Bayesian State Estimation of a Flexible Industrial Robot

Patrik Axelsson, Rickard Karlsson and Mikael Norrlöf

Original Publication:

Patrik Axelsson, Rickard Karlsson and Mikael Norrlöf, Bayesian State Estimation of a Flexible Industrial Robot, 2012, Control Engineering Practice, (20), 11, 1220-1228.
http://dx.doi.org/10.1016/j.conengprac.2012.06.004

Bayesian State Estimation of a Flexible Industrial Robot

Patrik Axelsson[a,*], Rickard Karlsson[b,a], Mikael Norrlöf[a]

[a]*Division of Automatic Control, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden*
[b]*Nira Dynamics, Teknikringen 6, SE-583 30 Linköping, Sweden*

Abstract

A sensor fusion method for state estimation of a flexible industrial robot is developed. By measuring the acceleration at the end-effector, the accuracy of the arm angular position, as well as the estimated position of the end-effector are improved. The problem is formulated in a Bayesian estimation framework and two solutions are proposed; the extended Kalman filter and the particle filter. In a simulation study on a realistic flexible industrial robot, the angular position performance is shown to be close to the fundamental Cramér-Rao lower bound. The technique is also verified in experiments on an ABB robot, where the dynamic performance of the position for the end-effector is significantly improved.

Keywords: Industrial robot, positioning, estimation, particle filter, extended Kalman filter, Cramér-Rao lower bound.

1. Introduction

Modern industrial robot control is usually based only on measurements from the motor angles of the manipulator. However, the ultimate goal is to move the tool according to a predefined path. In Gunnarsson et al. (2001) a method for improving the absolute accuracy of a standard industrial manipulator is described, where improved accuracy is achieved through identification of unknown or uncertain parameters in the robot system, and applying the *iterative learning control* (ILC) method, (Arimoto et al., 1984; Moore, 1993), 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 demanding applications such as laser cutting, where low cost sensors such as accelerometers are a feasible choice.

Two Bayesian state estimation techniques, the *extended Kalman filter* (EKF) and the *particle filter* (PF), are applied to a standard industrial manipulator and the result is evaluated with respect to the tracking performance in terms of position accuracy of the tool. The main contribution in this paper compared to previous papers in the field is the combination of: i) the evaluation of estimation results in relation to the *Cramér-Rao lower bound* (CRLB); ii) the utilization of motor angle measurement and accelerometer measurement in the filters; iii) the experimental evaluation on a commercial industrial robot, see Figure 1; iv) the extensive comparison of EKF and PF, and finally; v) the use of a manipulator model including a complete model of the manipulator's flexible modes. In addition, the utilization of the calibration of the accelerometer sensor from Axelsson and Norrlöf (2012) and the proposal density for the PF using an EKF, (Doucet et al., 2000; Gustafsson, 2010), is non standard.

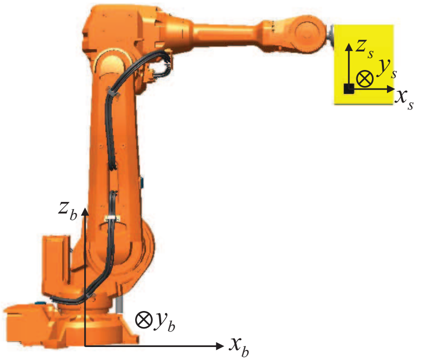

Figure 1: The ABB IRB4600 robot with the accelerometer. The base coordinate system, (x_b, y_b, z_b), and the coordinate system for the sensor (accelerometer), (x_s, y_s, z_s), are also shown.

Traditionally, many nonlinear Bayesian estimation problems are solved using the EKF (Anderson and Moore, 1979; Kailath et al., 2000). When the dynamic models and measurements are highly nonlinear and the measurement noise is not Gaussian, linearized methods may not always be a good approach. The PF (Gordon et al., 1993; Doucet et al., 2001) provides a general solution to many problems where linearizations and Gaussian approximations are intractable or would yield too low performance.

Bayesian techniques have traditionally been applied in

[*] Corresponding author. *E-mail address*: axelsson@isy.liu.se, *Phone*: +46 13 284 474, *Fax*: +46 13 139 282

Email addresses: axelsson@isy.liu.se (Patrik Axelsson), rickard.karlsson@niradynamics.se (Rickard Karlsson), mino@isy.liu.se (Mikael Norrlöf)

Preprint submitted to Control Engineering Practice *June 7, 2012*

mobile robot applications, see e.g. Kwok et al. (2004); Jensfelt (2001), and Doucet et al. (2001, Ch. 19). In the industrial robotics research area one example is Jassemi-Zargani and Necsulescu (2002) where an EKF is used to improve the trajectory tracking for a rigid 2 *degree-of-freedom* (DOF) robot using arm angle measurements and tool acceleration measurements. The extension to several DOF is presented in Quigley et al. (2010), where the EKF is used on a robot manipulator with 7 DOF and 3 accelerometers. A method for accelerometer calibration with respect to orientation is also presented. The idea of combining a vision sensor, accelerometers, and gyros when estimating the tool position is explored in Jeon et al. (2009) for a 2 DOF manipulator, using a kinematic Kalman filter. Another way is to use the acceleration of the tool as an input instead of a measurement as described in De Luca et al. (2007), where it is assumed that the friction is neglected, the damping and spring are assumed linear. As a result, the estimation can be done using a linear time invariant observer with dynamics based upon pole placement. For flexible link robots the Kalman filter has been investigated in Li and Chen (2001) for a single link, where the joint angle and the acceleration of the tool are used as measurements. Moreover, in Lertpiriyasuwat et al. (2000) the extended Kalman filter has been used for a two link manipulator using the joint angles and the tool position as measurements. In both cases, the manipulator is operating in a plane perpendicular to the gravity field. Sensor fusion techniques using particle filters have so far been applied to very few industrial robotic applications (Rigatos, 2009; Karlsson and Norrlöf, 2004, 2005), and only using simulated data. The PF method is also motivated since it provides the possibility to design control laws and perform diagnosis in a much more advanced way, making use of the full posterior *probability density function* (PDF). The PF also enables incorporation of hard constraints on the system parameters, and it provides a benchmark for simpler solutions, such as given by the EKF.

This paper extends the simulation studies introduced in Karlsson and Norrlöf (2004, 2005) with experimental results. A performance evaluation in a realistic simulation environment for both the EKF and the PF is presented and it is analyzed using the *Cramér-Rao lower bound* (CRLB), (Bergman, 1999; Kay, 1993). In addition to Karlsson and Norrlöf (2004, 2005), experimental data, from a state of the art industrial robot, are used for evaluation of the proposed methods. A detailed description of the experimental setup is given and also modifications of the PF for experimental data are presented.

The paper is organized as follows. In Section 2, the Bayesian theory estimation is introduced and the concept of the CRLB is presented. The robot, estimation, and sensor models, are presented in Section 3. The performance of the EKF and PF are compared to the Cramér-Rao lower bound limit for simulated data in Section 4. In Section 5 the experimental setup and performance are presented. Finally, Section 6 contains conclusive remarks and future work.

2. Bayesian Estimation

Consider the discrete state-space model

$$\begin{aligned} \mathbf{x}_{t+1} &= f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t), && \text{(1a)} \\ \mathbf{y}_t &= h(\mathbf{x}_t) + \mathbf{e}_t, && \text{(1b)} \end{aligned}$$

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 nonlinear 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 (Jazwinski, 1970) are given by

$$\begin{aligned} p(\mathbf{x}_{t+1}|\mathbb{Y}_t) &= \int_{\mathbb{R}^n} p(\mathbf{x}_{t+1}|\mathbf{x}_t) p(\mathbf{x}_t|\mathbb{Y}_t) d\mathbf{x}_t, && \text{(2a)} \\ p(\mathbf{x}_t|\mathbb{Y}_t) &= \frac{p(\mathbf{y}_t|x_t) p(\mathbf{x}_t|\mathbb{Y}_{t-1})}{p(\mathbf{y}_t|\mathbb{Y}_{t-1})}. && \text{(2b)} \end{aligned}$$

For the important special case of linear-Gaussian dynamics and linear-Gaussian observations, the Kalman filter (Kalman, 1960) provides the solution. For nonlinear and non-Gaussian systems, the PDF can not, 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, Sorenson (1988); Arulampalam et al. (2002). Here, two different approaches of solving the Bayesian equations are considered; *extended Kalman filter* (EKF) , and *particle filter* (PF). The EKF will solve the problem using a linearization of the system and assuming Gaussian noise. 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 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.

2.1. The Extended Kalman Filter (EKF)

For the special case of linear dynamics, linear measurements and additive Gaussian noise, the Bayesian recursions in (2) have an analytical solution given by the Kalman filter. 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 (Anderson and Moore, 1979; Kailath et al., 2000) where the model is linearized around the previous estimate. The time update and measurement update for the EKF are

$$\begin{cases} \hat{\mathbf{x}}_{t+1|t} = f(\hat{\mathbf{x}}_{t|t}, \mathbf{u}_t, 0), \\ \mathbf{P}_{t+1|t} = \mathbf{F}_t \mathbf{P}_{t|t} \mathbf{F}_t^T + \mathbf{G}_t \mathbf{Q}_t \mathbf{G}_t^T, \end{cases} \quad \text{(3a)}$$

$$\begin{cases} \hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - h(\hat{\mathbf{x}}_{t|t-1})), \\ \mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{H}_t \mathbf{P}_{t|t-1}, \\ \mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T + \mathbf{R}_t)^{-1}, \end{cases} \quad \text{(3b)}$$

where the linearized matrices are given as

$$\mathbf{F}_t = \nabla_{\mathbf{x}} f(\mathbf{x}_t, \mathbf{u}_t, 0)|_{\mathbf{x}_t=\hat{\mathbf{x}}_{t|t}}, \tag{4a}$$

$$\mathbf{G}_t = \nabla_{\mathbf{w}} f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t)|_{\mathbf{x}_t=\hat{\mathbf{x}}_{t|t}}, \tag{4b}$$

$$\mathbf{H}_t = \nabla_{\mathbf{x}} h(\mathbf{x}_t)|_{\mathbf{x}_t=\hat{\mathbf{x}}_{t|t-1}}. \tag{4c}$$

In (3), $\mathbf{P}_{t+1|t}$ and $\mathbf{P}_{t|t}$ denote the covariance matrices for the estimation errors at time $t+1$ and t given measurements up to time t, and the noise covariances are given as

$$\mathbf{Q}_t = \mathrm{Cov}\,(\mathbf{w}_t)\,,\ \mathbf{R}_t = \mathrm{Cov}\,(\mathbf{e}_t)\,, \tag{5}$$

where the process noise and measurement noise are assumed zero mean processes.

2.2. The Particle Filter (PF)

The PF from Doucet et al. (2001); Gordon et al. (1993); Ristic et al. (2004) 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 $\mathbf{x}_t$ denote the state of the observed system and $\mathbb{Y}_t = \{\mathbf{y}_i\}_{i=1}^t$ be the set of observed measurements until present time. 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, $\gamma_t^{(i)}$, 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 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, (Gordon et al., 1993). The PF is summarized in Algorithm 1, where the proposal distribution $p^{\text{prop}}(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(i)}, \mathbf{y}_{t+1})$ 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.,

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

but other choices, such as the ML-estimate, might be of interest. There exist theoretical limits (Doucet et al., 2001) that the approximated PDF converges to the true as the number of particles tends to infinity.

2.3. Cramér-Rao Lower Bound

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 linearization or model structure used, will affect the performance. The *Cramér-Rao lower bound* (CRLB) is such a characteristic for the second order moment (Kay, 1993; Cramér, 1946). Here, only state-space models with additive Gaussian noise are considered. The theoretical posterior CRLB for a general dynamic system was derived in Van Trees (1968); Tichavsky et al. (1998); Bergman (1999); Doucet et al. (2001). Here a continuous-time system is considered. By first linearizing and then discretizing the system, the fundamental limit can in practice be calculated as the stationary solution for every t, $\bar{\mathbf{P}} = \bar{\mathbf{P}}(\mathbf{x}_t^{\text{TRUE}})$, of the Riccati recursions in the EKF, where the linearizations are around the true state trajectory, $\mathbf{x}_t^{\text{TRUE}}$. Note that the approximation is fairly accurate for the industrial robot application due to a high sample rate and a small process noise. The predicted value of the stationary covariance for each time t, i.e., for each point in the state-space, $\mathbf{x}_t^{\text{TRUE}}$, is denoted $\bar{\mathbf{P}}_p$ and given as the solution to

Algorithm 1 The Particle Filter

```
1: Generate $N$ samples $\{\mathbf{x}_0^{(i)}\}_{i=1}^N$ from $p(x_0)$.
2: Compute the weights
       $\gamma_t^{(i)} = \gamma_{t-1}^{(i)} \cdot \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{p^{\text{prop}}(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)}$
   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 $\{\mathbf{x}_t^{(i\star)}\}_{i=1}^N$ by resampling with replacement $N$ times from $\{\mathbf{x}_t^{(i)}\}_{i=1}^N$, with probability $\bar{\gamma}_t^{(i)} = Pr\{\mathbf{x}_t^{(i\star)} = \mathbf{x}_t^{(i)}\}$ and reset the weights to $1/N$; otherwise let $\{\mathbf{x}_t^{(i\star)}\}_{i=1}^N = \{\mathbf{x}_t^{(i)}\}_{i=1}^N$.
4: Generate predictions from the proposal distribution $\mathbf{x}_{t+1}^{(i)} \sim p^{\text{prop}}(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i\star)}, \mathbf{y}_{t+1})$, $i = 1, \dots, N$.
5: Increase $t$ and continue to step 2.
```

$$\bar{\mathbf{P}}_p = \bar{\mathbf{F}}(\bar{\mathbf{P}}_p - \bar{\mathbf{K}}\bar{\mathbf{H}}\bar{\mathbf{P}}_p)\bar{\mathbf{F}}^T + \bar{\mathbf{G}}\mathbf{Q}\bar{\mathbf{G}}^T. \tag{7}$$

where the linearized matrices $\bar{\mathbf{F}}$, $\bar{\mathbf{G}}$ and $\bar{\mathbf{H}}$ are evaluated around the true trajectory, $\mathbf{x}_t^{\text{TRUE}}$, and

$$\bar{\mathbf{K}} = \bar{\mathbf{P}}_p\bar{\mathbf{H}}^T(\bar{\mathbf{H}}\bar{\mathbf{P}}_p\bar{\mathbf{H}}^T + \mathbf{R})^{-1}. \tag{8}$$

The CRLB limit can now be calculated as

$$\bar{\mathbf{P}} = \bar{\mathbf{P}}_p - \bar{\mathbf{K}}\bar{\mathbf{H}}\bar{\mathbf{P}}_p, \tag{9}$$

for each point along the true state-trajectory.

3. Dynamic Models

In this section a continuous-time 2 DOF robot model is discussed. The model is simplified and transformed into discrete time, to be used by the EKF and PF. The measurements are in both cases angle measurements from the motors, with or without acceleration information from the end-effector.

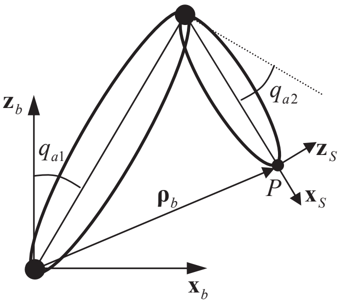

Figure 2: A 2 DOF robot model. The links are assumed to be rigid and the joints are described by a two mass system connected by a spring damping pair.

3.1. Robot Model

The robot model used in this work is a joint flexible two-axes model, see Figure 2. The model corresponds to axes 2 and 3 of a serial 6 DOF industrial robot like the one in Figure 1. 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 Kozlowski (1998). Here, it will be assumed that the transmission can be modelled as 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. A common way to obtain the dynamic model in industrial robotics is to use Lagrange's equation as described in Sciavicco and Siciliano (2000). The equation describing the torque balance for the motor becomes

$$\begin{aligned} M_m\ddot{\mathbf{q}}_m =& - f_m\dot{\mathbf{q}}_m - r_g k(r_g\mathbf{q}_m - \mathbf{q}_a) \\ & - r_g d(r_g\dot{\mathbf{q}}_m - \dot{\mathbf{q}}_a) + \boldsymbol{\tau}_m, \end{aligned} \tag{10}$$

where M_m is the motor inertia matrix, $\mathbf{q}_m = \begin{pmatrix} q_m^1 & q_m^2 \end{pmatrix}^T$ the motor angles, $\mathbf{q}_a = \begin{pmatrix} q_a^1 & q_a^2 \end{pmatrix}^T$ the arm angles, r_g the gear ratio, f_m the viscous friction at the motor, k the spring constant and d the damping coefficient. No couplings between motor 1 and 2 implies that M_m is a diagonal matrix. The parameters r_g, f_m, k, and d are two by two diagonal matrices, where the diagonal element ii corresponds to joint i. The inputs to the system are the motor torques, $\boldsymbol{\tau}_m = \begin{pmatrix} \tau_m^1 & \tau_m^2 \end{pmatrix}^T$. The corresponding relation for the arm becomes a nonlinear equation

$$\begin{aligned} M_a(\mathbf{q}_a)\ddot{\mathbf{q}}_a + & C(\mathbf{q}_a, \dot{\mathbf{q}}_a)\dot{\mathbf{q}}_a + g(\mathbf{q}_a) = \\ & k(r_g\mathbf{q}_m - \mathbf{q}_a) + d(r_g\dot{\mathbf{q}}_m - \dot{\mathbf{q}}_a), \end{aligned} \tag{11}$$

where $M_a(\cdot)$ is the arm inertia matrix, $C(\cdot)$ the Coriolis- and centrifugal terms and $g(\cdot)$ the gravitational torque. Here, it is assumed that there are no couplings between the arms and motors, which is valid if the gear ratio is high (Spong, 1987). 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. The extended flexible joint model proposed in Moberg (2010, Paper A), which improves the control accuracy, can also be used.

3.2. Estimation Model

The estimation model has to reflect the dynamics in the true system. A straight forward choice of estimation model is the state space equivalent of (10) and (11), which gives a nonlinear dynamic model with 8 states (motor and arm angular positions and velocities). This gives both a nonlinear state space model and a nonlinear measurement model. Instead, a linear state space model is suggested with arm angles, velocities and accelerations as state variables, in order to simplify the time update for the PF. Note that the measurement model is still nonlinear in this case. Bias states compensating for measurement and model errors have shown to improve the accuracy and are therefore also included. The state vector is now given as

$$\mathbf{x}_t = \begin{pmatrix} \mathbf{q}_{a,t}^T & \dot{\mathbf{q}}_{a,t}^T & \ddot{\mathbf{q}}_{a,t}^T & \mathbf{b}_{m,t}^T & \mathbf{b}_{\ddot{\rho},t}^T \end{pmatrix}^T, \tag{12}$$

where $\mathbf{q}_{a,t} = \begin{pmatrix} q_{a,t}^1 & q_{a,t}^2 \end{pmatrix}^T$ contains the arm angles from joint 2 and 3 in Figure 1, $\dot{\mathbf{q}}_{a,t}$ is the angular velocity, $\ddot{\mathbf{q}}_{a,t}$ is the angular acceleration, $\mathbf{b}_{m,t} = \begin{pmatrix} b_{m,t}^1 & b_{m,t}^2 \end{pmatrix}^T$ contains the bias terms for the motor angles, and $\mathbf{b}_{\ddot{\rho},t} = \begin{pmatrix} b_{\ddot{\rho},t}^1 & b_{\ddot{\rho},t}^2 \end{pmatrix}^T$ contains the bias terms for the acceleration at time t. The bias states are used to handle model errors in the measurement equation but also to handle drifts in the measured signals, especially in the acceleration signals. The first three states are given by a constant acceleration model discretized with zero order hold, and the bias states are modeled as random walk. This yields the following state space model in discrete time

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \mathbf{G}_u\mathbf{u}_t + \mathbf{G}_w\mathbf{w}_t, \tag{13a}$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{e}_t, \tag{13b}$$

where

$$\mathbf{F} = \left(\begin{array}{ccc:cc} \mathcal{I} & T\mathcal{I} & T^2/2\mathcal{I} & \mathcal{O} & \mathcal{O} \\ \mathcal{O} & \mathcal{I} & T\mathcal{I} & \mathcal{O} & \mathcal{O} \\ \mathcal{O} & \mathcal{O} & \mathcal{I} & \mathcal{O} & \mathcal{O} \\ \hdashline \mathcal{O} & \mathcal{O} & \mathcal{O} & \mathcal{I} & \mathcal{O} \\ \mathcal{O} & \mathcal{O} & \mathcal{O} & \mathcal{O} & \mathcal{I} \end{array}\right), \tag{14a}$$

$$\mathbf{G}_w = \left(\begin{array}{c:cc} \frac{T^3}{6}\mathcal{I} & \mathcal{O} & \mathcal{O} \\ \frac{T^2}{2}\mathcal{I} & \mathcal{O} & \mathcal{O} \\ T\mathcal{I} & \mathcal{O} & \mathcal{O} \\ \hdashline \mathcal{O} & \mathcal{I} & \mathcal{O} \\ \mathcal{O} & \mathcal{O} & \mathcal{I} \end{array}\right), \quad \mathbf{G}_u = \left(\begin{array}{c} \frac{T^3}{6}\mathcal{I} \\ \frac{T^2}{2}\mathcal{I} \\ T\mathcal{I} \\ \hdashline \mathcal{O} \\ \mathcal{O} \end{array}\right). \tag{14b}$$

The input, $\mathbf{u}_t$, is the arm jerk reference, i.e., the differentiated arm angular acceleration reference. The process noise, $\mathbf{w}_t$ and measurement noise $\mathbf{e}_t$ are considered Gaussian with zero mean and covariances, $\mathbf{Q}_t$ and $\mathbf{R}_t$ respectively. The sample time is denoted T and $\mathcal{I}$ and $\mathcal{O}$ are two by two identity and null matrices. The sensor model (13b) is described in full detail in the next section.

3.3. Sensor Model

The available measurements are motor angular positions from resolvers and the acceleration of the end-effector from the accelerometer. The sensor model is thus given by

$$h(\mathbf{x}_t) = \begin{pmatrix} \mathbf{q}_{m,t} + \mathbf{b}_{m,t} \\ \ddot{\boldsymbol{\rho}}_t + \mathbf{b}_{\ddot{\rho},t} \end{pmatrix}, \tag{15}$$

where $\mathbf{q}_{m,t} = \begin{pmatrix} q^1_{m,t} & q^2_{m,t} \end{pmatrix}^T$ is the motor angle and $\ddot{\boldsymbol{\rho}}_t = \begin{pmatrix} \rho^{\mathsf{x}}_t & \rho^{\mathsf{z}}_t \end{pmatrix}^T$ is the Cartesian acceleration vector in the accelerometer frame $O\mathbf{x}_s\mathbf{z}_s$, see Figure 2. With the simplified model described in Section 3.1, the motor angle $\mathbf{q}_{m,t}$ is computed from (11) according to

$$\begin{aligned}\mathbf{q}_{m,t} =& r_g^{-1}\Big(\mathbf{q}_{a,t} + k^{-1}\big(M_a(\mathbf{q}_{a,t})\ddot{\mathbf{q}}_{a,t} + g(\mathbf{q}_{a,t}) \\ &+ C(\mathbf{q}_{a,t}, \dot{\mathbf{q}}_{a,t})\dot{\mathbf{q}}_{a,t} - d(r_g\dot{\mathbf{q}}_{m,t} - \dot{\mathbf{q}}_{a,t})\big)\Big). \end{aligned}\tag{16}$$

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

The approach is similar to the one suggested in Gunnarsson and Norrlöf (2004), which uses the relation given by (11) in the case when the system is scalar and linear. The results presented here are more general, since a multivariable nonlinear system is considered.

The acceleration in frame $O\mathbf{x}_s\mathbf{z}_s$, in Figure 2, measured by the accelerometer, can be expressed as

$$\ddot{\boldsymbol{\rho}}_t = \mathbf{R}^b_s(\mathbf{q}_{a,t})\left(\ddot{\boldsymbol{\rho}}_{b,t} + \mathbf{n}_g\right), \tag{17}$$

where $\mathbf{R}^b_s(\mathbf{q}_{a,t})$ is the rotation matrix from $O\mathbf{x}_b\mathbf{z}_b$ to $O\mathbf{x}_s\mathbf{z}_s$, $\mathbf{n}_g = \begin{pmatrix} 0 & g \end{pmatrix}^T$ is the gravity vector and $\ddot{\boldsymbol{\rho}}_{b,t}$ is the second time derivative of the vector $\boldsymbol{\rho}_{b,t}$, see Figure 2. The vector $\boldsymbol{\rho}_{b,t}$ is described by the forward kinematics (Sciavicco and Siciliano, 2000) which is a nonlinear mapping from joint angles to Cartesian coordinates, i.e.,

$$\boldsymbol{\rho}_{b,t} = \begin{pmatrix} \mathsf{x}^{\text{ACC}}_t \\ \mathsf{z}^{\text{ACC}}_t \end{pmatrix} = \mathcal{T}_{\text{ACC}}(\mathbf{q}_{a,t}), \tag{18}$$

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

$$\ddot{\boldsymbol{\rho}}_{b,t} = \mathbf{J}(\mathbf{q}_{a,t})\ddot{\mathbf{q}}_{a,t} + \left(\sum_{i=1}^{2} \frac{\partial \mathbf{J}(\mathbf{q}_{a,t})}{\partial q^{(i)}_{a,t}} \dot{q}^{(i)}_{a,t}\right)\dot{\mathbf{q}}_{a,t}, \tag{19}$$

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

$$\mathbf{J}(\mathbf{q}_a) = \nabla_{\mathbf{q}_a}\mathcal{T}_{\text{ACC}}(\mathbf{q}_a). \tag{20}$$

Both the position model (16) for the motors and the acceleration model (19) are now a function of the state variables $\mathbf{q}_{a,t}$, $\dot{\mathbf{q}}_{a,t}$, and $\ddot{\mathbf{q}}_{a,t}$.

Remark: If the nonlinear dynamics (10) and (11), are used, see Section 3.1, the relation in (16) becomes linear since $\mathbf{q}_{m,t}$ is a state variable. However, the relation in (19) becomes more complex since $\ddot{\mathbf{q}}_{a,t}$ is no longer a state, but has to be computed using (11).

4. Analysis

4.1. Simulation Model

In order to perform *Cramér-Rao lower bound* (CRLB) analysis, the true robot trajectory must be known. Hence, in practice this must be conducted in a simulation environment since not all state variables are available as measurements. In the sequel, the simulation model described in (Karlsson and Norrlöf, 2005) is used, where the CRLB analysis is compared to Monte Carlo simulations of the EKF and PF.

4.2. Cramér-Rao lower bound Analysis of the Robot

In Section 2.3, the posterior *Cramér-Rao lower bound* (CRLB) was defined for a general nonlinear system with additive Gaussian noise. In this section the focus is on the CRLB expression for the industrial robot presented in Section 3.1. Solving for the acceleration in (11) yields

$$\begin{aligned}\kappa(\mathbf{q}_a, \dot{\mathbf{q}}_a) \triangleq \ddot{\mathbf{q}}_a = -M_a(\mathbf{q}_a)^{-1}\big(k(r_g\mathbf{q}_m - \mathbf{q}_a) \\ - d(r_g\dot{\mathbf{q}}_m - \dot{\mathbf{q}}_a) - g(\mathbf{q}_a) - C(\mathbf{q}_a, \dot{\mathbf{q}}_a)\dot{\mathbf{q}}_a\big).\end{aligned}\tag{21}$$

Here, the motor angular velocity, $\dot{\mathbf{q}}_m$, is considered as an input signal, hence not part of the state-vector, $\mathbf{x}(t) = \begin{pmatrix} \mathbf{q}_a & \dot{\mathbf{q}}_a & \ddot{\mathbf{q}}_a \end{pmatrix}^T$. The system can be written in state space form as

$$\dot{\mathbf{x}} = \frac{d}{dt}\begin{pmatrix} \mathbf{q}_a \\ \dot{\mathbf{q}}_a \\ \ddot{\mathbf{q}}_a \end{pmatrix} = f^c(\mathbf{x}(t)) = \begin{pmatrix} \dot{\mathbf{q}}_a \\ \ddot{\mathbf{q}}_a \\ \Lambda(\mathbf{q}_a, \dot{\mathbf{q}}_a, \ddot{\mathbf{q}}_a) \end{pmatrix}, \tag{22a}$$

$$\Lambda(\mathbf{q}_a, \dot{\mathbf{q}}_a, \ddot{\mathbf{q}}_a) = \frac{d}{dt}\kappa(\mathbf{q}_a, \dot{\mathbf{q}}_q). \tag{22b}$$

The differentiation of κ is performed symbolically, using the MATLAB symbolic toolbox. According to Section 2.3 the CRLB is defined as the stationary Riccati solution of the EKF around the true trajectory, $\mathbf{x}^{\text{TRUE}}_t$. This is formulated for a discrete-time system. Hence, the continuous-time robot model from (22) must be discretized. This can be done by first linearizing the system and then discretizing it, (Gustafsson, 2010). The differentiation is done numerically around the true trajectory, to avoid the very complex symbolic gradient, and the result becomes,

$$\mathbf{A}^c = \nabla_{\mathbf{x}} f^c(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^{\text{TRUE}}_t} \tag{23}$$

$$= \begin{pmatrix} \mathcal{O} & \mathcal{I} & \mathcal{O} \\ \mathcal{O} & \mathcal{O} & \mathcal{I} \\ \frac{\partial\Lambda(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})}{\partial\mathbf{q}} & \frac{\partial\Lambda(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})}{\partial\dot{\mathbf{q}}} & \frac{\partial\Lambda(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})}{\partial\ddot{\mathbf{q}}} \end{pmatrix}. \tag{24}$$

The desired discrete time system matrix is now given as

$$\bar{\mathbf{F}} = e^{\mathbf{A}^c \cdot T}, \tag{25}$$

where T is the sample time. The CRLB is presented in Figure 3.

4.3. Estimation Performance

The performance of the EKF and PF are compared against the *Cramér-Rao lower bound* (CRLB) calculated in Section 4.2, using simulated data. The model is implemented and simulated using the Robotics Toolbox (Corke, 1996) in MATLAB Simulink. The robot is stabilized using a single PID-controller. The estimation model and sensor model will not use the bias states described in Section 3.2 because no model errors or drift are included in the simulation. This means that only the upper left corner of the matrices in (14) are used.

The simulation study is based mainly around the EKF approach, since it is a fast method well suited for large Monte Carlo simulations. The PF is much slower, therefore, a smaller Monte Carlo study is performed. The Monte Carlo simulations use the following covariance matrices for the process and measurement noise

$$\mathbf{Q} = 4 \cdot 10^{-6}\mathcal{I}, \quad \mathbf{R} = \begin{pmatrix} 10^{-6} \cdot \mathcal{I} & \mathcal{O} \\ \mathcal{O} & 10^{-4} \cdot \mathcal{I} \end{pmatrix}. \tag{26}$$

The measurement covariance is basically given by the motor angle and accelerometer uncertainty, and the process noise covariance is considered as a filter design parameter. The system is simulated around the nominal trajectory and produces different independent noise realizations for the measurement noise in each simulation. The continuous-time Simulink model of the robot is sampled in 1 kHz. The data is then decimated to 100 Hz before any estimation method is applied.

The estimation performance is evaluated using the *root mean square error* (RMSE) which is defined as

$$\text{RMSE}(t) = \left(\frac{1}{N_{\text{MC}}} \sum_{j=1}^{N_{\text{MC}}} \|\mathbf{x}_t^{\text{TRUE}} - \hat{\mathbf{x}}_t^{(j)}\|_2^2 \right)^{1/2}, \tag{27}$$

where N_{MC} is the number of Monte Carlo simulations, $\mathbf{x}_t^{\text{TRUE}}$ is the true state vector and $\hat{\mathbf{x}}_t^{(j)}$ is the estimated state vector in Monte Carlo simulation j. Here, the state vector is divided up into states corresponding to angular position, angular velocity, and angular acceleration, before (27) is used.

EKF. In Figure 3 the *root mean square error* (RMSE) from 500 Monte Carlo simulations are compared to the CRLB limit, both with and without acceleration measurements. The CRLB is computed as the square root of the trace for the covariance matrix part corresponding to the angular states. As seen, the RMSE is close the fundamental limit. The discrepancy is due to model errors, i.e., neglected damping term and the fact that the estimator uses a simplified system matrix consisting of integrators only. Also note that the accelerometer measurements reduce the estimation uncertainty. The results in Figure 3 are of course for the chosen trajectory, but the acceleration values are not that large, so greater differences will occur for larger accelerations. The RMSE, ignoring the initial transient is given in Table 1 for both angular position, velocity and acceleration. The improvements are substantial in angular position, but for control, the improvements in angular velocity and acceleration are important.

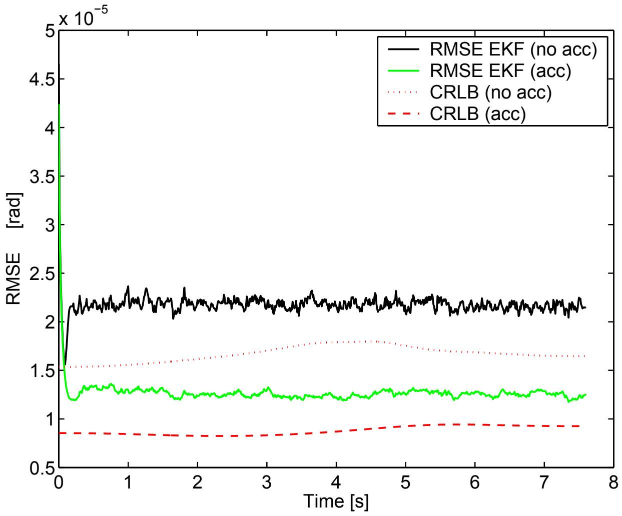

Figure 3: Angular position RMSE from 500 Monte Carlo simulations using the EKF with and without accelerometer sensor are compared to the CRLB limit for every time, i.e., the square root of the trace of the angular position from the time-varying CRLB covariance.

Table 1: The RMSE for arm-side angular position ($\mathbf{q}_a$), angular velocity ($\dot{\mathbf{q}}_a$) and angular acceleration ($\ddot{\mathbf{q}}_a$), with and without the accelerometer, using 500 Monte Carlo simulations.

	Accelerometer	**No accelerometer**
RMSE q_a	$1.25 \cdot 10^{-5}$	$2.18 \cdot 10^{-5}$
RMSE $\dot{q}_a$	$7.57 \cdot 10^{-5}$	$4.08 \cdot 10^{-4}$
RMSE $\ddot{q}_a$	$1.23 \cdot 10^{-3}$	$3.91 \cdot 10^{-3}$

PF. The proposal density $p^{\text{prop}}(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(i)}, \mathbf{y}_{t+1})$ in Algorithm 1 is chosen as the conditional prior of the state vector, i.e., $p(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(i)})$, and resampling is selected every time, which gives

$$\gamma_t^{(i)} = p(\mathbf{y}_t|\mathbf{x}_t^{(i)}), \quad i = 1, \ldots, N. \tag{28}$$

The particle filter is rather slow compared to the EKF for this model structure. Hence, the given MATLAB implementation of the system is not well suited for large Monte Carlo simulations. Instead, a small Monte Carlo study over a short part of the trajectory used for the EKF case is considered. The PF and the EKF are compared, and a small improvement in performance is noted. The result is given in Figure 4. One explanation for the similar results between the EKF and PF is that the nonlinearities may not

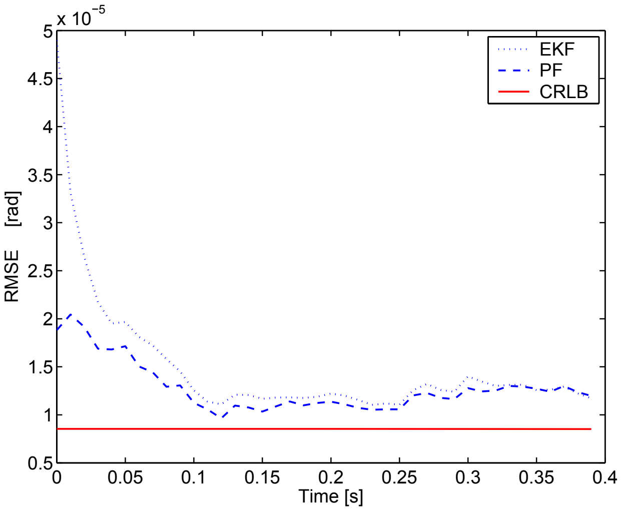

Figure 4: EKF and PF angular position RMSE with external accelerometer signal from 20 Monte Carlo simulations.

give a multi modal distribution, hence the point estimates are quite similar. The advantage with the PF is that it can utilize hard constraints on the state variables and it can also be used for control and diagnosis where the full posterior PDF is available. Even though the PF is slow, it gives more insight in the selection of simulation parameters than the EKF, where the filter performance is more dependent on the ratio between the process and measurement noise.

5. Experiments on an ABB IRB4600 Robot

The experiments were performed on an ABB IRB4600 industrial robot, like the one seen in Figure 1. To illuminate the tracking capacity of the filters, the servo tuning of the robot was not optimal, which introduces more oscillations in the path. The accelerometer used in the experiments is the triaxial accelerometer CXL02LF3 from Crossbow Technology, which has a range of ±2 g, and a sensitivity of 1 V/g (Crossbow Technology, 2004). In the next sections the experimental setup and results are given.

5.1. Experimental Setup

The orientation and position of the accelerometer were estimated using the method described in Axelsson and Norrlöf (2012). 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 low pass filter before any estimation method is applied to better reflect the tool movement. The path used in the evaluation is illustrated in Figure 5, and it is programmed such that only joints 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), $\mathrm{x}_t^{\text{TCP}}$ and $\mathrm{z}_t^{\text{TCP}}$, 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 (Leica Geosystems, Metrology Products, 2008). The measured tool position is however 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.

5.2. Experimental Results

The only measured quantity to compare the estimates with is the measured tool position, as was mentioned in Section 5.1. Therefore, the estimated arm angles are used to compute an estimate of the TCP using the kinematic relation, i.e.,

$$\begin{pmatrix} \hat{\mathrm{x}}_t^{\text{TCP}} \\ \hat{\mathrm{z}}_t^{\text{TCP}} \end{pmatrix} = \mathcal{T}_{\text{TCP}}(\hat{\mathbf{q}}_{a,t}), \tag{29}$$

where $\hat{\mathbf{q}}_{a,t}$ 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 angles [1], i.e., $\mathcal{T}_{\text{TCP}}(\mathbf{q}_{m,t})$. In the evaluation study the estimates from the EKF, PF, and $\mathcal{T}_{\text{TCP}}(\mathbf{q}_{m,t})$ are compared to measurements from the Leica system. When computing the 2-norm of the RMSE the first 0.125 seconds are disregarded in order to evaluate the tracking performance only, and not include filter transients.

In the evaluation of the experiment, the focus is on position error only since the Leica laser reference system measures position only. However, the estimation technique presented is general, so the velocity estimates will be improved as well, which is important for many control applications. In simulations this has been verified, see Section 4.3 and Table 1. Since the position is based on integrating the velocity model, this will in general be true when applied to experimental data as well. However, the current measurement system cannot be used to verify this.

EKF. Figure 5 shows that the estimated paths follow the true path. The performance of the estimates are better shown in Figure 6 and 7, where the four sides are magnified. At first, it can be noticed that $\mathcal{T}_{\text{TCP}}(\mathbf{q}_{m,t})$ cannot estimate the oscillations of the true path. This 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. However, as seen the accelerometer based sensor fusion method performs very well. It can also be noticed that the EKF estimate goes somewhat past the corners before it changes direction. An

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

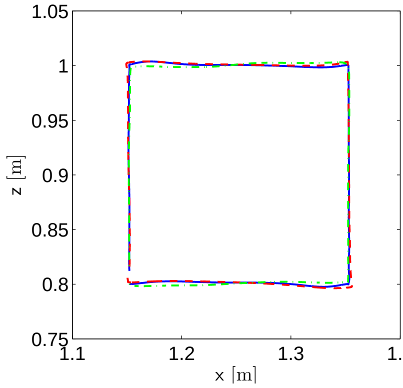

Figure 5: The path start at the lower left corner and is counter-clockwise. A laser tracking system from Leica Geosystems has been used to measure the true tool position (solid). The estimated tool position using the EKF (dashed) and $\mathcal{T}_{\text{TCP}}(\mathbf{q}_{m,t})$ (dash-dot) are also shown.

explanation to this phenomena can be that the jerk reference is used as an input to the estimation model. The jerk reference does not coincide with the actual jerk as a result of model errors and control performance. The initial transient for the EKF, due to incorrect initialization of the filter, rapidly approaches the true path. In this case $\mathcal{T}_{\text{TCP}}(\mathbf{q}_{m,t})$ starts near the true path, but $\mathcal{T}_{\text{TCP}}(\mathbf{q}_{m,t})$ can start further away for another path. The position RMSE is presented in Figure 8, where the EKF with acceleration measurements shows a significantly improve in the performance. The 2-norm of the RMSE[2] for the EKF is reduced by 25 % compared to $\mathcal{T}_{\text{TCP}}(\mathbf{q}_{m,t})$. This is based on the single experimental trajectory, but the result is in accordance with the simulation result and the theoretical calculations. Figure 8 also shows that the EKF converges fast. 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.

PF. The proposal density used during the simulation did not work properly for the experimental data due to a high *signal to noise ratio* (SNR) and also model errors. One could use an optimal proposal density, (Doucet et al., 2000; Gustafsson, 2010), but the problem is that it is difficult to sample from that. Instead, the proposal density is approximated using an EKF, (Doucet et al., 2000; Gustafsson, 2010)

$$p^{\text{prop}}(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t) = \\ \mathcal{N}(f(\mathbf{x}_{t-1}^{(i)}) + \mathbf{K}_t^{(i)}(\mathbf{y}_t - \hat{\mathbf{y}}_t^{(i)}), (\mathbf{H}_t^{(i)}\mathbf{R}_t^{\dagger}\mathbf{H}_t^{(i)} + \mathbf{Q}_{t-1}^{\dagger})^{\dagger}), \tag{30}$$

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

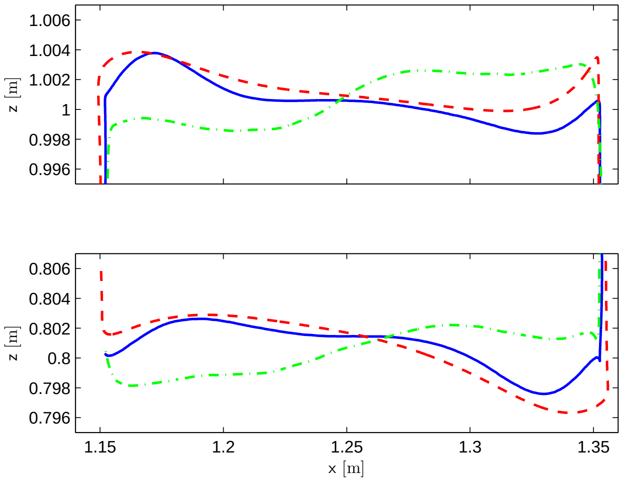

Figure 6: The top side (upper diagram) and bottom side (lower diagram) of the square path in Figure 5 for the true tool position (solid) and tool position estimates using the EKF (dashed) and $\mathcal{T}_{\text{TCP}}(\mathbf{q}_{m,t})$ (dash-dot).

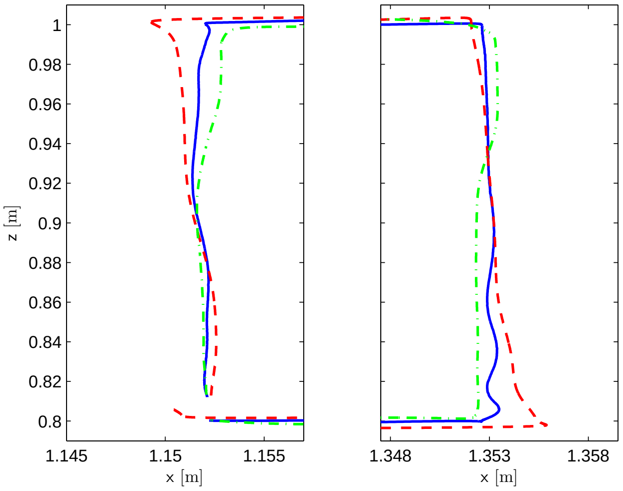

Figure 7: The left side (left diagram) and right side (right diagram) of the square path in Figure 5 for the true tool position (solid) and tool position estimates using the EKF (dashed) and $\mathcal{T}_{\text{TCP}}(\mathbf{q}_{m,t})$ (dash-dot).

[2] The RMSE is computed without considering the first 0.125 seconds where the EKF has a transient behavior.

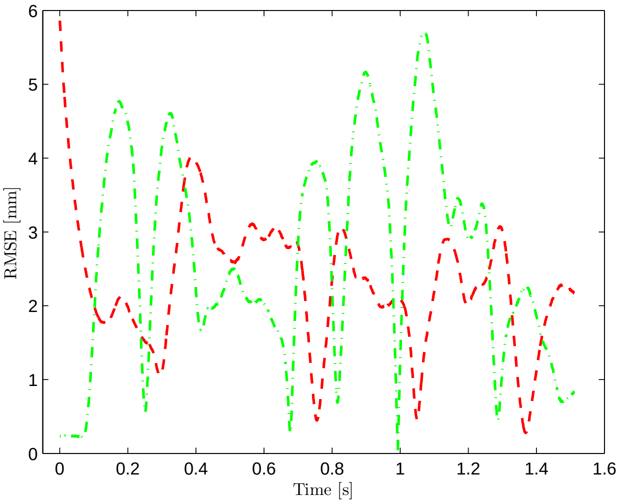

Figure 8: Tool position RMSE for the EKF (dashed) and $\mathcal{T}_{\text{TCP}}(\mathbf{q}_{m,t})$ (dash-dot). The 2-norm of the RMSE-signals, without the first 0.125 seconds, are 0.1246 and 0.1655 for the EKF and $\mathcal{T}_{\text{TCP}}(\mathbf{q}_{m,t})$, respectively.

The result of the PF compared to the EKF can be found in Figure 9 and Figure 10. The PF performs better in the corners, i.e., the estimated path does not go past the corners before it changes. The motive that the PF can handle the problem with the jerk input better than the EKF can be that the particle cloud covers a larger area of the state space. The PF estimate is also closer to the true path, at least at the vertical sides. Figure 11 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 0.0818, which is approximately 66 % of the EKF and 49 % of $\mathcal{T}_{\text{TCP}}(\mathbf{q}_{m,t})$. Note that the transients are not included, i.e., the first 0.125 seconds are removed. The PF converges much faster than the EKF as can be seen clearly in Figure 11. The PF in the proposed implementation is far from real-time and the bias states are needed to control the model errors.

6. Conclusions and Future Work

A sensor fusion approach to find estimates of the tool position, velocity, and acceleration by combining a triaxial accelerometer at the end-effector and the measurements from the motor angles of an industrial robot is presented. The estimation is formulated as a Bayesian problem and two solutions are proposed; the extended Kalman filter and the particle filter. The algorithms were tested on simulated data from a realistic robot model as well as on experimental data.

Sufficiently accurate estimates are produced for simulated data, where the performance both with and without accelerometer measurements are close to the fundamental Cramér-Rao lower bound limit in Monte Carlo simulations. The dynamic performance for experimental data is

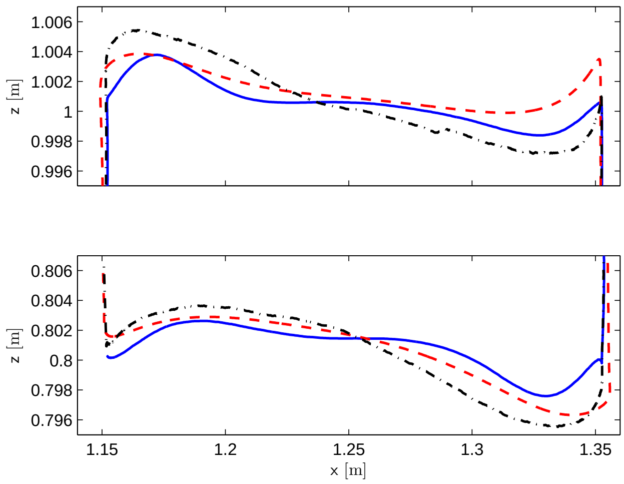

Figure 9: The top side (upper diagram) and bottom side (lower diagram) of the square path in Figure 5 for the true tool position (solid) and tool position estimates using the EKF (dashed) and the PF (dash-dot).

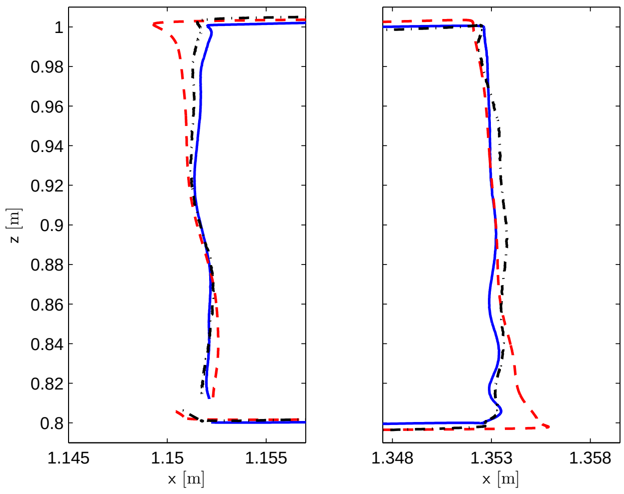

Figure 10: The left side (left diagram) and right side (right diagram) of the square path in Figure 5 for the true tool position (solid) and tool position estimates using the EKF (dashed) and the PF (dash-dot).

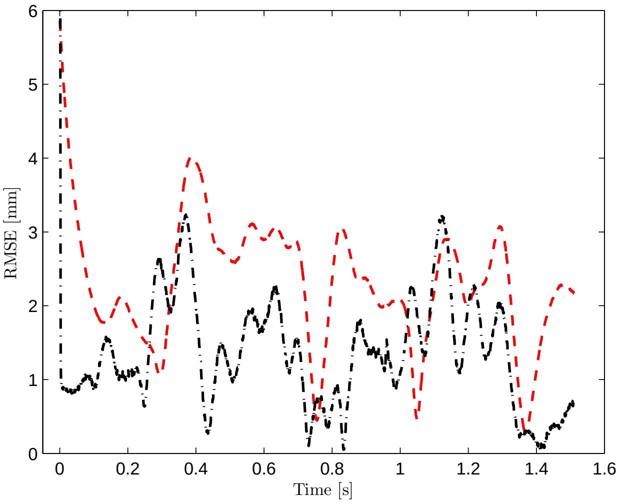

Figure 11: Tool position RMSE for the EKF (dashed) and the PF (dash-dot). The 2-norm of the RMSE-signals, without the first 0.125 seconds, are 0.1246 and 0.0818 for the EKF and the PF, respectively.

also significantly better using the accelerometer method. The velocity estimates are also proven to be much more accurate when the filter uses information from the accelerometer. This is important for control design in order to give a well damped response at the robot arm.

Since the intended use of the estimates is to improve position control using an off-line method, like iterative learning control, there are no real-time issues using the computational 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. The main effect is a larger state space model giving more time-consuming calculations and also a more complex measurement equation. The most time-consuming step in the EKF is the matrix multiplications $\mathbf{F}_t\mathbf{P}_{t|t}\mathbf{F}_t^T$. The two matrix multiplications require in total $4n^3$ flops[3]. For example, going from 2 to 6 DOF increases the computational cost with a factor of 27. For the PF it is not as easy to give a description of the increased computational complexity.

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

Anderson, B., Moore, J. B., 1979. Optimal Filtering. Prentice Hall, Englewood Cliffs, NJ.

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

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

Axelsson, P., Norrlöf, M., 2012. Method to estimate the position and orientation of a triaxial accelerometer mounted to an industrial manipulator. In: Proceedings of the 10th International IFAC Symposium on Robot Control. Dubrovnik, Croatia.

Bergman, N., 1999. Recursive Bayesian estimation: Navigation and tracking applications. Linköping Studies in Science and Technology. Dissertations No. 579, Linköping University, Linköping, Sweden.

Corke, P., March 1996. A robotics toolbox for MATLAB. IEEE Robotics and Automation Magazine 3 (1), 24–32.

Cramér, H., 1946. Mathematical Methods of Statistics. Princeton, NJ: Princeton University Press.

Crossbow Technology, Jan. 2004. Accelerometers, High Sensitivity, LF Series, CXL02LF3. URL: http://www.xbow.com.

De Luca, A., Schröder, D., Thümmel, M., April 2007. An acceleration-based state observer for robot manipulators with elastic joints. In: Proceedings of the IEEE International Conference on Robotics and Automation. Roma, Italy, pp. 3817–3823.

Doucet, A., de Freitas, N., Gordon, N. (Eds.), 2001. Sequential Monte Carlo Methods in Practice. Springer Verlag.

Doucet, A., Godsill, S., Andrieu, C., 2000. On sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and Computing 10 (3), 197–208.

Gordon, N. J., Salmond, D. J., Smith, A., 1993. A novel approach to nonlinear/non-Gaussian Bayesian state estimation. In: IEE Proceedings on Radar and Signal Processing. Vol. 140. pp. 107–113.

Gunnarsson, S., Norrlöf, M., Sep. 2004. Iterative learning control of a flexible robot arm using accelerometers. In: IEEE Conference on Control Applications. Taipei, Taiwan, pp. 1012–1016.

Gunnarsson, S., Norrlöf, M., Hovland, G., Carlsson, U., Brogårdh, T., Svensson, T., Moberg, S., Apr. 2001. Pathcorrection for an industrial robot. European Patent Application No. EP1274546.

Gustafsson, F., 2010. Statistical Sensor Fusion. Studentlitteratur, Lund, Sweden.

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

Jazwinski, A. H., 1970. Stochastic Processes and Filtering Theory. Vol. 64 of Mathematics in Science and Engineering. Academic Press.

Jensfelt, P., 2001. Approaches to mobile robot localization in indoor environments. Ph.D. thesis, Royal Institute of Technology.

Jeon, S., Tomizuka, M., Katou, T., March 2009. Kinematic Kalman filter (KKF) for robot end-effector sensing. Journal of Dynamic Systems, Measurement, and Control 131 (2).

Kailath, T., Sayed, A., Hassibi, B., 2000. Linear Estimation. Information and System Sciences. Prentice Hall, Upper Saddle River, New Jersey.

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

Karlsson, R., Norrlöf, M., September 2004. Bayesian position estimation of an industrial robot using multiple sensors. In: Proceedings of the IEEE Conference on Control Applications. Taipei, Taiwan, pp. 303–308.

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

Kay, S., 1993. Fundamentals of Statistical Signal Processing. Prentice Hall.

Kozlowski, K., 1998. Modelling and Identification in Robotics. Springer-Verlag.

Kwok, C., Fox, D., Meila, M., March 2004. Real-time particle filters. Proceedings of the IEEE 92 (3), 469–484.

Leica Geosystems, Metrology Products, 2008. Case study ABB robotics - Västerås. Http://www.leica-geosystems.com.

[3] A flop is one of the elementary scalar operations +, −, ∗, /.

Lertpiriyasuwat, V., Berg, M. C., Buffinton, K. W., March 2000. Extended Kalman filtering applied to a two-axis robotic arm with flexible links. The International Journal of Robotics Research 19 (3), 254–270.

Li, Y. F., Chen, X. B., September 2001. End-point sensing and state observation of a flexible-link robot. IEEE/ASME Transactions on Mechatronics 6 (3), 351–356.

Moberg, S., December 2010. Modeling and control of flexible manipulators. Linköping Studies in Science and Technology. Dissertations. No. 1349, Linköping University, Linköping, Sweden, http://www.control.isy.liu.se/publications/.

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

Quigley, M., Brewer, R., Soundararaj, S. P., Pradeep, V., Le, Q., Ng, A. Y., October 2010. Low-cost accelerometers for robotic manipulator perception. In: Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. Taipei, Taiwan, pp. 6168–6174.

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

Ristic, B., Arulampalam, S., Gordon, N., 2004. Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House.

Sciavicco, L., Siciliano, B., 2000. Modelling and Control of Robot Manipulators. Springer.

Sorenson, H. W., 1988. Recursive Estimation for Nonlinear Dynamic Systems. In Bayesian Analysis of Time Series and Dynamic Models (Ed. J. C. Spall), Dekker.

Spong, M. W., December 1987. Modeling and control of elastic joint robots. Journal of Dynamic Systems, Measurement, and Control 109, 310–319.

Tichavsky, P., Muravchik, P., Nehorai, A., 1998. Posterior cramer-rao bounds for discrete-time nonlinear filtering. IEEE Transactions on Signal Processing 46 (5), 1386–1396.

Van Trees, H., 1968. Detection, Estimation and Modulation Theory. Wiley, New York.