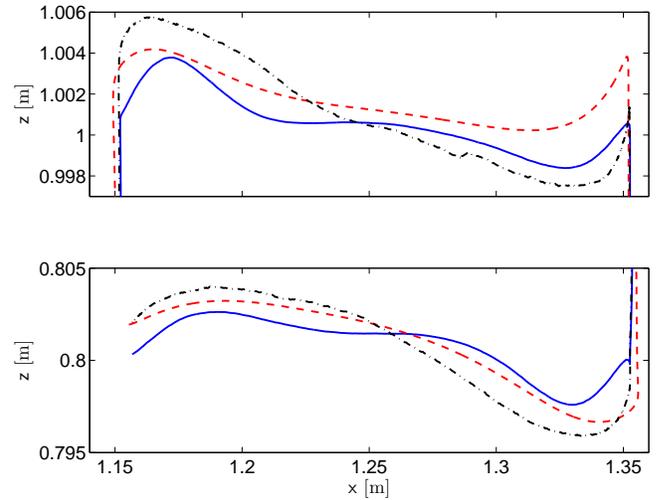


Fig. 7. Horizontal sides for the true tool position (solid) and tool position estimates using the EKF (dashed) and the PF (dash-dot).

changes. The PF estimate is also closer to the true path, at least at the vertical sides. Fig 9 shows the RMSE for the PF which is below the RMSE for the EKF most of the time. The resulting 2-norm of the RMSE for the PF is 0.0827, which is approximately 2/3 of the EKF and 1/2 of $\mathcal{T}_{TCP}(q_{m,t})$. The PF is in its current implementation far from real-time and the bias states are needed to control the model errors. Note however, if the method is intended for ILC, real-time implementations might not be necessary.

V. CONCLUSIONS AND FUTURE WORK

A sensor fusion approach to estimate the tool position by combining a 3-axes accelerometer at the end-effector and

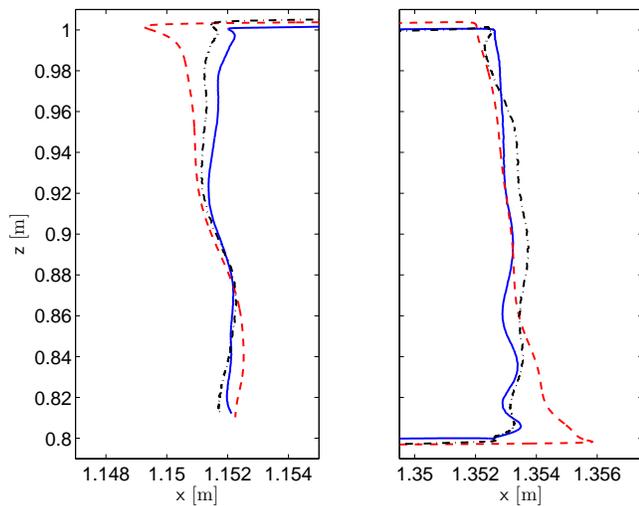


Fig. 8. Vertical sides for the true tool position (solid) and tool position estimates using the EKF (dashed) and the PF (dash-dot).

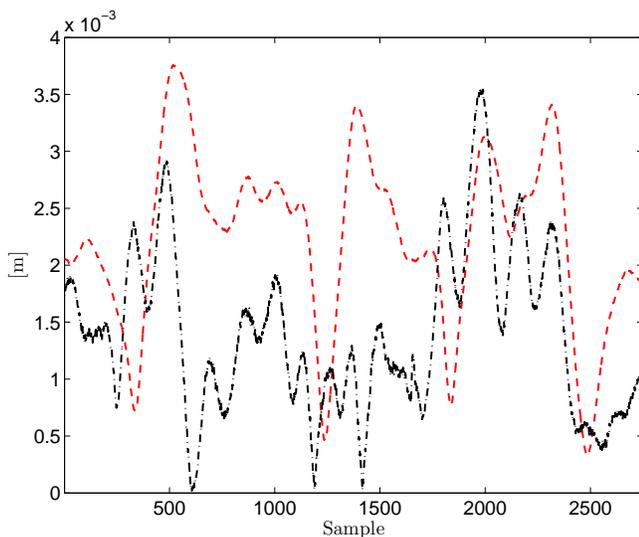


Fig. 9. Tool position RMSE for the EKF (dashed) and the PF (dash-dot). The 2-norm of the RMSE-signals are 0.1239 and 0.0827 for the EKF and the PF, respectively.

measurements from the motor angles of an industrial robot is presented. The estimation is formulated as a Bayesian problem and two solutions are proposed; extended Kalman filter and particle filter respectively, both evaluated on experimental data from an ABB robot. The dynamic performance is significantly better using the proposed accelerometer method.

Since the intended use of the estimates is to improve tool position control using an off-line method, like ILC, there are no real-time issues using the computationally demanding particle filter algorithm, however the extended Kalman filter runs in real-time in MATLAB. The estimation methods presented in this paper are general and can be extended to higher degrees of freedom robots and additional sensors, such as gyros and camera systems.

ACKNOWLEDGMENT

This work was supported by Vinnova Excellence Center LINK-SIC at Linköping University, Sweden, particularly the partner ABB, and the SSF project Collaborative Localization.

REFERENCES

- [1] B.D.O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice Hall, Englewood Cliffs, NJ, 1979.
- [2] S. Arimoto, S. Kawamura, and F. Miyazaki. Bettering operation of robots by learning. *Journal of Robotic Systems*, 1(2):123–140, 1984.
- [3] P. Axelsson and M. Norrlöf. Method to estimate the position and orientation of a triaxial accelerometer mounted to an industrial manipulator. Technical Report LiTH-ISY-R-3025, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, September 2011.
- [4] Crossbow Technology. Accelerometers, High Sensitivity, LF Series, CXL02LF3, January 2004. <http://www.xbow.com>.
- [5] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.
- [6] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.
- [7] N. J. Gordon, D. J. Salmond, and A.F.M. Smith. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.
- [8] S. Gunnarsson and M. Norrlöf. Iterative learning control of a flexible robot arm using accelerometers. In *IEEE Conference on Control Applications*, pages 1012–1016, Taipei, Taiwan, September 2004.
- [9] S. Gunnarsson, M. Norrlöf, G. Hovland, U. Carlsson, T. Brogårdh, T. Svensson, and S. Moberg. Pathcorrection for an industrial robot. European Patent Application No. EP1274546, April 2001.
- [10] F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, Lund, Sweden, 2010.
- [11] R. Jassemi-Zargani and D. Neculescu. 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.
- [12] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, 1970.
- [13] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation*. Information and System Sciences. Prentice Hall, 2000.
- [14] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the AMSE—Journal of Basic Engineering*, 82:35–45, 1960.
- [15] R. Karlsson and M. Norrlöf. Bayesian position estimation of an industrial robot using multiple sensors. In *Proceedings of the IEEE Conference on Control Applications*, pages 303–308, Taipei, Taiwan, September 2004.
- [16] 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.
- [17] K. Kozłowski. *Modelling and Identification in Robotics*. Springer-Verlag, 1998.
- [18] Leica Geosystems, Metrology Products. Case study ABB robotics - Västerås, 2008. <http://www.leica-geosystems.com>.
- [19] S. Moberg. *Modeling and Control of Flexible Manipulators*. Linköping Studies in Science and Technology. Dissertations. No. 1349, Linköping University, Linköping, Sweden, December 2010.
- [20] K. L. Moore. *Iterative Learning Control for Deterministic Systems*. Advances in Industrial Control. Springer-Verlag, London, 1993.
- [21] G. Rigatos. Particle filtering for state estimation in nonlinear industrial systems. *IEEE Transactions on Instrumentation and Measurement*, 58(11):3885–3900, November 2009.
- [22] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.
- [23] L. Sciacivico and B. Siciliano. *Modelling and Control of Robot Manipulators*. Springer, 2000.