10th IFAC Symposium on Robot Control
International Federation of Automatic Control
September 5-7, 2012. Dubrovnik, Croatia

Evaluation of Six Different Sensor Fusion Methods for an Industrial Robot using Experimental Data ⋆

Patrik Axelsson

Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden (e-mail: axelsson@isy.liu.se).

Abstract: Experimental evaluations for path estimation are performed on an ABB IRB4600 robot. Different observers using Bayesian techniques with different estimation models are proposed. The estimated paths are compared to the true path measured by a laser tracking system. There is no significant difference in performance between the six observers. Instead, execution time, model complexities and implementation issues have to be considered when choosing the method.

Keywords: Estimation, Extended Kalman filter, Particle filter, Accelerometer, Industrial robots

1. INTRODUCTION

The first industrial robots were big and heavy with rigid links and joints. The development of new robot models has been focused on increasing the performance along with cost reduction, safety improvement and introduction of new functionalities as described in Brogårdh (2007). One way to reduce the cost is to lower the weight of the robot which conduces to lower mechanical stiffness in the links. Also, the components of the robot are changed such that the cost is reduced, which can infer larger individual variations and unwanted nonlinearities. The most crucial component, when it comes to flexibilities, is the gearbox. The gearbox has changed more and more to a flexible component, where the flexibilities have to be described by nonlinear relations in the models in order to have good control performance. There is therefore a demand of new approaches for motion control where less accurate models can be sufficient. One solution can be to estimate the position and orientation of the end-effector along the path and then use the estimated position and orientation in the feedback loop of the motion controller. The most simple observer is to use the measured motor angular positions in the forward kinematic model to get the position and orientation of the end-effector. The performance is insufficient and the reason is that the oscillations on the arm side do not influence the motor side of the gearbox that much due to the flexibilities. The flexibilities can also distort the oscillations of the arm side. The observer can consequently not track the true position and another observer is therefore needed. The observer requires a dynamic model of the robot in order to capture the oscillations on the arm side of the gearbox as well as more measurements than only the motor angular positions. One way to obtain information about the oscillations on the arm side can be to attach an accelerometer on the robot, e.g. at the end-effector.

⋆ This work was supported by Vinnova Excellence Center LINK-SIC.

A natural question is, how to estimate the arm angular positions from the measured acceleration as well as the measured motor angular positions. A common solution for this kind of problems is to apply sensor fusion methods for state estimation. The acceleration of the end-effector as well as the measured motor angular positions can be used as measurements in e.g. an *extended Kalman filter* (EKF) or *particle filter* (PF). In Karlsson and Norrlöf (2004, 2005), and Rigatos (2009) the EKF and PF are evaluated on a flexible joint model using simulated data only. The estimates from the EKF and PF are also compared with the theoretical Cramér-Rao lower bound in Karlsson and Norrlöf (2005) to see how good the filters are. An evaluation of the EKF using experimental data is presented in Henriksson et al. (2009), and in Jassemi-Zargani and Necsulescu (2002) with different types of estimation models. A method using the measured acceleration of the end-effector as input instead of using it as measurements is described in De Luca et al. (2007). The observer, in this case, is a linear dynamic observer using pole placement, which has been evaluated on experimental data.

2. STATE ESTIMATION

The estimation problem for the discrete time nonlinear state space model

$$x_{k+1} = f(x_k, u_k) + g(x_k)v_k, \tag{1a}$$

$$y_k = h(x_k, u_k) + e_k, \tag{1b}$$

is to find the state vector $x_k \in \mathbb{R}^{n_x}$ at time k given the measurements $y_k \in \mathbb{R}^{n_y}$ $k = 1, \ldots, N$. The estimation problem can be seen as calculation/approximation of the posterior density $p(x_k|y_{1:l})$ using all measurements up to time l, where $y_{1:l} = \{y_1, \ldots, y_l\}$. There are two types of problems, filtering and smoothing. Filtering uses only measurements up to present time and smoothing uses future measurements also, i.e., $l = k$ for filtering and $l > k$ for smoothing. Using Bayes' law, and the Markov property for the state space model, repeatedly, the optimal solution for the Bayesian inference can be obtained. See

Jazwinski (1970) for details. The solution to the Bayesian inference can in most cases not be given by an analytical expression. For the special case of linear dynamics, linear measurements and additive Gaussian noise the Bayesian recursions have an analytical solution, which is known as the *Kalman filter* (KF). Approximative methods must be used for nonlinear and non-Gaussian systems. Here three approximative solutions are considered; the *extended Kalman filter* (EKF), the *extended Kalman smoother* (EKS) and the *particle filter* (PF).

EKF: The EKF (Anderson and Moore, 1979) solves the Bayesian recursions using a first order Taylor expansion of the nonlinear system equations around the previous estimate. The process noise v_k and measurement noise e_k are assumed to be Gaussian with zero mean and covariance matrices Q_k and R_k, respectively. The time and measurement updates are

$$\hat{x}_{k|k-1} = f\left(\hat{x}_{k-1|k-1}, u_k\right) \tag{2a}$$

$$\text{TU:}\quad P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^{\mathsf{T}} + g\left(\hat{x}_{k-1|k-1}\right)Q_k g^{\mathsf{T}}\left(\hat{x}_{k-1|k-1}\right) \tag{2b}$$

$$K_k = P_{k|k-1}H_k^{\mathsf{T}}\left(H_k P_{k|k-1}H_k^{\mathsf{T}} + R_k\right)^{-1} \tag{3a}$$

$$\text{MU:}\quad \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(y_k - h\left(\hat{x}_{k|k-1}, u_k\right)\right) \tag{3b}$$

$$P_{k|k} = \left(I - K_k H_k\right)P_{k|k-1} \tag{3c}$$

where

$$F_{k-1} = \left.\frac{\partial f\left(x, u_k\right)}{\partial x}\right|_{x=\hat{x}_{k-1|k-1}}, \quad H_k = \left.\frac{\partial h(x, u_k)}{\partial x}\right|_{x=\hat{x}_{k|k-1}}.$$

The notation $\hat{x}_{k|k}$, $P_{k|k}$, $\hat{x}_{k|k-1}$ and $P_{k|k-1}$ means estimates of the state vector x and covariance matrix P at time k using measurements up to time k and $k-1$, respectively.

EKS: The EKS (Yu et al., 2004) solves the Bayesian recursions in the same way as the EKF. The difference is that future measurements are available. First, the EKF equations are used forward in time, then the backward time recursion

$$\hat{x}^s_{k|N} = \hat{x}_{k|k} + P_{k|k}F_k^{\mathsf{T}}P^{-1}_{k+1|k}\left(\hat{x}^s_{k+1|N} - \hat{x}_{k+1|k}\right), \tag{4a}$$

$$P^s_{k|N} = P_{k|k} + P_{k|k}F_k^{\mathsf{T}}P^{-1}_{k+1|k}\left(P^s_{k+1|N} - P_{k+1|k}\right) \times P^{-1}_{k+1|k}F_k P_{k|k}, \tag{4b}$$

is used, where F_k is given above.

PF: The PF (Doucet et al., 2001; Gordon et al., 1993) solves the Bayesian recursions using stochastic integration. The PF approximates the posterior density $p(x_k|y_{1:k})$ by a large set of N particles $\left\{x^i_k\right\}_{i=1}^N$, where each particle has an assigned relative weight w^i_k, chosen such that $\sum_{i=1}^N w^i_k = 1$. The PF updates the particle location and the corresponding weights recursively with each new observed measurement. Theoretical results show that the approximated posterior density converges to the true density when the number of particles tends to infinity, see e.g. Doucet et al. (2001). The PF is summarised in Algorithm 1, where the proposal density $p^{\text{prop}}\left(x^i_{k+1}|x^i_k, y_{k+1}\right)$ can be chosen arbitrary as long as it is possible to draw samples from it. In this work the optimal proposal density, approximated by an EKF, is used. See Doucet et al. (2000), and Gustafsson (2010) for details. The state estimate for each sample k is often chosen as the minimum mean square estimate

$$\hat{x}_{k|k} = \mathrm{E}\left[x_k\right] = \int_{\mathbb{R}^{n_x}} x_k p(x_k|y_{1:k})\,\mathrm{d}x_k \approx \sum_{i=1}^N w^i_k x^i_k. \tag{5}$$

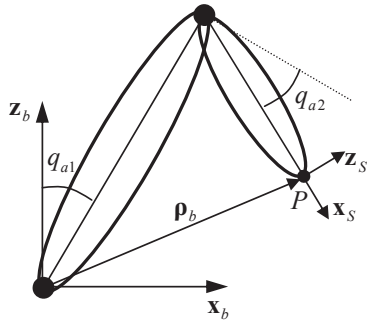

Fig. 1. A two degrees of freedom robot model. The links are assumed to be rigid and the joints are described by a two mass system connected by a spring damper pair. The accelerometer is attached in the point $\mathcal{P}$.

Algorithm 1 The Particle Filter (PF)

(1) Generate N samples $\left\{x^i_0\right\}_{i=1}^N$ from $p(x_0)$.
(2) Compute

$$w^i_k = w^i_{k-1} \cdot \frac{p\left(y_k|x^i_k\right)p\left(x^i_k|x^i_{k-1}\right)}{p^{\text{prop}}\left(x^i_k|x^i_{k-1}, y_k\right)}$$

and normalise, i.e., $\bar{w}^i_k = w^i_k / \sum_{j=1}^N w^j_k$, $i = 1, \ldots, N$.
(3) (Optional). Generate a new set $\left\{x^{i\star}_k\right\}_{i=1}^N$ by resampling with replacement N times from $\left\{x^i_k\right\}_{i=1}^N$, with probability $\bar{w}^i_k = \Pr\left\{x^{i\star}_k = x^i_k\right\}$ and reset the weights to $1/N$.
(4) Generate predictions from the proposal density $x^i_{k+1} \sim p^{\text{prop}}\left(x_{k+1}|x^{i\star}_k, y_{k+1}\right)$, $i = 1, \ldots, N$.
(5) Increase k and continue to step 2.

3. DYNAMIC MODELS

3.1 Robot Model

This section describes a two-link nonlinear flexible robot model, corresponding to joint two and three for a serial six DOF industrial robot. The dynamic robot model is a joint flexible two-axes model from Moberg et al. (2008), see Figure 1. Each link is modelled as a rigid-body and the joints are modelled as a spring damping pair with nonlinear spring torque and linear damping. The deflection in each joint is given by the arm angle q_{ai} and the motor angle q_{mi}. Let

$$q_a = \left(q_{a1}\ q_{a2}\right)^{\mathsf{T}}, \quad q^a_m = \left(q_{m1}/\eta_1\ q_{m2}/\eta_2\right)^{\mathsf{T}},$$
$$\tau^a_m = \left(\tau_{m1}\eta_1\ \tau_{m2}\eta_2\right)^{\mathsf{T}},$$

where τ_{mi} is the motor torque and $\eta_i = q_{mi}/q_{ai} > 1$ is the gear ratio. A dynamic model can be derived as

$$M_a(q_a)\ddot{q}_a + C(q_a, \dot{q}_a) + G(q_a) + A(q) = 0, \tag{6a}$$

$$M_m\ddot{q}^a_m + F(\dot{q}^a_m) - A(q) = \tau^a_m, \tag{6b}$$

using Lagrange's equation, where $A(q) = T(q_a - q^a_m) + \bar{D}(\dot{q}_a - \dot{q}^a_m)$, M_a and M_m are the inertia matrices for

the arms and motors, respectively, C is the Coriolis- and centrifugal terms, G is the gravity torque, F is the friction torque, T is the spring stiffness torque and $\bar{D}$ is the damping torque. See Moberg et al. (2008) for a full description of the model.

3.2 Accelerometer Model

The accelerometer attached to the robot measures the acceleration due to the motion the robot performs, the gravity component and in addition some measurement noise are introduced. When modelling the accelerometer it is also important to include a bias term. The acceleration is measured in a frame Ox_sz_s fixed to the accelerometer relative an inertial frame, which is chosen to be the world fixed base frame Ox_bz_b, see Figure 1. The acceleration in Ox_sz_s can thus be expressed as

$$\ddot{\rho}_s(q_a) = \mathcal{R}_{b/s}(q_a)\left(\ddot{\rho}_b(q_a) + G_b\right) + \mathfrak{b}^{\text{ACC}}, \tag{7}$$

where $\ddot{\rho}_b(q_a)$ is the acceleration due to the motion and $G_b = (0\ g)^\mathsf{T}$ models the gravitation, both expressed in the base frame Ox_bz_b. The bias term is denoted by $\mathfrak{b}^{\text{ACC}}$ and is expressed in Ox_sz_s. $\mathcal{R}_{b/s}(q_a)$ is a rotation matrix that represents the rotation from frame Ox_bz_b to frame Ox_sz_s.

The vector $\ddot{\rho}_b(q_a)$ can be calculated as the second derivative of $\rho_b(q_a)$ which is shown in Figure 1. Using the forward kinematic relation, the vector ρ_b can be written as

$$\rho_b(q_a) = \Upsilon_{\text{ACC}}(q_a), \tag{8}$$

where Υ_{ACC} is a nonlinear function. Taking the derivative of (8) with respect to time twice gives

$$\ddot{\rho}_b(q_a) = \frac{d^2}{dt^2}\Upsilon_{\text{ACC}}(q_a) = \mathcal{J}_{\text{ACC}}(q_a)\ddot{q}_a + \left(\frac{d}{dt}\mathcal{J}_{\text{ACC}}(q_a)\right)\dot{q}_a, \tag{9}$$

where $\mathcal{J}_{\text{ACC}}(q_a) = \frac{\partial \Upsilon_{\text{ACC}}}{\partial q_a}$ is the Jacobian matrix. The final expression for the acceleration measured by the accelerometer is given by (7) and (9).

3.3 Modelling of Bias

In (7) a bias component is included in the accelerometer model, which is unknown and may vary over time. The bias component $\mathfrak{b}_k = \left(\mathfrak{b}_k^1 \ \dots \ \mathfrak{b}_k^{n_\mathfrak{b}}\right)^\mathsf{T}$ can be modelled as a random walk, i.e.,

$$\mathfrak{b}_{k+1} = \mathfrak{b}_k + v_k^\mathfrak{b}, \tag{10}$$

where $v_k^\mathfrak{b} = \left(v_k^{\mathfrak{b},1} \ \dots \ v_k^{\mathfrak{b},n_\mathfrak{b}}\right)^\mathsf{T}$ is process noise and $n_\mathfrak{b}$ is the number of bias terms. The random walk model is then included in the estimation problem and the bias terms are estimated simultaneously as the other states. Let a state space model without any bias states be given by (1). The augmented model with the bias states can then be written as

$$\begin{pmatrix} x_{k+1} \\ \mathfrak{b}_{k+1} \end{pmatrix} = \begin{pmatrix} f(x_k, u_k) \\ \mathfrak{b}_k \end{pmatrix} + \begin{pmatrix} g(x_k) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} v_k \\ v_k^\mathfrak{b} \end{pmatrix}, \tag{11}$$

$$y_k = h(x_k, u_k) + \mathcal{C}\mathfrak{b}_k + e_k, \tag{12}$$

where $\mathbf{I}$ and $\mathbf{0}$ are the identity and null matrices, respectively, and $\mathcal{C} \in \mathbb{R}^{n_y \times n_\mathfrak{b}}$ is a constant matrix

4. ESTIMATION MODELS

Four different estimation models are presented using the robot and acceleration models described in Section 3. The process noise v_k and measurement noise e_k are in all four estimation models assumed to be Gaussian with zero mean and covariance matrices Q and R, respectively.

4.1 Nonlinear Estimation Model

Let the state vector be

$$x = \left(x_1^\mathsf{T}\ x_2^\mathsf{T}\ x_3^\mathsf{T}\ x_4^\mathsf{T}\right)^\mathsf{T} = \left(q_a^\mathsf{T}\ q_m^{a,\mathsf{T}}\ \dot{q}_a^\mathsf{T}\ \dot{q}_m^{a,\mathsf{T}}\right)^\mathsf{T}, \tag{13}$$

where $q_a = (q_{a1}\ q_{a2})^\mathsf{T}$ are the arm positions and $q_m^a = (q_{m1}^a\ q_{m2}^a)^\mathsf{T}$ are the motor positions on the arm side of the gearbox. Let also the input vector $u = \tau_m^a$. Taking the derivative of x with respect to time and using (6) give

$$\dot{x} = \begin{pmatrix} x_3 \\ x_4 \\ M_a^{-1}(x_1)\left(-C(x_1, x_3) - G(x_1) - A(x)\right) \\ M_m^{-1}\left(u - F(x_4) + A(x)\right) \end{pmatrix}. \tag{14}$$

In order to use the estimation methods described in Section 2 the continuous state space model (14) has to be discretised. The time derivative of the state vector can be approximated using Euler forward according to

$$\dot{x} = \frac{x_{k+1} - x_k}{T_s}, \tag{15}$$

where T_s is the sample time. Taking the right hand side in (14) and (15) equal to each other give the nonlinear discrete state space model

$$x_{k+1} = \begin{pmatrix} x_{1,k} + T_s x_{3,k} \\ x_{2,k} + T_s x_{4,k} \\ x_{3,k} + T_s M_a^{-1}(x_{1,k}) B(x_k) \\ x_{4,k} + T_s M_m^{-1}\left(u_k - F(x_{4,k}) + A(x_k)\right) \end{pmatrix}. \tag{16}$$

where $B(x_k) = -C(x_{1,k}, x_{3,k}) - G(x_{1,k}) - A(x_k)$. The noise is modelled as a torque disturbance on the arms and motors, giving a model according to (1a) where $f(x_k, u_k)$ is given by the right hand side in (16) and

$$g(x_k) = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ T_s M_a^{-1}(x_{1,k}) & \mathbf{0} \\ \mathbf{0} & T_s M_m^{-1} \end{pmatrix}, \tag{17}$$

where $\mathbf{0}$ is a two by two null matrix and the noise $v_k \in \mathbb{R}^4$.

The measurements are the motor positions q_m^a and the end-effector acceleration $\ddot{\rho}_s^M(q_a)$. The measurement model (1b) can therefore be written as

$$y_k = \begin{pmatrix} x_{2,k} \\ \mathcal{R}_{b/s}(x_{1,k})\left(\ddot{\rho}_b(x_k) + G_b\right) \end{pmatrix} + e_k, \tag{18}$$

where $\ddot{\rho}_b(x_k)$ is given by (9) and $e_k \in \mathbb{R}^4$. In (9) are q_a and $\dot{q}_a$ given as states, whereas $\ddot{q}_a$ is given by the third row in (14). The accelerometer bias $\mathfrak{b}_k^{\text{ACC}} = \left(\mathfrak{b}_k^{\text{ACC},\text{x}}\ \mathfrak{b}_k^{\text{ACC},\text{z}}\right)^\mathsf{T}$ is modelled as it is described in Section 3.3 with

$$\mathcal{C} = \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix}, \tag{19}$$

where $\mathbf{I}$ and $\mathbf{0}$ are two by two identity and null matrices, respectively.

4.2 Estimation Model with Linear Dynamic

A linear dynamic model with arm positions, velocities and accelerations as state variables is suggested. Let the state vector be

$$x = \left(x_1^\mathsf{T}\ x_2^\mathsf{T}\ x_3^\mathsf{T}\right) = \left(q_a^\mathsf{T}\ \dot{q}_a^\mathsf{T}\ \ddot{q}_a^\mathsf{T}\right)^\mathsf{T}, \tag{20}$$

where $q_a = (q_{a1}\ q_{a2})^\mathsf{T}$ are the arm positions. This yields the following state space model in discrete time

$$x_{k+1} = Fx_k + G_u u_k + G_v v_k, \tag{21}$$

where u_k is the input vector and the process noise $v_k \in \mathbb{R}^2$. The constant matrices are given by

$$F = \begin{pmatrix} \mathbf{I} & T_s\mathbf{I} & \frac{T_s^2}{2}\mathbf{I} \\ \mathbf{0} & \mathbf{I} & T_s\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix}, \quad G_u = G_v = \begin{pmatrix} \frac{T_s^3}{6}\mathbf{I} \\ \frac{T_s^2}{2}\mathbf{I} \\ T_s\mathbf{I} \end{pmatrix} \tag{22}$$

where $\mathbf{I}$ and $\mathbf{0}$ are two by two identity and null matrices, respectively. The input, u_k, is the arm jerk reference, i.e., the differentiated arm angular acceleration reference.

The motor positions are calculated from (6a) where the spring is linear, i.e.,

$$T(q_m^a - q_a) = \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix}(q_m^a - q_a) = K(q_m^a - q_a). \tag{23}$$

The damping term is small compared to the other terms (Karlsson and Norrlöf, 2005) and is therefore neglected for simplicity. The measurement model for the accelerometer is the same as in (18) where $\ddot{q}_{a,k}$ is a state in this case. The measurement model can now be written as

$$y_k = \begin{pmatrix} q_{m,k}^a \\ \mathcal{R}_{b/s}(x_{1,k})\left(\ddot{\rho}_b(x_k) + G_b\right) \end{pmatrix} + e_k. \tag{24}$$

where $e_k \in \mathbb{R}^4$ and

$$\begin{aligned} q_{m,k}^a =& x_{1,k} + K^{-1}\left(M_a(x_{1,k})x_{3,k}\right. \\ &\left. + C(x_{1,k}, x_{2,k}) + G(x_{1,k})\right), \end{aligned} \tag{25a}$$

$$\ddot{\rho}_b(x_k) = J_{\text{ACC}}(x_{1,k})x_{3,k} + \left(\frac{d}{dt}J_{\text{ACC}}(x_{1,k})\right)x_{2,k}. \tag{25b}$$

Once again, the accelerometer bias $\mathfrak{b}_k^{\text{ACC}}$ is modelled according to Section 3.3. However, the estimation result is improved if bias components for the motor measurements also are included. The explanation is model errors in the measurement equation. The total bias component is $\mathfrak{b}_k = \left(\mathfrak{b}_k^{q_m,\mathsf{T}}\ \mathfrak{b}_k^{\text{ACC},\mathsf{T}}\right)^\mathsf{T}$, where $\mathfrak{b}_k^{q_m} = \left(\mathfrak{b}_k^{q_{m1}}\ \mathfrak{b}_k^{q_{m2}}\right)^\mathsf{T}$ and $\mathfrak{b}_k^{\text{ACC}} = \left(\mathfrak{b}_k^{\text{ACC},\mathsf{x}}\ \mathfrak{b}_k^{\text{ACC},\mathsf{z}}\right)^\mathsf{T}$. The matrix $\mathcal{C}$ in (12) is for this model a four by four identity matrix.

4.3 Linear Estimation Model with Acceleration as Input

In De Luca et al. (2007) a model using the arm angular acceleration as input is presented. Identifying the third row in (14) as the arm angular acceleration and use that as an input signal with (13) as state vector give the following model,

$$\dot{x} = \begin{pmatrix} x_3 \\ x_4 \\ \ddot{q}_a^{\text{IN}} \\ M_m^{-1}\left(u - F(x_4) + A(x)\right) \end{pmatrix}, \tag{26}$$

where $\ddot{q}_a^{\text{IN}}$ is the new input signal. If the friction, spring stiffness and damping are modelled with linear relations, then

$$\dot{x} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ M_m^{-1}K & -M_m^{-1}K & M_m^{-1}D & -M_m^{-1}(D + F_d) \end{pmatrix} x + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & M_m^{-1} \end{pmatrix} \begin{pmatrix} \ddot{q}_a^{\text{IN}} \\ u \end{pmatrix}, \tag{27}$$

where

$$K = \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix}, \quad D = \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix}, \quad F_d = \begin{pmatrix} \eta_1^2 f_{d1} & 0 \\ 0 & \eta_1^2 f_{d2} \end{pmatrix}.$$

The linear state space model is discretised using *zero order hold* (ZOH). ZOH is used instead of Euler forward since it gives a better result and an explicit solution exist when the model is linear. The only remaining measurements are the motor positions, which give a linear measurement model according to

$$y_k = (\mathbf{0}\ \mathbf{I}\ \mathbf{0}\ \mathbf{0})\, x_k + e_k, \tag{28}$$

where $e_k \in \mathbb{R}^2$. Note that the arm angular acceleration $\ddot{q}_a^{\text{IN}}$ is not measured direct, instead it has to be calculated from the accelerometer signal using (7) and (9), which is possible as long as the Jacobian $J_{\text{ACC}}(x_{1,k})$ has full rank.

4.4 Nonlinear Estimation Model with Acceleration as Input

The linear model presented in Section 4.3 is here reformulated as a nonlinear model. Given the model in (26) and using Euler forward for discretisation give

$$x_{k+1} = \begin{pmatrix} x_{1,k} + T_s x_{3,k} \\ x_{2,k} + T_s x_{4,k} \\ x_{3,k} + T_s \ddot{q}_{a,k}^{\text{IN}} \\ x_{4,k} + T_s M_m^{-1}\left(u_k - F(x_{4,k}) + A(x_k)\right) \end{pmatrix}. \tag{29}$$

The noise model is assumed to be the same as in Section 4.1, and the measurement is the same as in (28).

5. EXPERIMENTS ON AN ABB IRB4600 ROBOT

5.1 Experimental Setup

The accelerometer used in the experiments is a triaxial accelerometer from Crossbow Technology, with a range of $\pm 2\,\text{g}$, and a sensitivity of $1\,\text{V/g}$ (Crossbow Technology, 2004). The orientation and position of the accelerometer are estimated using the method described in Axelsson and Norrlöf (2012). All measured signals are synchronous and sampled with a rate of 2 kHz. The accelerometer measurements are filtered with an LP-filter before any estimation method is applied to better reflect the tool movement. The path used in the evaluation is illustrated in Figure 2 and it is programmed such that only joint two and three are moved. Moreover, the wrist is configured such that the couplings to joint two and three are minimised. The dynamic model parameters are obtained using a grey-box identification method described in Wernholt and Moberg (2011).

It is not possible to get measurements of the true state variables, as is the case for simulation, instead, only the true trajectory of the tool, more precise the TCP, x and z coordinates, is used for evaluation. The true trajectory

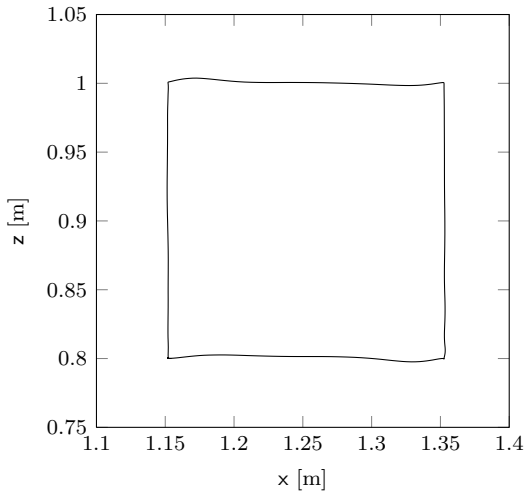

Fig. 2. Measured path for the end-effector used for experimental evaluations.

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, 2008). However, the measured tool position is not synchronised with the other measured signals. Resampling of the measured signal and a manual synchronisation 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

Six observers using the four different estimation models described in Section 4 are evaluated. The observers are based on the EKF, EKS, PF or a linear dynamic observer using pole placement (Franklin et al., 2002).

OBS1: EKF with the nonlinear model in Section 4.1.
OBS2: EKS with the nonlinear model in Section 4.1.
OBS3: EKF with the linear state model and nonlinear measurement model in Section 4.2.
OBS4: PF with the linear state model and nonlinear measurement model in Section 4.2.
OBS5: EKF with the nonlinear model where the acceleration of the end-effector is input, see Section 4.4 .
OBS6: Linear dynamic observer using pole placement with the linear model where the acceleration of the end-effector is input, see Section 4.3. (De Luca et al., 2007)

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.,

$$\begin{pmatrix} \hat{\mathsf{x}}_k \\ \hat{\mathsf{z}}_k \end{pmatrix} = \Upsilon_{\text{TCP}}(\hat{q}_{a,k}), \tag{30}$$

where $\hat{q}_{a,k}$ is the result from one of the six observers at time k. The result is presented with diagrams of the true and estimated paths for the horizontal parts of the path in Figure 2. The path error

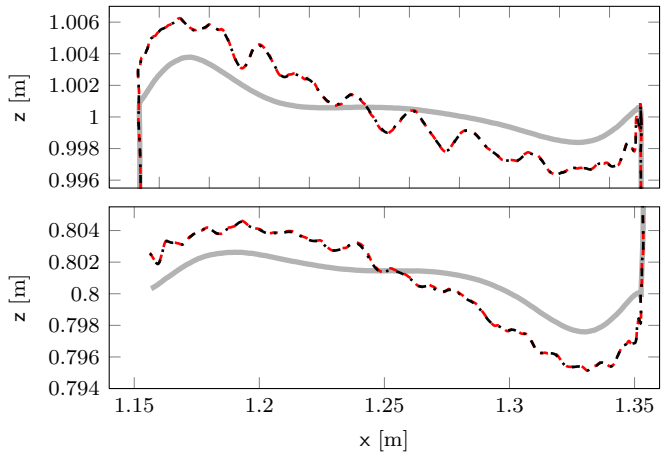

Fig. 3. The two horizontal sides of the true path (solid), and the estimated path using OBS1 (dashed), and OBS2 (dash-dot).

$$\mathsf{e}_k = \sqrt{(\mathsf{x}_k - \hat{\mathsf{x}}_k)^2 + (\mathsf{z}_k - \hat{\mathsf{z}}_k)^2}, \tag{31}$$

where x_k, $\hat{\mathsf{x}}_k$, z_k and $\hat{\mathsf{z}}_k$ are the true and estimated position for the tool in the x- and z-direction at time k, as well as the *root mean square error* (RMSE)

$$\epsilon = \sqrt{\frac{1}{N}\sum_{k=1}^{N} \mathsf{e}_k^2}, \tag{32}$$

where N is the number of samples, are also used for evaluation. Moreover, the first 250 samples are always removed because of transients. The execution time for the observers is also examined. Note that the execution times are with respect to the current MATLAB implementation. The execution time may be faster after some optimisation of the MATLAB code or by using another programming language, e.g. C++. The observers are first paired such that the same estimation model is used, hence OBS1–OBS2, OBS3–OBS4, and OBS5–OBS6 are compared. After that, the best observers from each pair are compared to each other.

OBS1 and OBS2. It is expected that OBS2 (EKS with nonlinear model) will give a better result than OBS1 (EKF with nonlinear model) since the EKS uses both previous and future measurements. This is not the case as can be seen in Figure 3. The reason for this can be model errors and in particular the nonlinearities in the joint stiffness.

One interesting observation is the higher orders of oscillations in the estimated paths. The oscillations can be reduced if the covariance matrix Q for the process noise is decreased. However, this leads to a larger path error. The RMSE values can be found in Table 1. The table shows that OBS2 is slightly better than OBS1.

With the current MATLAB implementation the execution times are around five and seven seconds, respectively, and the total length of the measured path is four seconds, hence none of the observers are real-time. Most of the time is spent in evaluating the Jacobian H_k in the EKF and it is probably possible to decrease that time with a more efficient implementation. Another possibility can be to run the filter with a lower sample rate. OBS2 is slower since an EKF is used first and then the backward time recursion in (4). However, most of the time in the EKS is spent in the EKF. As a matter of fact, the execution time is irrelevant

Table 1. RMSE values of the path error e for the end-effector position given in mm for the six observers.

	OBS1	OBS2	OBS3	OBS4	OBS5	OBS6
ϵ	1.5704	1.5664	2.3752	1.5606	1.6973	1.7624

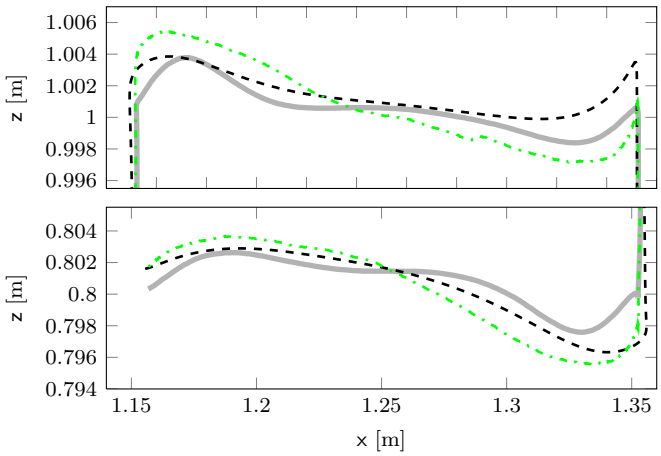

Fig. 4. The two horizontal sides of the true path (solid), and the estimated path using OBS3 (dashed), and OBS4 (dash-dot).

for OBS2 since the EKS uses future measurements and has to be implemented offline.

None of the two observers can be said to be better than the other in terms of estimation performance and execution time. The decision is whether future measurement can be used or not. OBS1 is chosen as the one that will be compared with the other observers.

OBS3 and OBS4. Figure 4 shows that the estimated paths follow the true path for both observers. It can be noticed that the estimate for OBS3 (EKF with linear dynamic model) goes somewhat past the corners before it changes direction and that OBS4 (PF with linear dynamic model) performs better in the corners. The estimate for OBS4 is also closer to the true path, at least at the vertical sides. The RMSE values of the path error for OBS3 and OBS4 are presented in Table 1. The RMSE for OBS4 is approximately two-thirds of the RMSE for OBS3.

The MATLAB implementation of OBS3 is almost real-time, just above four seconds, and the execution time for OBS4 is about ten hours. The execution time for OBS3 can be reduced to real-time without losing performance if the measurements are decimated to approximately 200 Hz.

The best observer in terms of the path error is obviously OBS4 but if the execution time is of importance, OBS3 is preferable. OBS4 will be compared with the other observers since the path error is of more interest in this paper.

OBS5 and OBS6. OBS6 (linear model with acceleration as input using pole placement) performs surprisingly good although a linear time invariant model is used, see Figure 5. It can also be seen that OBS5 (linear model with acceleration as input using EKF) performs a bit better. OBS5 also has a higher order oscillation as was the case with OBS1 and OBS2. This is a matter of tuning where less oscillations induce higher path error. The RMSE values of the path error are showed in Table 1.

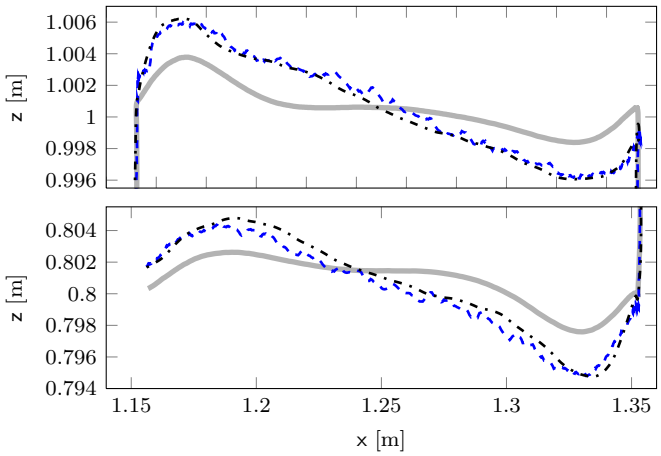

Fig. 5. The two horizontal sides of the true path (solid), and the estimated path using OBS5 (dashed), and OBS6 (dash-dot).

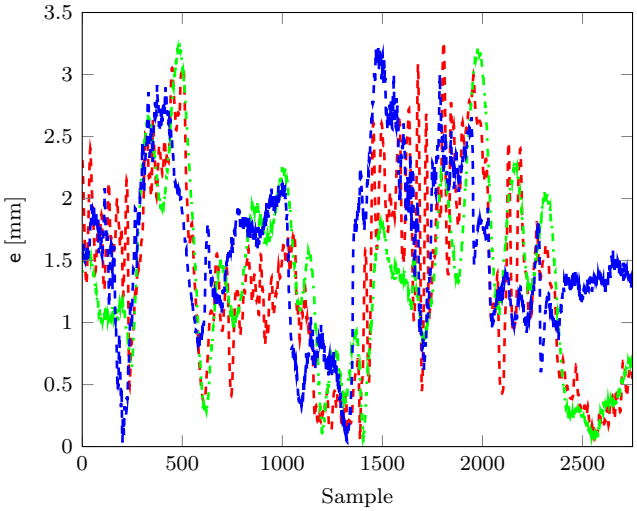

Fig. 6. The path error for the estimated path using OBS1 (red), OBS2 (green), and OBS6 (blue).

Both observers execute in real-time. The execution times are just below one second and around one-fifth of a second, respectively. OBS6 is clearly the fastest one of the six proposed observers. OBS5 is the one that will be compared to the other observers.

Summary

The three observers OBS1, OBS4, and OBS5 are the best ones from each pair. From Table 1 it can be seen that OBS1 and OBS4 have the same performance and that OBS5 is a bit worse, see also the path errors in Figure 6. The differences are small so it is difficult to say which one that is the best. Instead of filter performance, other things have to be considered, such as complexity, computation time, and robustness.

Complexity. The complexity of the filters can be divided into model complexity and implementation complexity. The implementation of OBS1 is straightforward and no particular tuning has to be performed in order to get an estimate. The tuning is of course important to get a good estimate. Instead, most of the time has to be spent

on a rigorous modelling work and identification of the parameters to minimise model errors.

For OBS4 the opposite is true. The model is simple and requires not that much work. Most of the time has to be spent on implementing the PF. The standard choices of a proposal distribution did not work due to high SNR and non-invertible measurement model. Instead, an approximation of the optimal proposal, using an EKF, was required. The consequence is more equations to implement and more tuning knobs to adjust.

The model complexity for OBS5 is in between OBS1 and OBS2. No model for the rigid body motion on the arm side is needed which is a difference from the other two. However, a nonlinear model for the flexibilities and friction is still required, which is not the case for OBS4.

Computation time. The computation time differs a lot for the three observers. OBS5 is in real-time with the current MATLAB implementation and OBS1 can probably be executed in real-time after some optimisation of the MATLAB code or with another programming language. The computation time for OBS4 is in the order of hours and is therefore far from real-time.

Robustness. An advantage with OBS5, compared to the other two, is that the equations describing the arm dynamics are removed, hence no robustness issues concerning the model parameters describing the arm, such as inertia, masses, centre of gravity, etcetera. However, the model parameters describing the flexibilities remains.

Other advantages. An advantage with OBS4 is that the PF provides the entire distribution of the states, which is approximated as a Gaussian distribution in the EKF. The information about the distribution can be used in e.g. control and diagnosis.

6. CONCLUSIONS

A sensor fusion approach to estimate the end-effector position by combining a triaxial accelerometer at the end-effector and the motor angular positions of an industrial robot is presented. The estimation is formulated as a Bayesian estimation problem and has been evaluated on experimental data from a state of the art industrial robot.

Different types of observers where both the estimation model and the filter were changed, have been used. The three observers with the best performance were

a) an EKF using a nonlinear dynamic model,
b) a particle filter using a linear dynamic model, and
c) an EKF with a nonlinear model, where the acceleration of the end-effector is used as an input instead of a measurement.

The performance of these three observers was very similar when considering the path error. The execution time for a) was just above the real-time limit, for c) just below the limit, and for b) in the order of hours. The time required for modelling and implementation is also different for the three different observers. For b), most of the time was spent to implement the filter and get it to work, whereas most of the time for a) was spent on modelling the dynamics.

REFERENCES

Anderson, B.D.O. and Moore, J.B. (1979). *Optimal Filtering*. Information and System Sciences Series. Prentice Hall Inc., Englewood Cliffs, New Jersey, USA.

Axelsson, P. and 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.

Brogårdh, T. (2007). Present and future robot control development—an industrial perspective. *Annual Reviews in Control*, 31(1), 69–79.

Crossbow Technology (2004). Accelerometers, High Sensitivity, LF Series, CXL02LF3. Available at `http://www.xbow.com`.

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

Doucet, A., de Freitas, N., and Gordon, N. (eds.) (2001). *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, New York, USA.

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

Franklin, G.F., Powell, J.D., and Emami-Naeini, A. (2002). *Feedback Control of Dynamic Systems*. Prentice Hall Inc., Upper Saddle River, New Jersey, USA, fourth edition.

Gordon, N.J., Salmond, D.J., and Smith, A.F.M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140(2), 107–113.

Gustafsson, F. (2010). *Statistical Sensor Fusion*. Studentlitteratur, Lund, Sweden, first edition.

Henriksson, R., Norrlöf, M., Moberg, S., Wernholt, E., and Schön, T.B. (2009). Experimental comparison of observers for tool position estimation of industrial robots. In *Proceedings of the 48th IEEE Conference on Decision and Control*, 8065–8070. Shanghai, China.

Jassemi-Zargani, R. and Necsulescu, D. (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*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, New York, USA.

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

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

Leica Geosystems (2008). Case study ABB robotics - Västerås. Available at `http://metrology.leica-geosystems.com/en/index.htm`.

Moberg, S., Öhr, J., and Gunnarsson, S. (2008). A benchmark problem for robust control of a multivariable nonlinear flexible manipulator. In *Proceedings of the 17th IFAC World Congress*, 1206–1211. Seoul, Korea. URL: `http://www.robustcontrol.org`.

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

Wernholt, E. and Moberg, S. (2011). Nonlinear gray-box identification using local models applied to industrial robots. *Automatica*, 47(4), 650–660.

Yu, B.M., Shenoy, K.V., and Sahani, M. (2004). Derivation of extended Kalman filtering and smoothing equations. URL: `http://www-npl.stanford.edu/~byronyu/papers/derive_eks.pdf`.