# Bayesian State Estimation of a Flexible Industrial Robot

Patrik Axelsson, Rickard Karlsson and Mikael Norrlöf

## Contribution

A sensor fusion method for state estimation of a flexible industrial robot is presented. By measuring the acceleration at the end-effector, the accuracy of the arm angular position is improved significantly. The technique is verified on experiments on the ABB IRB4600 robot.

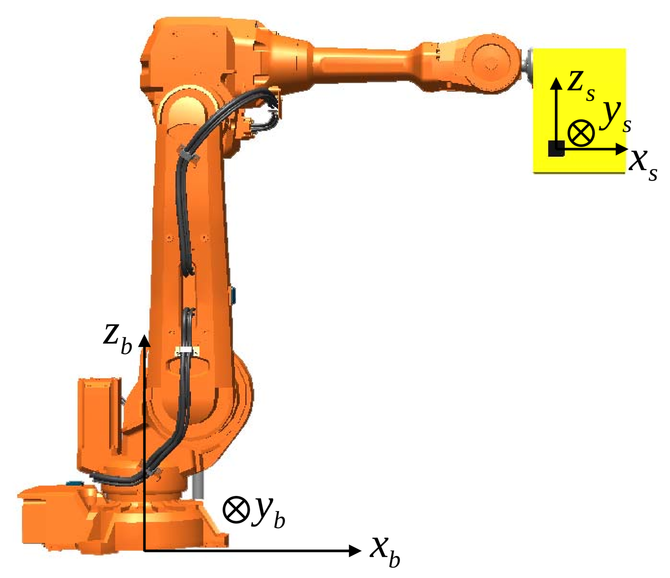
## Introduction

Industrial robot control is based only on measurements from the motor angles. The aim here is to evaluate the

- *extended Kalman filter* (EKF)
- *particle filter* (PF)

for estimation of the end-effector position with

- motor angles, and
- end-effector acceleration,



as measurements using a state space model with *linear dynamic*.

## Bayesian Estimation

Consider the discrete state-space model

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

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 EKF linearises the nonlinear model around the previous estimate giving a time variant linear system where the Kalman filter can be applied.

The PF approximates the density $p(x_t|\mathbb{Y}_t)$ by a large set of $N$ particles $\{x_t^{(i)}\}_{i=1}^N$, where each particle has an assigned relative weight $\gamma_t^{(i)}$, chosen such that all weights sum to unity. The particles and weights are updated with each new observation. The PF is summarised according to

1. Generate $N$ samples $\{\mathbf{x}_0^{(i)}\}_{i=1}^N$ from $p(x_0)$.
2. Compute
$$\gamma_t^{(i)} = \gamma_{t-1}^{(i)} \cdot \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{p^{\text{prop}}(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)}$$
and normalize, i.e., $\bar{\gamma}_t^{(i)} = \gamma_t^{(i)} / \sum_{j=1}^N \gamma_t^{(j)}, \ i = 1, \ldots, N.$
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 $1/N$.
4. Generate predictions from the proposal distribution $\mathbf{x}_{t+1}^{(i)} \sim p^{\text{prop}}(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i\star)}, \mathbf{y}_{t+1}), \ i = 1, \ldots, N.$
5. Increase $t$ and continue to step 2.

## Models

A linear state space model for the dynamics with arm angles, arm velocities and arm accelerations as state variables, together with bias terms compensating for model errors and sensor drift, is propsed. The state vector is given by

$$\mathbf{x}_t = \left(\mathbf{q}_{a,t}^T \ \dot{\mathbf{q}}_{a,t}^T \ \ddot{\mathbf{q}}_{a,t}^T \ \mathbf{b}_{m,t}^T \ \mathbf{b}_{\ddot{\rho},t}^T\right)^T,$$

where $\mathbf{q}_{a,t}$ contains the arm angles from joint 2 and 3, $\dot{\mathbf{q}}_{a,t}$ is the angular velocity, $\ddot{\mathbf{q}}_{a,t}$ is the angular acceleration, $\mathbf{b}_{m,t}$ is the bias terms for the motor angles, and $\mathbf{b}_{\ddot{\rho},t}$ is the bias terms for the acceleration at time $t$. This yields the following state space model in discrete time

$$\mathbf{x}_{t+1} = \mathbf{F}_t\mathbf{x}_t + \mathbf{G}_{u,t}\mathbf{u}_t + \mathbf{G}_{w,t}\mathbf{w}_t,$$
$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{e}_t,$$

where

$$\mathbf{F}_t = \begin{pmatrix} \mathbf{I} & T\mathbf{I} & T^2/2\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & T\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix}, \quad \mathbf{G}_{w,t} = \begin{pmatrix} \frac{T^3}{6}\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \frac{T^2}{2}\mathbf{I} & \mathbf{0} & \mathbf{0} \\ T\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix}, \quad \mathbf{G}_{u,t} = \begin{pmatrix} \frac{T^3}{6}\mathbf{I} \\ \frac{T^2}{2}\mathbf{I} \\ T\mathbf{I} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}.$$

The input, $\mathbf{u}_t$, is the arm jerk reference, i.e., the differentiated arm angular acceleration reference.

The observation relation is given by

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

where

$$\mathbf{q}_{m,t} = r_g^{-1}\Big(\mathbf{q}_{a,t} + k^{-1}\big(M_a(\mathbf{q}_{a,t})\ddot{\mathbf{q}}_{a,t} + g(\mathbf{q}_{a,t}) + C(\mathbf{q}_{a,t}, \dot{\mathbf{q}}_{a,t})\dot{\mathbf{q}}_{a,t}\big)\Big)$$

is the motor angles from the robot dynamic equation and

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

is the acceleration of the end-effector, where $\mathbf{J}(\mathbf{q}_{a,t})$ is the Jacobian.

## Results

The following three estimates are compared to the true position

$$\mathcal{T}_{\text{TCP}}(\hat{\mathbf{q}}_{\text{EKF},t}), \quad \mathcal{T}_{\text{TCP}}(\hat{\mathbf{q}}_{\text{PF},t}), \quad \text{and} \quad \mathcal{T}_{\text{TCP}}(\mathbf{q}_{m,t}),$$

where $\mathcal{T}_{\text{TCP}}(\cdot)$ is the forward kinematic model. The true position is measured with a laser tracking system from Leica Geosytems.

- The EKF and PF track the entire path, i.e., the filters do not diverge.
- The EKF has some problems in the corners, where the PF is much better.
- The PF is closer to the true path.
- Implementing the EKF in MATLAB gives almost a real-time solution, whereas the PF is much slower.