Chapter 4

Sensitivity-Based Adaptive SRUKF for State, Parameter, and Covariance Estimation on Mechatronic Systems

Mauro Hernán Riva, Mark Wielitzka and Tobias Ortmaier

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.72470

Abstract

Since the initial developments in the state-space theory in the 1950s and 1960s, the state estimation has become an extensively researched and applied discipline. All systems that can be modelled mathematically are candidates for state estimators. The state estimators reconstruct the states that represent internal conditions and status of a system at a specific instant of time using a mathematical model and the information received from the system sensors. Moreover, the estimator can be extended for system parameter estimation. The resulting Kalman filter (KF) derivatives for state and parameter estimation also require knowledge about the noise statistics of measurements and the uncertainties of the system model. These are often unknown, and an inaccurate parameterization may lead to decreased filter performance or even divergence. Additionally, insufficient system excitation can cause parameter estimation drifts. In this chapter, a sensitivity-based adaptive square-root unscented KF (SRUKF) is presented. This filter combines a SRUKF and the recursive prediction-error method to estimate system states, parameters and covariances online. Moreover, local sensitivity analysis is performed to prevent parameter estimation drifts, while the system is not sufficiently excited. The filter is evaluated on two testbeds based on an axis serial mechanism and compared with the joint state and parameter UKF.

Keywords: Unscented Kalman, filter, recursive prediction-error method, state estimation, parameter estimation, covariance estimation, sensitivity analysis

1. Introduction

State estimation is applicable to almost all areas of engineering and science. It is interesting to engineers for different reasons such as the control of a system using a state-feedback controller or

monitoring the system states that are not measureable with sensors, or the sensors are too expensive or too difficult to install. The system states can be defined as variables, which provide a representation of internal conditions and status of a system at a specific instant of time. Applications that include a mathematical model of any system are candidates for state estimation. The estimations can be useful, for example, car assistance systems [1], predictive maintenance [2], structure health estimation [3], and many other applications (see [4] and references therein).

Different algorithms were proposed for online state estimation. A historical survey of the filtering algorithms can be found in [5]. The Kalman filter (KF) was presented in [6] and nowadays is the most widely applied algorithm for state estimation on linear systems. The KF is a linear optimal estimator [7]. This means that the KF is the best filter that uses a linear combination of the system measurements and states in order to estimate the last ones. The main operation of the KF is the propagation of the mean and covariance of the (Gaussian) random variables (RVs) through time. The KF assumes that the model and the noise statistics affecting the system are known. Otherwise, the estimates can degrade.

Different derivatives of the KF have been developed for nonlinear systems during the last decades. The extended Kalman filter (EKF) presented in [8] is the most commonly used estimator for nonlinear system. This filter linearizes the system and measurement equations at the current estimate. This may lead to poor performances for highly nonlinear or highly noisy systems [9]. To address the linearization errors of the EKF, the unscented Kalman filter (UKF) was presented in [10]. This filter uses the unscented transformation (UT) to pick a minimal set of points around the mean of the GRV. These points capture the true mean and covariance of the GRV, and they are then propagated through the true nonlinear function capturing the a posteriori mean and covariance more accurately.

The mathematical models usually describe the behaviour of the systems, and generally the structure and the parameters need to be determined. Once the structure is defined, system inputs and measurements can be used to identify the model parameters. This can be performed offline [11, 12]. However, the parameters usually may vary during operations. In order to monitor these variations online, the nonlinear extensions of the KF can be extended for parameter estimation [9].

The KF derivatives can only achieve good performances under a priori assumptions, for example, accurate system models, noise statistics knowledge, and proper initial conditions [7, 9, 13]. If one of these assumptions is not guaranteed, the KF derivative can potentially become unstable and the estimations can be diverged [14–16]. Moreover, tuning the performance of these filters implies primarily adjusting the process and measurement noise covariances to match the (unknown) real-system noise statistics. In the last decades, numerous methods were presented to estimate these unknown covariances. The autocovariance least-square method was presented in [17, 18], and it was extended (and simplified) in [19], and diagonal process and noise covariances were considered in [20]. This method estimates the noise covariances using least squares and it can only be used with KF. The method was extended for nonlinear or time-varying systems using an EKF in [21]. Online covariance estimation for EKF and square-root cubature Kalman filter (SRCuKF) was presented in [22]. These methods implement a combination of a KF

derivative and a recursive prediction-error method (RPEM) to estimate covariances online. In [23], an adaptive UKF was presented to estimate only covariances online.

In this chapter, a sensitivity-based adaptive square-root unscented Kalman filter (SB-aSRUKF) is presented. This filter estimates system states, parameters and covariances online. Using local state sensitivity models (SMs), this filter prevents parameter and covariance estimation drifts, while the system is not sufficiently excited. Sensitivity analysis (SA) for the UKF is also presented. The performance of this filter is validated in simulations on two testbeds and compared with the joint UKF for parameter and state estimation.

Section 2 covers some algorithms for recursive estimation of states, parameters, and covariances. The SB-aSRUKF is the main topic of this chapter. This filter uses a KF derivative for state estimation. In Section 2.1, the KF for state estimation in linear dynamic systems is presented. The UKF, a nonlinear extension of the KF, is described in Section 2.2 and also extended for estimating system parameters. Section 2.3 covers parameter estimation using the RPEM. The UKF and the RPEM are combined in Section 2.4 to obtain the aSRUKF. In order to identify unknown parameters, the system inputs should be persistently exciting. Sensitivity models (SMs) are presented in this section and are used to evaluate the system excitation and prevent parameter estimation drifts while the system is not sufficiently excited.

Section 3 covers the testbed used for the filter evaluations. A planar one-link robot system is described in Section 3.1, and a pendulum robot (pedubot) is mathematically modelled in Section 3.2. The first testbed is used for the SM analysis, and the chaotic system is used to compare the filter performance with the joint SRUKF. The evaluation results of the SB-aSRUKF are presented in Section 4. The SMs are analysed with different system inputs on the first testbed in Section 4.1, and the filter performance for state and parameter estimation is compared with the joint SRUKF in Section 4.2. Section 5 completes the chapter with conclusions.

2. Recursive estimation

This section discusses some recursive approaches to estimate states, parameters and covariances of a general system. The KF as the optimal linear estimator for linear dynamic systems is presented. Nonlinear extensions of the KF are discussed, as well as an extension for parameter estimation. A recursive Gauss-Newton method for parameter estimation is also presented in this section. Finally, the last subsection discusses the SB-aSRUKF, which is the main topic of this chapter, and the SMs that are used for excitation monitoring.

2.1. Kalman filter (KF)

The KF is the most widely applied algorithm for state estimation on linear dynamic systems that are corrupted by stochastic noises (e.g. Gaussian noise). It uses a parametric mathematical model of the system and a series of (noisy) measurements from, for example, sensors to estimate the system states online [4]. In general, the state distribution of a system can be approximated by random variables (RVs). The main operation of the KF is the propagation of

the mean and covariance of these (Gaussian) RVs through time. The KF is an optimal linear filter for these types of systems [7, 9]. It is a recursive algorithm, which enables new measurements to be processed as they arrive to correct and update the state and measurement estimates.

In general, a linear discrete-time system corrupted by additive noises can be written as follows:

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} + \boldsymbol{w}_k, \\ \boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}\boldsymbol{u}_k + \boldsymbol{v}_k, \end{aligned} \tag{1}$$

where $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ is the system state vector at discrete time k, and $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$ and $\boldsymbol{y}_k \in \mathbb{R}^{n_y}$ correspond to the system input and measurement vectors, respectively. The matrices $\boldsymbol{A} \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{B} \in \mathbb{R}^{n_x \times n_u}$, $\boldsymbol{C} \in \mathbb{R}^{n_y \times n_x}$ and $\boldsymbol{D} \in \mathbb{R}^{n_y \times n_u}$ are often called system, input, output and feedforward matrices, respectively, and describe the system behaviour. The random variable vectors $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ represent the process and measurement noises. These are considered white Gaussian, zero mean, and uncorrelated and have covariance matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$, respectively, as

$$\begin{aligned} w_k &\backsim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{Q}_k), \\ v_k &\backsim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{R}_k). \end{aligned} \tag{2}$$

The KF iterative nature can be separated in two main steps: the process update and the correction step. In the process update, based on the knowledge of the system dynamics, the state estimate ($\widehat{\boldsymbol{x}}_{k-1}^{+}$)[1] from the previous time step ($k-1$) is used to calculate a new estimate at the current time (k). This step does not include any information of the system measurements and the resulting state estimate is called a priori estimate ($\widehat{\boldsymbol{x}}_k^{-}$). In the correction step, the a priori estimate is combined with the current system measurement ($\boldsymbol{y}_k$) to improve the state estimate. This estimate is called the a posteriori state estimate ($\widehat{\boldsymbol{x}}_k^{+}$). The vectors $\widehat{\boldsymbol{x}}_k^{-}$ and $\widehat{\boldsymbol{x}}_k^{+}$ estimate both the same quantity, but the difference between them is that the last one takes the measurement ($\boldsymbol{y}_k$) into account. A Kalman gain matrix ($\boldsymbol{K}_k$) is calculated at every discrete step and weights the influence of the model and the measurements on the current state estimate. This gain is calculated using the system matrices and the process ($\boldsymbol{Q}_k$) and measurement ($\boldsymbol{R}_k$) covariances. More information about the KF equations and generalizations can be found in [4, 7, 9].

The KF is a linear optimal estimator, but it assumes that the system model and noise statistics are known. Otherwise, the filter estimates can degrade. Tuning the performance of the filter implies primarily adjusting the process and measurement covariance matrices to match the (unknown) real-system noise statistics. In practical implementations of the KF, the filter tuning is performed online, and empirical values are normally used. Extensive research has been done in this field to estimate the noise covariances from data (see [17–20] and references therein).

As mentioned before, the KF is the optimal linear estimator, which estimates states of a linear dynamic system using the inputs, measurements and a parametric mathematical model of the system. Even though many systems are close enough to linear and linear estimators give

[1] The hat ˆ over a vector represents the estimate of the vector, for example, $\widehat{\boldsymbol{x}}$ describes the estimate of the state vector $\boldsymbol{x}$.

acceptable results, all systems are ultimately nonlinear. Extensions of the KF have been presented in the last decades to deal with nonlinear systems. Some examples are the EKF and the sigma-point Kalman filters (SPKFs).

2.2. Nonlinear filtering

The EKF and the UKF (a SPKF type) are derivatives of the KF for nonlinear systems. The EKF was originally proposed in [8] and is the most commonly applied state estimator for nonlinear systems. However, if the system nonlinearities are severe or the noises affecting the system are high, the EKF can be difficult to tune, often gives wrong estimates and can lead to filter divergence easily. This is because the EKF uses linearized system and measurement models at the current estimate and propagates the mean and covariance of the GRVs through these linearizations. The UKF was presented in [10] and addresses the deficiencies of the EKF linearization providing a direct and explicit mechanism for approximating and transforming the mean and covariance of the GRVs.

In general, a discrete-time state-space model of a nonlinear system can be described by

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{\theta}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{w}_{k-1}, \\ \boldsymbol{y}_k &= \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{\theta}_k, \boldsymbol{u}_k) + \boldsymbol{v}_k, \end{aligned} \tag{3}$$

where $\boldsymbol{\theta}_k \in \mathbb{R}^{n_p}$ is the (unknown) parameter vector and $\boldsymbol{f}$ and $\boldsymbol{h}$ are arbitrary vector-valued functions usually called system and measurement functions. As a KF derivative, the UKF aim is to minimize the covariance of the state estimation error to find an optimal estimation of the state true dynamic probability density function (pdf). The main component of this filter is the UT. This transformation uses a set of appropriately selected weighted points to parameterize the mean and covariance of the pdf. Two steps characterize also the UKF. In the process update, the sigma points are calculated and then propagated through the nonlinear system functions to recover the mean and covariance of the new a priori estimates. The estimated measurement ($\widehat{\boldsymbol{y}}_k$) is calculated in the correction step and together with the actual measurement are used to correct the a priori estimate. This results in the a posteriori state estimate. While the UKF matches the true mean of $\boldsymbol{x}_k$ correctly up to the third order, the EKF only matches up to the first order. Both filters approximate the true covariance of $\boldsymbol{x}_k$ up to the third order. However, the UKF correctly approximates the signed of the terms to the fourth power and higher meaning that the resulting error should be smaller [7, 9].

The nonlinear extensions of the KF can also estimate the unknown parameters of a system. The UKF was extended for joint state and parameter estimation in [24]. In this case, the system state vector $\boldsymbol{x}_k$ was extended by including the unknown parameters $\boldsymbol{\theta}_k$ to obtain a joint state and parameter vector as

$$\tilde{\boldsymbol{x}}_k = \begin{pmatrix} \boldsymbol{x}_k \\ \boldsymbol{\theta}_k \end{pmatrix}, \tag{4}$$

remaining $\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1}$ during the process update.

Square-root (SR) filtering increases mathematically the precision of the KF when hardware precision is not available. In [25], an SR version of the UKF was presented, which uses linear

algebra techniques such as the QR decomposition and the Cholesky factor [26] to calculate the SR of the estimation error covariance. The SR form improves the numerical stability of the filter and guarantees positive semi-definiteness of this covariance. Additionally, the computational complexity for state and parameter estimation is reduced [25].

2.3. Recursive prediction-error method

In this section, the recursive prediction-error method (RPEM) is briefly discussed. This method is extensively analysed in [11, 12] and uses a parameterized predictor that estimates the system outputs at the current time step. The resulting predicted system output is then compared to the actual system measurement, and the predictor parameters are corrected such as that the prediction error is minimized.

The quadratic criterion function defined as

$$V_k(\boldsymbol{\theta}_k) = \frac{1}{2}\boldsymbol{e}_k^{\mathrm{T}}(\boldsymbol{\theta}_k)\boldsymbol{\Lambda}^{-1}\boldsymbol{e}_k(\boldsymbol{\theta}_k), \tag{5}$$

is minimized using the stochastic Gauss-Newton method in order to obtain the predictor parameters. The prediction error $\boldsymbol{e}_k(\boldsymbol{\theta}_k)$ at the discrete time k is described as

$$\boldsymbol{e}_k(\boldsymbol{\theta}_k) = \boldsymbol{y}_k - \widehat{\boldsymbol{y}}_k(\boldsymbol{\theta}_k), \tag{6}$$

where $\boldsymbol{y}_k$ corresponds to the actual system measurement, $\widehat{\boldsymbol{y}}_k(\boldsymbol{\theta}_k)$ refers to the parameterized predictor output using parameter set $\boldsymbol{\theta}_k$ and Λ is a user-defined weight factor.

The recursive solution that minimizes the quadratic criterion function in Eq. (5) is given by the following scheme:

$$\begin{aligned}
\boldsymbol{\Delta}_k &= \boldsymbol{\Delta}_{k-1} + (1-\lambda)\left(\boldsymbol{e}_k\boldsymbol{e}_k^{\mathrm{T}} - \boldsymbol{\Delta}_{k-1}\right),\\
\boldsymbol{S}_k &= \lambda\boldsymbol{\Delta}_k + \frac{\mathrm{d}\widehat{\boldsymbol{y}}_k}{\mathrm{d}\widehat{\boldsymbol{\theta}}_{k-1}}\boldsymbol{\Theta}_{k-1}\frac{\mathrm{d}\widehat{\boldsymbol{y}}_k^{\mathrm{T}}}{\mathrm{d}\widehat{\boldsymbol{\theta}}_{k-1}},\\
\boldsymbol{L}_k &= \boldsymbol{\Theta}_{k-1}\frac{\mathrm{d}\widehat{\boldsymbol{y}}_k^{\mathrm{T}}}{\mathrm{d}\widehat{\boldsymbol{\theta}}_{k-1}}\boldsymbol{S}_k^{-1},\\
\boldsymbol{\Theta}_k &= \left(\boldsymbol{I}^{n_\theta} - \boldsymbol{L}_k\frac{\mathrm{d}\widehat{\boldsymbol{y}}_k}{\mathrm{d}\widehat{\boldsymbol{\theta}}_{k-1}}\right)\boldsymbol{\Theta}_{k-1}\left(\boldsymbol{I}^{n_\theta} - \boldsymbol{L}_k\frac{\mathrm{d}\widehat{\boldsymbol{y}}_k^{\mathrm{T}}}{\mathrm{d}\widehat{\boldsymbol{\theta}}_{k-1}}\right)/\lambda + \boldsymbol{L}_k\boldsymbol{\Delta}_k\boldsymbol{L}_k^{\mathrm{T}},\\
\widehat{\boldsymbol{\theta}}_k &= \widehat{\boldsymbol{\theta}}_{k-1} + \boldsymbol{L}_k\boldsymbol{e}_k.
\end{aligned} \tag{7}$$

The user-defined parameter $0 < \lambda \leq 1$ is often called the forgetting factor. The matrix $\boldsymbol{\Delta}_k$ is calculated using the prediction error. This matrix is used to calculate $\boldsymbol{S}_k$, where the derivative of the output w.r.t. to the unknown parameter vector $\left(\frac{\mathrm{d}\widehat{\boldsymbol{y}}_k}{\mathrm{d}\widehat{\boldsymbol{\theta}}_{k-1}}\right)$ appears. The gain vector $\boldsymbol{L}_k$ is multiplied by the innovation error to update then the parameter estimation. It should be noted that besides the matrix $\widehat{\boldsymbol{y}}_k = \frac{\mathrm{d}\widehat{\boldsymbol{y}}_k}{\mathrm{d}\widehat{\boldsymbol{\theta}}_{k-1}}$, all parameters, vectors, and matrices are defined after an initialization. The matrix $\widehat{\boldsymbol{y}}_k$ can be calculated modifying a KF derivative.

http://dx.doi.org/10.5772/intechopen.72470

The selection of the forgetting factor essentially defines the measurements that are relevant for the current estimation of the parameter predictor. The most common choice is to take a constant forgetting factor for systems that change gradually. Other criterions for selection of this factor and the convergence of the RPEM are discussed extensively in [11, 12].

2.4. Sensitivity-based adaptive square-root Kalman filter

This is the main section of this chapter. The earlier sections were written to provide the needed methods for this section, and the later sections are written to analyse and test the performance of the filter described in this section.

The aSRUKF is discussed in this section. This filter combines the SRUKF and the RPEM. While the KF derivative estimates the system states and measurements, the RPEM calculates the unknown parameters and covariances.

In this filter, the innovation error in Eq. (6) is calculated and minimized using the recursive scheme presented in Eq. (7) in order to estimate the unknown system parameters and covariances. Besides the matrix $\widehat{\boldsymbol{y}}_k$, all parameters, vectors, and matrices of the recursive scheme are defined. The derivative of the estimated measurement $(\widehat{\boldsymbol{y}}_k)$ w.r.t. the vector $(\widehat{\boldsymbol{\theta}}_{k-1})$ containing the unknown values of parameters and covariances needs to be calculated. This matrix is also called the output sensitivity and describes the influence of a variation of a parameter on the system output. The output sensitivity can be obtained using a KF derivative.

The equations of a SRUKF are then extended in order to calculate the output sensitivity. To simplify the reading flow, the following definitions are presented:

$$
\begin{aligned}
&w_i^m = w_i^c = \frac{1}{2\left(n_x + \lambda_f\right)}, \quad i = 1, \ldots, m = 2n_x, \quad \lambda_f = \alpha^2(n_x + \kappa), \\
&w_0^m = \frac{\lambda_f}{2\left(n_x + \lambda_f\right)}, \qquad w_0^c = w_0^m + \left(1 - \alpha^2 + \beta\right), \qquad \eta = \sqrt{n_x + \lambda_f}, \\
&\boldsymbol{m}_1 = 1_{1,\, n_x}, \qquad \boldsymbol{m}_2 = 1_{1,\, l}, \qquad l = 2n_x + 1, \\
&\boldsymbol{w}^c = \left(w_0^c, \ldots, w_m^c\right), \quad \boldsymbol{w}^m = \left(w_0^m, \ldots, w_m^m\right), \qquad \boldsymbol{W}_{\mathrm{kr}}^c = \boldsymbol{w}^c \otimes \boldsymbol{m}_1^{\mathrm{T}},
\end{aligned}
\tag{8}
$$

where $w_i^{m,\,c}$ are a set of scalar weights, α determines the spread of sigma points around the estimated state $\widehat{\boldsymbol{x}}_k$, β incorporates information about the noise distribution (e.g. $\beta = 2$ assumes that the system is affected by Gaussian noise), and κ is a scaling factor, which can be used to reduce the higher-order errors of the mean and covariance approximations [9]. The Kronecker product [27] is described by $\otimes$.

The process update step of the aSRUKF is presented in **Table 1**. After the filter initialization, the sigma points $(\boldsymbol{X}_{k-1})$ that describe the pdf of the state estimate are calculated using the UT. At the same time, the sigma-point derivatives $(\boldsymbol{\Phi}_{k-1})$ are also obtained. The sigma points and its derivatives are propagated through the system function and the system derivative function, respectively, to obtain the a priori state estimate $(\widehat{\boldsymbol{x}}_k^-)$ and the a priori state estimate sensitivity $(\widehat{\boldsymbol{\mathcal{X}}}_k^-)$. Considering additive process noises, the SR factor of the estimation error covariance $(\boldsymbol{S}_{\boldsymbol{xx},\,k}^-)$ is calculated using the QR decomposition (qr()) and the rank-one update to Cholesky

aSRUKF	Initialization	$\widehat{\boldsymbol{x}}_0 = \boldsymbol{x}_{\text{init}} \in \mathbb{R}^{n_x}, \quad \widehat{\boldsymbol{\theta}}_0 = \boldsymbol{\theta}_{\text{init}} \in \mathbb{R}^{n_\theta}, \quad \boldsymbol{\mathcal{S}}_{yy,k} = \frac{\mathrm{d}\boldsymbol{S}_{yy,k}}{\mathrm{d}\boldsymbol{\theta}_k}, \quad \boldsymbol{\Theta}_k = \boldsymbol{\Theta}_{\text{init}} \in \mathbb{R}^{n_\theta \times n_\theta}$ $\boldsymbol{\mathcal{P}}_{xx,k} = \frac{\mathrm{d}\boldsymbol{P}_{xx,k}}{\mathrm{d}\boldsymbol{\theta}_k}, \quad \boldsymbol{\mathcal{P}}_{yy,k} = \frac{\mathrm{d}\boldsymbol{P}_{yy,k}}{\mathrm{d}\boldsymbol{\theta}_k}, \quad \boldsymbol{\mathcal{S}}_{xx,k} = \frac{\mathrm{d}\boldsymbol{S}_{xx,k}}{\mathrm{d}\boldsymbol{\theta}_k}, \quad \boldsymbol{\Delta}_k = \mathbf{0} \in \mathbb{R}^{n_y \times n_y},$ $\boldsymbol{P}_{xx,0} = \boldsymbol{S}_{xx,0}\boldsymbol{S}_{xx,0}^{\mathrm{T}} = \boldsymbol{P}_{xx,\text{init}} \in \mathbb{R}^{n_x \times n_x},$ $\boldsymbol{\mathcal{P}}_{xx,0} = \boldsymbol{\mathcal{S}}_{xx,0} = \mathbf{0} \in \mathbb{R}^{n_x \times n_x n_\theta}, \quad \boldsymbol{\mathcal{P}}_{yy,0} = \boldsymbol{\mathcal{S}}_{yy,0} = \mathbf{0} \in \mathbb{R}^{n_y \times n_y n_\theta}$
SRUKF	Sigma points	$\boldsymbol{X}_{k-1} = (\boldsymbol{X}_{k-1,1}, \ldots, \boldsymbol{X}_{k-1,l}) = \left(\widehat{\boldsymbol{x}}_{k-1} \widehat{\boldsymbol{x}}_{k-1} \otimes \boldsymbol{m}_1 + \eta \boldsymbol{S}_{xx,k-1} \widehat{\boldsymbol{x}}_{k-1} \otimes \boldsymbol{m}_1 - \eta \boldsymbol{S}_{xx,k-1}\right)$
	Sigma-points propagation	$\boldsymbol{X}_k^* = f\left(\boldsymbol{X}_{k-1}, \widehat{\boldsymbol{\theta}}_{k-1}, \boldsymbol{u}_{k-1}\right)$
SM	Sigma-point derivatives	$\boldsymbol{\Phi}_{k-1,j} = \left(\boldsymbol{\Phi}_{k-1,1,j}, \ldots, \boldsymbol{\Phi}_{k-1,l,j}\right) = \left(\frac{\mathrm{d}\boldsymbol{X}_{k,1}}{\mathrm{d}\widehat{\boldsymbol{\theta}}_{k,j}}, \ldots, \frac{\mathrm{d}\boldsymbol{X}_{k,m}}{\mathrm{d}\widehat{\boldsymbol{\theta}}_{k,j}}\right) = \left(\widehat{\boldsymbol{\mathcal{X}}}_{k-1,j} \widehat{\boldsymbol{\mathcal{X}}}_{k-1,j} \otimes \boldsymbol{m}_1 + \eta \boldsymbol{\mathcal{S}}_{xx,k-1,j} \widehat{\boldsymbol{\mathcal{X}}}_{k-1,j} \otimes \boldsymbol{m}_1 - \eta \boldsymbol{\mathcal{S}}_{xx,k-1,j}\right)$ with $\widehat{\boldsymbol{\mathcal{X}}}_{k-1} = \frac{\mathrm{d}\widehat{\boldsymbol{x}}_{k-1}}{\mathrm{d}\widehat{\boldsymbol{\theta}}_{k-1}}$
	Sensitivity propagation	$\boldsymbol{\Phi}_{k,j,i}^* = \frac{\partial f}{\partial \boldsymbol{x}_k}\Big\vert_{\boldsymbol{X}_{k-1}, \hat{\boldsymbol{\theta}}_{k-1}} \boldsymbol{\Phi}_{k-1,j,i} + \frac{\partial f}{\partial \widehat{\boldsymbol{\theta}}_{k,j}}\Big\vert_{\boldsymbol{X}_{k-1}, \hat{\boldsymbol{\theta}}_{k-1}}$
SRUKF	A priori state estimate	$\widehat{\boldsymbol{x}}_k^- = \boldsymbol{X}_k^* (\boldsymbol{w}^m)^{\mathrm{T}}$
	SR estimation error covariance	$\boldsymbol{S}_{xx,k}^- = \mathrm{qr}\left(\left(\sqrt{w_1^c}\left(\boldsymbol{X}_{1:2n_x,k}^* - \widehat{\boldsymbol{x}}_k^-\right) \quad \boldsymbol{S}_{Q,k}\right)\right),$ $\boldsymbol{S}_{xx,k}^- = \mathrm{cholupdate}\left(\boldsymbol{S}_{xx,k}^-, \sqrt{\lvert w_0^c \rvert}\left(\boldsymbol{X}_{0,k}^* - \widehat{\boldsymbol{x}}_k^-\right), \mathrm{sign}(w_0^c)\right)$
SM	A priori state sensitivity	$\widehat{\boldsymbol{\mathcal{X}}}_k^- = \boldsymbol{\Phi}_k^*\left(\boldsymbol{I}^{n_\theta} \otimes (\boldsymbol{w}^m)^{\mathrm{T}}\right)$
	Derivative of estimation error covariance	$\boldsymbol{\mathcal{S}}_{xx,k}^- = \mathrm{treshapeM}\left(\left(\boldsymbol{I}^{n_\theta} \otimes \boldsymbol{A}_{\mathrm{ls},\boldsymbol{S}_{xx,k}^-}^{\dagger}\right)\left(\boldsymbol{\mathcal{P}}_{xx,k}^-\right)_s\right),$ with $\boldsymbol{A}_{\mathrm{ls},\boldsymbol{S}_{xx,k}^-} = \left(\boldsymbol{S}_{xx,k}^- \otimes \boldsymbol{I}^{n_x} + \left(\boldsymbol{I}^{n_x} \otimes \boldsymbol{\mathcal{S}}_{xx,k}^-\right)\boldsymbol{\mathcal{N}}_{(n_x)}\right)\boldsymbol{\mathcal{E}}_{(n_x)}$ and $\boldsymbol{\mathcal{P}}_{xx,k}^- = \left(\boldsymbol{\Phi}_k^* - \widehat{\boldsymbol{\mathcal{X}}}_k^- \otimes \boldsymbol{m}_2\right)\left(\boldsymbol{I}^{n_\theta} \otimes \left(\left(\boldsymbol{W}_{\mathrm{kr}}^c\right)^{\mathrm{T}} \odot \left(\boldsymbol{X}_k^* - \widehat{\boldsymbol{x}}_k^- \otimes \boldsymbol{m}_2\right)^{\mathrm{T}}\right)\right)$ $+ \boldsymbol{W}_{\mathrm{kr}}^c \odot \left(\boldsymbol{X}_k^* - \widehat{\boldsymbol{x}}_k^- \otimes \boldsymbol{m}_2\right)\left(\boldsymbol{\mathcal{T}}_{(l)} \boldsymbol{I}^{n_\theta} \otimes \left(\boldsymbol{\Phi}_k^* - \widehat{\boldsymbol{\mathcal{X}}}_k^- \otimes \boldsymbol{m}_2\right)^{\mathrm{T}}\right) + \frac{\mathrm{d}\boldsymbol{Q}_k}{\mathrm{d}\boldsymbol{\theta}_k}\Big\vert \widehat{\boldsymbol{\theta}}_{k-1}$ in which $\boldsymbol{\mathcal{T}}_{(l)} = \left(\boldsymbol{I}^0, \boldsymbol{T}_1, \ldots, \boldsymbol{T}_{n_\theta}\right)$, and $\boldsymbol{T} = \left(\mathbf{0}_1^{l \times l}, \ldots, \mathbf{0}_{n_\theta}^{l \times l}, \boldsymbol{I}^l\right)$. Further details of the construction of the matrix $\boldsymbol{\mathcal{N}}_{(n_x)}$ and the elimination matrix $\boldsymbol{\mathcal{E}}_{(n_x)}$ can be found in [28]

Table 1. aSRUKF: filter initialization, sigma-points calculation and filter process update step. The Kronecker product is described by $\otimes$ and $\odot$ defines the element-wise multiplication. The $(\,)_s$ operator stacks the matrix columns to form a column vector.

factorization (cholupdate() [2]) in which the signum function (sign()) is used to determine the sign of the first calculated weight factor. If the weight factor results negative, a Cholesky downdate takes place; otherwise, a Cholesky update occurs. The next step calculates the derivative of the SR factor of the estimation error covariance ($\boldsymbol{\mathcal{S}}_{xx,k}^-$). In this step, the function treshapeM() is used. This function converts a vector of dimension $((n_x(n_x+1)/2)n_\theta \times 1)$ into a $(n_x \times n_x n_\theta)$ matrix with n_θ lower triangular blocks of size $(n_x \times n_x)$. Additionally, the operator $()_s$ is utilized to stack the matrix columns to form a column vector. Further information about this step can be found in [28].

[2] Matlab does not allow the use of cholupdate() in real-time applications; using coder.extrinsic('cholupdate'), it is possible to use the function in Simulink but the application does not run in real time. The cholupdate() line can be replaced with $\mathrm{chol}\left(\left(\boldsymbol{S}_{xx,k}^-\right)^{\mathrm{T}} \boldsymbol{S}_{xx,k}^- + w_0^c\left(\boldsymbol{X}_{0,k}^* - \widehat{\boldsymbol{x}}_k^-\right)\right)$.

http://dx.doi.org/10.5772/intechopen.72470

The correction step is shown in **Table 2**. A new set of sigma points ($\boldsymbol{X}_k^-$) and its derivatives ($\boldsymbol{\Phi}_k$) can be generated using steps (a) and (b) from **Table 2**. If the nonlinearity is not severe, these steps can be skipped. This saves computational effort but sacrifices filter performance. The (new) sigma points and its derivatives are then propagated through the measurement function and its derivative, respectively. The resulting points are used to calculate the estimated measurements ($\boldsymbol{Y}_k$) as well as the output sensitivities ($\widehat{\boldsymbol{y}}_k$). These are used within the RPEM to estimate the system parameters and covariances.

The SR factor of the innovation error covariance ($\boldsymbol{S}_{yy,k}$), the cross-covariance ($\boldsymbol{P}_{xy,k}$) together with its derivatives matrices ($\boldsymbol{\mathcal{S}}_{yy,k}$, $\boldsymbol{\mathcal{P}}_{xy,k}$) are calculated in order to obtain the Kalman gain matrix ($\boldsymbol{K}_k$) and its sensitivity ($\boldsymbol{\mathcal{K}}_k$). The aSRUKF treats also the measurement noises as additive. The a posteriori state estimation ($\widehat{x}_k^+$), the a posteriori state sensitivity ($\widehat{\boldsymbol{\mathcal{X}}}_k^+$) together with the SR factor of the estimation error covariance ($\boldsymbol{S}_{xx,k}$) and its sensitivity ($\boldsymbol{\mathcal{S}}_{xx,k}$) close the loop of the aSRUKF.

Local sensitivity analysis can be utilized to determine if a system input or a system modification can excite the system parameters in order to identify them. The a posteriori state sensitivity from **Table 2 (d)** can be used to determine the influence of parameters to the system states. This sensitivity results from the correction step of the aSRUKF. As long as the sensitivity $\widehat{\boldsymbol{\mathcal{X}}}_k^+$ remains below a user-defined threshold, the parameter update from **Table 2 (e)** can be skipped to prevent parameter estimation drifts. A time window (N_w) can be used to calculate $\max\left(\left\|\widehat{\boldsymbol{\mathcal{X}}}_{k-N_w}^+\right\|_2, \ldots, \left\|\widehat{\boldsymbol{\mathcal{X}}}_k^+\right\|_2\right)$ to normalize the sensitivity values. A threshold vector $\boldsymbol{\tau_v}$ is then defined with values between 0 and 1. The update procedure can be described as

$$
\begin{aligned}
&\textbf{for } p = 1 \textbf{ to } n_\theta \textbf{ do}\\
&\quad sa = 0\\
&\quad \textbf{for } n = 1 \textbf{ to } n_x \textbf{ do}\\
&\qquad \mathrm{sa} = \mathrm{sa} + \frac{dx_{k,n}}{d\theta_{k,p}}\\
&\textbf{if } \mathrm{sa} > \tau_v(p) \textbf{ then}\\
&\mathrm{update\_values} = \mathrm{True}
\end{aligned}
\tag{9}
$$

The variable *sa* represents the sensitivity sum w.r.t. a system parameter $\theta_{k,p}$ over all system states ($x_{k,1}, \ldots, x_{k,n_x}$). The threshold vector $\boldsymbol{\tau_v}$ should be selected with caution. Too high values prevent parameter estimation drifts but can increase the convergence time of the filter. Moreover, the parameter excitation can be significantly reduced and the resulting estimation can be biased. The performance of the SB-aSRUKF is evaluated in Section 4.

The local state sensitivity can be also calculated as follows (cf. [29]):

$$
\frac{\mathrm{d}\boldsymbol{x}_k}{\mathrm{d}\theta_{k,j}} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}_k}\right|_{\widehat{\boldsymbol{x}}_{k-1}, \widehat{\boldsymbol{\theta}}_{k-1}} \frac{\mathrm{d}\boldsymbol{x}_{k-1}}{\mathrm{d}\theta_{k,j}} + \left.\frac{\partial \boldsymbol{f}}{\partial \theta_{k,j}}\right|_{\widehat{\boldsymbol{x}}_{k-1}, \widehat{\boldsymbol{\theta}}_{k-1}} \tag{10}
$$

This sensitivity computation is compared in Section 4 with a posterior state sensitivity obtained using the SB-aSRUKF.

SRUKF	Sigma points	$\boldsymbol{X}_k^- = (\boldsymbol{X}_{k,1}, \ldots, \boldsymbol{X}_{k,l}) = \left(\widehat{\boldsymbol{x}}_k^- \widehat{\boldsymbol{x}}_k^- \otimes \boldsymbol{m}_1 + \eta \boldsymbol{S}_{\boldsymbol{xx},k}^- \widehat{\boldsymbol{x}}_k^- \otimes \boldsymbol{m}_1 - \eta \boldsymbol{S}_{\boldsymbol{xx},k-1}^- \right)$	(a)
	Output sigma points	$\boldsymbol{Y}_k = \boldsymbol{h}\left(\boldsymbol{X}_k^-, \widehat{\boldsymbol{\theta}}_{k-1}, \boldsymbol{u}_k \right)$	
	Estimated measurement	$\widehat{\boldsymbol{y}}_k = \boldsymbol{Y}_k (\boldsymbol{w}^m)^{\mathrm{T}}$	
SM	Sigma-points derivative	$\boldsymbol{\Phi}_{k,j} = \left(\boldsymbol{\Phi}_{k,1,j}, \ldots, \boldsymbol{\Phi}_{k,l,j} \right) == \left(\widehat{\boldsymbol{\mathcal{X}}}_{k,j} \widehat{\boldsymbol{\mathcal{X}}}_{k,j} \otimes \boldsymbol{m}_1 + \eta \boldsymbol{\mathcal{S}}_{\boldsymbol{xx},k,j}^- \widehat{\boldsymbol{\mathcal{X}}}_{k,j} \otimes \boldsymbol{m}_1 - \eta \boldsymbol{\mathcal{S}}_{\boldsymbol{xx},k,j}^- \right)$	(b)
	Output sigma-points derivative	$\boldsymbol{\Psi}_{k,j,i} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_k}\Big|_{\boldsymbol{X}_k^-, \widehat{\boldsymbol{\theta}}_{k-1}} \boldsymbol{\Phi}_{k,j,i} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}_{k,j}}\Big|_{\boldsymbol{X}_k^-, \widehat{\boldsymbol{\theta}}_{k-1}}$	
	Output sensitivity	$\widehat{\boldsymbol{y}}_k = \boldsymbol{\Psi}_k \left(\boldsymbol{I}^{n_\theta} \otimes (\boldsymbol{w}^m)^{\mathrm{T}} \right)$ with $\boldsymbol{\Psi}_k = (\boldsymbol{\Psi}_{k,1}, \ldots, \boldsymbol{\Psi}_{k,n_\theta})$	(c)
SRUKF	SR innovation error covariance	$\boldsymbol{S}_{\boldsymbol{yy},k} = \mathrm{qr}\left(\left(\sqrt{w_1^c} \left(\boldsymbol{Y}_{1:2n_x,k} - \widehat{\boldsymbol{y}}_k \right) \quad \boldsymbol{S}_{\boldsymbol{R},k} \right) \right)$, $\boldsymbol{S}_{\boldsymbol{yy},k} = \mathrm{cholupdate}\left(\boldsymbol{S}_{\boldsymbol{yy},k}, \sqrt{\left| w_0^c \right|} \left(\boldsymbol{Y}_{0,k} - \widehat{\boldsymbol{y}}_k \right), \mathrm{sign}\left(w_0^c \right) \right)$	
	Cross-covariance matrix	$\boldsymbol{P}_{\boldsymbol{xy},k} = \boldsymbol{W}_{\mathrm{kr}}^c \odot \left(\left(\boldsymbol{Y}_k - \widehat{\boldsymbol{y}}_k \otimes \boldsymbol{m}_2 \right) \left(\boldsymbol{X}_k - \widehat{\boldsymbol{x}}_k^- \otimes \boldsymbol{m}_2 \right)^{\mathrm{T}} \right)$	
SM	Derivative of innovation error covariance	$\boldsymbol{\mathcal{S}}_{\boldsymbol{yy},k} = \mathrm{treshapeM}\left(\left(\boldsymbol{I}^{n_\theta} \otimes \boldsymbol{A}_{\mathrm{ls},\boldsymbol{S}_{\boldsymbol{yy},k}}^{\dagger} \right) \left(\boldsymbol{\mathcal{P}}_{\boldsymbol{yy},k} \right)_s \right)$, with $\boldsymbol{A}_{\mathrm{ls},\boldsymbol{S}_{\boldsymbol{yy},k}} = \left(\boldsymbol{S}_{\boldsymbol{yy},k} \otimes \boldsymbol{I}^{n_y} + \left(\boldsymbol{I}^{n_y} \otimes \boldsymbol{\mathcal{S}}_{\boldsymbol{yy},k} \right) \boldsymbol{\mathcal{N}}_{(n_y)} \right) \boldsymbol{\mathcal{E}}_{(n_y)}$ and $\boldsymbol{\mathcal{P}}_{\boldsymbol{yy},k} = \left(\boldsymbol{\Psi}_k - \widehat{\boldsymbol{y}}_k \otimes \boldsymbol{m}_2 \right) \left(\boldsymbol{I}^{n_\theta} \otimes \left(\left(\boldsymbol{W}_{\mathrm{kr}}^c \right)^{\mathrm{T}} \odot \left(\boldsymbol{Y}_k - \widehat{\boldsymbol{y}}_k \otimes \boldsymbol{m}_2 \right)^{\mathrm{T}} \right) \right) + \boldsymbol{W}_{\mathrm{kr}}^c \odot \left(\boldsymbol{Y}_k - \widehat{\boldsymbol{y}}_k \otimes \boldsymbol{m}_2 \right) \left(\boldsymbol{\mathcal{T}}_{(l)} \boldsymbol{I}^{n_\theta} \otimes \left(\boldsymbol{\Psi}_k - \widehat{\boldsymbol{y}}_k \otimes \boldsymbol{m}_2 \right)^{\mathrm{T}} \right)$	
	Derivative of Cross-covariance matrix	$\boldsymbol{\mathcal{P}}_{\boldsymbol{xy},k} = \left(\boldsymbol{\Phi}_k - \widehat{\boldsymbol{\mathcal{X}}}_k^- \otimes \boldsymbol{m}_2 \right) \left(\boldsymbol{I}^{n_\theta} \otimes \left(\left(\boldsymbol{W}_{\mathrm{kr}}^c \right)^{\mathrm{T}} \odot \left(\boldsymbol{Y}_k - \widehat{\boldsymbol{y}}_k \otimes \boldsymbol{m}_2 \right)^{\mathrm{T}} \right) \right) + \boldsymbol{W}_{\mathrm{kr}}^c \odot \left(\boldsymbol{X}_k - \widehat{\boldsymbol{x}}_k^- \otimes \boldsymbol{m}_2 \right) \left(\boldsymbol{\mathcal{T}}_{(l)} \boldsymbol{I}^{n_\theta} \otimes \left(\boldsymbol{\Psi}_k - \widehat{\boldsymbol{y}}_k \otimes \boldsymbol{m}_2 \right)^{\mathrm{T}} \right)$	
SRUKF	Kalman gain	$\boldsymbol{K}_k = \left(\boldsymbol{P}_{\boldsymbol{xy},k} / \boldsymbol{S}_{\boldsymbol{yy}}^{\mathrm{T}} \right) / \boldsymbol{S}_{\boldsymbol{yy},k}$	
	A posteriori state estimate	$\widehat{\boldsymbol{x}}_k^+ = \widehat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k \left(\boldsymbol{y}_k - \widehat{\boldsymbol{y}}_k \right)$	
	SR estimation error covariance	**for** z = 1 **to** $\mathrm{n_y}$ **do** $\boldsymbol{S}_{\boldsymbol{xx},k} = \mathrm{cholupdate}\left(\boldsymbol{S}_{\boldsymbol{xx},k}^-, \boldsymbol{U}_k(:,z), '-' \right)$ with $\boldsymbol{U}_k = \boldsymbol{K}_k \boldsymbol{S}_{\boldsymbol{yy},k}$	
SM	Kalman gain derivative	$\boldsymbol{\mathcal{K}}_k = \boldsymbol{\mathcal{P}}_{\boldsymbol{xy},k} / \left(\boldsymbol{I}^{n_\theta} \otimes \boldsymbol{P}_{\boldsymbol{yy},k} \right) - \left(\boldsymbol{P}_{\boldsymbol{xy},k} / \boldsymbol{P}_{\boldsymbol{yy},k} \right) \left(\boldsymbol{\mathcal{P}}_{\boldsymbol{yy},k} / \left(\boldsymbol{I}^{n_\theta} \otimes \boldsymbol{P}_{\boldsymbol{yy},k} \right) \right)$	
	A posteriori state sensitivity	$\widehat{\boldsymbol{\mathcal{X}}}_k^+ = \widehat{\boldsymbol{\mathcal{X}}}_k^- - \boldsymbol{K}_k \widehat{\boldsymbol{y}}_k + \boldsymbol{\mathcal{K}}_k \left(\boldsymbol{I}^{n_\theta} \otimes \left(\boldsymbol{y}_k - \widehat{\boldsymbol{y}}_k \right) \right)$	(d)
	Derivative of estimation error covariance update	$\boldsymbol{\mathcal{S}}_{\boldsymbol{xx},k} = \mathrm{treshapeM}\left(\left(\boldsymbol{I}^{n_\theta} \otimes \boldsymbol{A}_{\mathrm{ls},\boldsymbol{S}_{\boldsymbol{xx},k}^-}^{\dagger} \right) \left(\boldsymbol{\mathcal{P}}_{\boldsymbol{xx},k} \right)_s \right)$, with $\boldsymbol{\mathcal{P}}_{\boldsymbol{xx},k} = \boldsymbol{\mathcal{P}}_{\boldsymbol{xx},k}^- - \boldsymbol{P}_{\boldsymbol{xy},k} \boldsymbol{\mathcal{T}}_{(n_y)} \left(\boldsymbol{I}^{n_\theta} \otimes \boldsymbol{\mathcal{K}}_k^{\mathrm{T}} \right) - \boldsymbol{\mathcal{P}}_{\boldsymbol{xy},k} \left(\boldsymbol{I}^{n_\theta} \otimes \boldsymbol{K}_k^{\mathrm{T}} \right)$	
RPEM	Parameter and covariance estimation	$\boldsymbol{\Delta}_k = \boldsymbol{\Delta}_{k-1} + (1-\lambda)\left(\boldsymbol{e}_k \boldsymbol{e}^{\mathrm{T}} - \boldsymbol{\Delta}_{k-1} \right)$, $\boldsymbol{S}_k = \lambda \boldsymbol{\Delta}_k + \widehat{\boldsymbol{y}}_k \boldsymbol{\Theta}_{k-1} \widehat{\boldsymbol{y}}_k^{\mathrm{T}}$, $\boldsymbol{L}_k = \boldsymbol{\Theta}_{k-1} \, \widehat{\boldsymbol{y}}_k^{\mathrm{T}} \, \boldsymbol{S}_k^{-1}$, $\boldsymbol{\Theta}_k = \left(I^{n_\theta} - \boldsymbol{L}_k \widehat{\boldsymbol{y}}_k \right) \boldsymbol{\Theta}_{k-1} \left(I^{n_\theta} - \boldsymbol{L}_k \widehat{\boldsymbol{y}}_k^{\mathrm{T}} \right) / \lambda + \boldsymbol{L}_k \boldsymbol{\Delta}_k \boldsymbol{L}_k^{\mathrm{T}}$, $\widehat{\boldsymbol{\theta}}_k = \widehat{\boldsymbol{\theta}}_{k-1} + \boldsymbol{L}_k \boldsymbol{e}_k$.	(e)

Table 2. aSRUKF: filter correction step and the RPEM for parameter and covariance estimation.

The RPEM can be combined with different KF derivatives to estimate system parameters and covariances. An EKF and a SRCuKF were used to calculate the output sensitivity in [22], which is then used to estimate the unknown values. More information about the aSRUKF can be found in [28, 30].

3. Testbeds

In this section, two testbeds are presented and modelled. These modelled systems are used in Section 4 to test the performance of the SB-aSRUKF. The planar one-link robot system is presented and extended with a second arm to form a pendulum robot (pendubot). The pendubot is a chaotic and inherently unstable system.

3.1. Planar one-link robot system

This section describes the planar one-link robot system shown in **Figure 1**. It consists of a long rectangular aluminium link driven by a DC motor via a shaft and a one-state toothed gear.

The angular position is measured with an incremental rotary encoder and the motor torque is determined by measuring the motor current. To simplify the motor model, it is assumed that the motor current is directly proportional to the armature current and that the motor torque is proportional to this current by a constant factor. Additionally, the link acceleration is measured using a micro-electro-mechanical sensor (MEMS) accelerometer attached to the link. The motor position is controlled by a classical cascade structure with P- and P-feedback controllers for position and speed.

The corresponding continuous state-space representation of the planar one-link robot system, where the link angular position and acceleration are measured, can be written as follows:

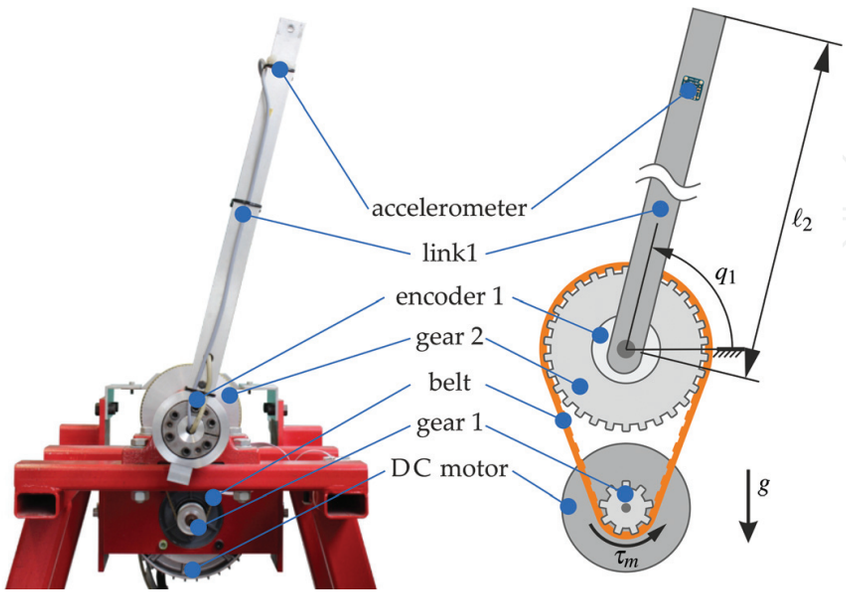

Figure 1. Planar one-link robot system: structure and functionality.

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}(\boldsymbol{x}, u),$$
$$\boldsymbol{y} = \boldsymbol{c}(\boldsymbol{x}, u).$$

The system states are the link angular position ($x_1 = q_1$) and the link speed ($x_2 = \dot{q}_1$). The input u corresponds to the motor torque ($u = \tau_\mathrm{m}$). The measurements are the link angular position ($y_1 = q_1$) and acceleration ($y_2 = \ddot{q}_1$). The matrix $\boldsymbol{A}$ and the vector-valued functions $\boldsymbol{b}$ and $\boldsymbol{c}$ are then described as

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 \\ 0 & -\dfrac{\mu_\mathrm{v}}{J_\mathrm{tot}} \end{pmatrix},$$

$$\boldsymbol{b}(\boldsymbol{x}, u) = \begin{pmatrix} 0 \\ -\dfrac{\mu_\mathrm{d}}{J_\mathrm{tot}} \arctan(k\, x_2) - \dfrac{m_\mathrm{a} l_2}{2\, J_\mathrm{tot}} \sin(x_1) + \dfrac{\tau_\mathrm{m}}{J_\mathrm{tot}} \end{pmatrix},$$

$$\boldsymbol{c}(\boldsymbol{x}, u) = \begin{pmatrix} x_1 \\ -\dfrac{\mu_\mathrm{v}}{J_\mathrm{tot}} x_2 - \dfrac{\mu_\mathrm{d}}{J_\mathrm{tot}} \arctan(k\, x_2) - \dfrac{m_\mathrm{a} l_2\, g}{2\, J_\mathrm{tot}} \sin(x_1) + \dfrac{\tau_\mathrm{m}}{J_\mathrm{tot}} \end{pmatrix},$$

where J_tot represents the total inertia including motor, shaft and link inertias. The link friction is modelled as a dry Coulomb (μ_d and k) and viscous friction (μ_v). The parameters m_a, l_2, and g are the link mass, length, and the gravity of Earth coefficient, respectively.

3.2. Pendubot

This section describes the pendulum robot (pendubot) that is presented in **Figure 2**. The pendubot is built adding an under-actuated link to the planar one-link robot system described in Section 3.1. The actuated joint (q_1) is located at the shoulder of the first link (arm) and the elbow joint (q_2) allows the second link (forearm) to swing free. This joint contains only a second incremental rotatory encoder that measures the angle between the two links.

The system states result as $x_1 = q_1$, $x_2 = \dot{q}_1$, $x_3 = q_2$, and $x_4 = \dot{q}_2$, where q_i and $\dot{q}_i$ are the corresponding position and speed of the i–joint, respectively. The state-space representation of the pendubot can be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}(\boldsymbol{x}, u),$$
$$\boldsymbol{y} = (x_1, x_3, \dot{x}_2)^\mathrm{T},$$

in which

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$
$$\boldsymbol{b}(\boldsymbol{x}, u) = (\,0 \quad \dot{x}_2 \quad 0 \quad \dot{x}_4\,)^\mathrm{T},$$

http://dx.doi.org/10.5772/intechopen.72470

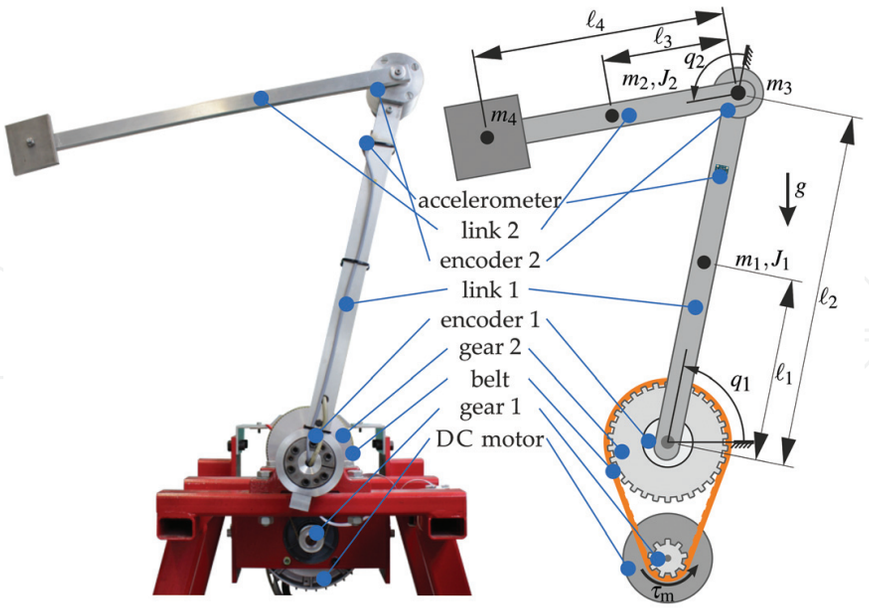

Figure 2. Pendubot: structure and functionality.

where

$$\begin{pmatrix} \dot{x}_2 \\ \dot{x}_4 \end{pmatrix} = \ddot{\boldsymbol{q}} = \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix} = \boldsymbol{D}(\boldsymbol{q})^{-1} \begin{pmatrix} \tau_m - \mu_{v_1} \dot{q}_1 - \mathrm{d}(\dot{q}_1) \\ \mu_{v_2} \dot{q}_2 \end{pmatrix} - \boldsymbol{D}(\boldsymbol{q})^{-1} \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} - \boldsymbol{D}(\boldsymbol{q})^{-1} \boldsymbol{g}(\boldsymbol{q}).$$

The viscous and Coulomb frictions are described with the parameters μ_{v_1} and μ_{v_2} and the function $\mathrm{d}(\dot{q}_1) = \mu_d \arctan(k\,\dot{q}_1)$. The matrices $\boldsymbol{D}(\boldsymbol{q})$ and $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and the vector $\boldsymbol{g}(\boldsymbol{q})$ are the inertial and the Coriolis matrices and the gravity vector, respectively. They are defined as follows

$$\boldsymbol{D}(\boldsymbol{q}) = \begin{pmatrix} \vartheta_1 + \vartheta_2 + 2\vartheta_3 \cos(q_2) & \vartheta_2 + \vartheta_3 \cos(q_2) \\ \vartheta_2 + \vartheta_3 \cos(q_2) & \vartheta_2 \end{pmatrix},$$

$$\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \vartheta_3 \sin(q_2) \begin{pmatrix} -\dot{q}_2 & -\dot{q}_1 - \dot{q}_2 \\ \dot{q}_1 & 0 \end{pmatrix},$$

$$\boldsymbol{g}(\boldsymbol{q}) = \begin{pmatrix} \vartheta_4\, g \cos(q_1) + \vartheta_5\, g \cos(q_1 + q_2) \\ \vartheta_5\, g \cos(q_1 + q_2) \end{pmatrix},$$

where the ϑ_i parameters are defined as

$$\vartheta_1 = m_1 l_1^2 + (m_2 + m_3 + m_4) l_2^2 + J_1,$$

$$\vartheta_2 = m_2 l_3^2 + m_4 l_4^2 + J_2,$$

$$\vartheta_3 = (m_2 l_3 + m_4 l_4)\, l_2,$$

$$\vartheta_4 = m_1 l_1 + (m_2 + m_3 + m_4) l_2,$$
$$\vartheta_5 = m_2 l_3 + m_4 l_4.$$

The parameters J_1 and J_2 correspond to the moments of inertia of link 1 and link 2 about their centroids. J_1 includes motor, gear and shaft inertias. The m_i and l_i parameters are defined in **Figure 2**.

4. Experiments

In this section, the SB-aSRUKF is tested on the planar one-link robot system and on the pendubot. Both testbed models were discretized using first-order explicit Euler with a sample time of 0.2 ms. System states, parameters and covariances were estimated online. The SB-aSRUKF is also compared with the joint state and parameter SRUKF in this section. Sensitivity analysis is also discussed.

4.1. State sensitivity analysis and parameter and covariance estimation

Sensitivity analysis (SA) was performed on simulation using the planar one-link robot system. The system parameters were identified offline on the real testbed using Prediction-Error Method. The parameters defined as

$$\widehat{\boldsymbol{\theta}}_{\text{true}} = \left(\widehat{J}_{\text{tot}}\ \widehat{\mu}_{\text{v}}\ \widehat{\mu}_{\text{d}}\ \widehat{m}_{\text{a}}\widehat{l}_2\ \widehat{k}\right)^{\text{T}} = \left(5.59{\cdot}10^{-2}\text{kg m}^2 \quad 0.05\ \text{N m}\tfrac{\text{s}}{\text{rad}} \quad 0.27\ \text{N m} \quad 0.11\ \text{kg m} \quad 10\tfrac{\text{s}}{\text{rad}}\right)^{\text{T}},$$

were used for the simulation. Noise distributions with the following covariance matrices

$$Q = \text{diag}\left(\left(10^{-11} \quad 5{\cdot}10^{-5}\right)\right)^3,$$
$$R = \text{diag}\left(\left(5{\cdot}10^{-7} \quad 5\right)\right),$$

were added to the simulation to incorporate model and measurement uncertainties. An s-curve profile was considered as a desired link angular position.

The following system states, parameters and covariances were estimated:

$x_1 = q_1$ → link position,
$x_2 = \dot{q}_1$ → link speed,
$\theta_{\text{a}} = q_{11}$ → process covariance value (related to the link ang.pos.),
$\theta_1 = \mu_{\text{v}}$ → viscous friction coefficient,
$\theta_2 = \mu_{\text{d}}$ → Coulomb friction coefficient,
$\theta_3 = J_{\text{tot}}^{-1}$ → inverse inertia.

[3] The function diag(v) transforms the $v \in \mathbb{R}^n$ vector into a $(n \times n)$ square matrix with the elements of v on the diagonal of the matrix.

http://dx.doi.org/10.5772/intechopen.72470

The remaining tuning factors for SB-aSRUKF were set as

$$Q = q_{22} = 5{\cdot}10^{-5}, \quad R = \mathrm{diag}\left(\begin{pmatrix} 5{\cdot}10^{-7} & 2{\cdot}10^{1} \end{pmatrix}\right),$$

$$\alpha = 0.08 \quad \beta = 2 \quad \kappa = 3 - n_x \quad \lambda = 0.999,$$

$$P_{\boldsymbol{xx},\mathrm{init}} = \boldsymbol{I}^2, \quad \boldsymbol{\Theta}_{\mathrm{init}} = \mathrm{diag}\left(\begin{pmatrix} 5{\cdot}10^{-13} & 10^{-1} & 10^{-6} & 10^{-4} & 10^{-8} \end{pmatrix}\right).$$

The values of $\boldsymbol{\Theta}_{\mathrm{init}}$ tune the parameter and covariance estimation, and the index order is the same as the above-defined θ_i values. This means that the first value tunes the estimation of θ_{a} (process covariance value), the second value tunes θ_1 (viscous friction coefficient) and so on.

The filter initial system states were set to zero and the initial system parameters were set as the true values multiply by a random factor between $0 < \theta_{\mathrm{factor},i} \leq 10$ as

$$\boldsymbol{x}_{\mathrm{init}} = \boldsymbol{0},$$

$$\boldsymbol{\theta}_{\mathrm{init}} = \begin{pmatrix} 2{\cdot}10^{-8}\mathrm{rad}^2 & 1.5\left(1/\widehat{J}_{\mathrm{tot}}\right) & 3\widehat{\mu}_{\mathrm{v}} & 8\widehat{\mu}_{\mathrm{d}} & 0.25\left(\widehat{m}_{\mathrm{a}}\widehat{l}_2\right) \end{pmatrix}^{\mathrm{T}}.$$

In order to test the sensitivity-based section of the filter, the link angular position was held at $q_1 = \pi/2$ after ca. 11 s for about 4.5 s. At the same time, the system parameters $\widehat{\mu}_{\mathrm{v}}$, $\widehat{\mu}_{\mathrm{d}}$, and $\widehat{m}_{\mathrm{a}}$ were quadrupled.

Figure 3 compares the a posteriori state sensitivity calculated using the SB-aSRUKF and the state sensitivity using Eq. (10). The first diagram shows the estimated and true link angular position of the planar one-link robot system. The following diagrams present the normalized SA related to the link angular speed ($\widehat{\dot{q}}_1$) and corresponding to the inverse inertia, viscous and Coulomb friction coefficients, and the link mass and length parameter. While the state sensitivity calculated using Eq. (10) was affected directly by input noises, the a posteriori state sensitivity provided an almost noise-free estimation. While the SAJ_{tot}^{-1} maxima were related with the acceleration (speed-up and brake phases), the SAμ_{v} maxima coincided with the link maximal speed. The SA$m_{\mathrm{a}}l_2$ was only zero while the arm was crossing the 0 rad position and the SAμ_{d} was the sensitivity value more affected by the system noise. This is caused because the maximal change rate of arctan() occurs when the argument is near zero. When the link speed is zero, the added noise varies near this value and its effect is amplified by arctan().

Figure 4 shows the state, parameter and covariance estimation of the planar one-link robot system. The aSRUKF was used in two configurations: SB-aSRUKF utilized SA to monitor the system excitation while aSRUKF did not. After the initialization, the two estimators needed almost the same time to converge to the offline identified values. The parameters estimated using the SB-aSRUKF did not diverge while the link position was held. Because two of the estimated parameters using the aSRUKF diverged, this filter needed more time to converge after the stop phase. The two filters were able to estimate the link mass and length parameter during the stop phase. While the viscous and the Coulomb friction coefficients and the inverse inertia estimated with the SB-aSRUKF remained constant during the stop phase, the aSRUKF was able to estimate the Coulomb friction with bias (before μ_{v} diverged). Because of the added noise, the parameter was excited and could be identified. This can be seen in the fourth

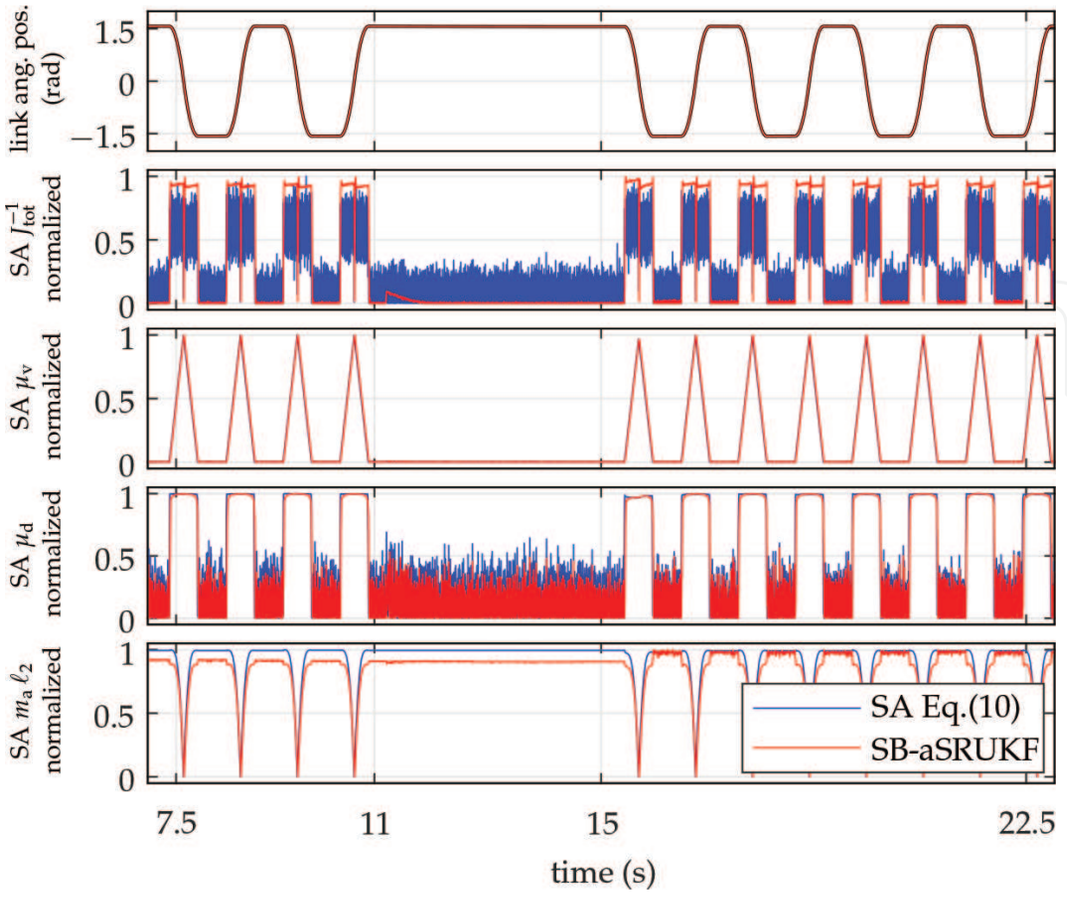

Figure 3. Sensitivity analysis (SA): comparison between the a posteriori state sensitivity obtained using the SB-aSRUKF and the resulting using Eq. (10). The desired link position was set as an s-curve between $-\pi/2$ and $\pi/2$. The link position was held at $\pi/2$ after ca. 11 s for about 4.5 s. The parameter sensitivities are related to the link angular speed ($\widehat{x}_2 = \widehat{q}_1$).

diagram of **Figure 3**. These SA values remained under the threshold used on the SB-aSRUKF and were filtered. The parameter estimation stayed then constant. It should be noted that the diagram corresponding to the viscous friction coefficient is zoomed to present the parameter change, and the oscillations of the aSRUKF are cut. These reached up more than 50 times the parameter true value.

The SB-aSRUKF was able to estimate the parameters and covariance of the proposed testbed. The online estimations converged to the offline identified values without bias. The sensitivity-based approach used as a system excitation monitor prevented parameter estimation drifts and did not modify the convergence time of the filter.

4.2. Comparison between SB-aSRUKF and joint state and parameter SRUKF

The SB-aSRUKF and the joint SRUKF were compared on the pendubot for state and parameter estimation. The SB-aSRUKF identified also covariances.

The system parameters were identified offline and used for the simulation as

$$\widehat{\theta}_{\text{true}} = \left(\widehat{\vartheta}_1\ \widehat{\vartheta}_2\ \widehat{\vartheta}_3\ \widehat{\vartheta}_4\ \widehat{\vartheta}_5\right)^{\mathrm{T}} = \left(0.339\,\text{kg m}^2 \quad 0.0667\,\text{kg m}^2 \quad 0.0812\,\text{kg m}^2 \quad 0.717\,\text{kg m} \quad 0.146\,\text{kg m}\right)^{\mathrm{T}},$$

$$\left(\widehat{\mu}_{\text{v1}}\ \widehat{\mu}_{\text{d}}\ \widehat{\mu}_{\text{v2}}\ \widehat{k}\right) = \left(0.09\,\text{N m}\frac{\text{s}}{\text{rad}} \quad 0.226\,\text{N m} \quad 0.003\,\text{N m}\frac{\text{s}}{\text{rad}} \quad 10\frac{\text{s}}{\text{rad}}\right).$$

http://dx.doi.org/10.5772/intechopen.72470

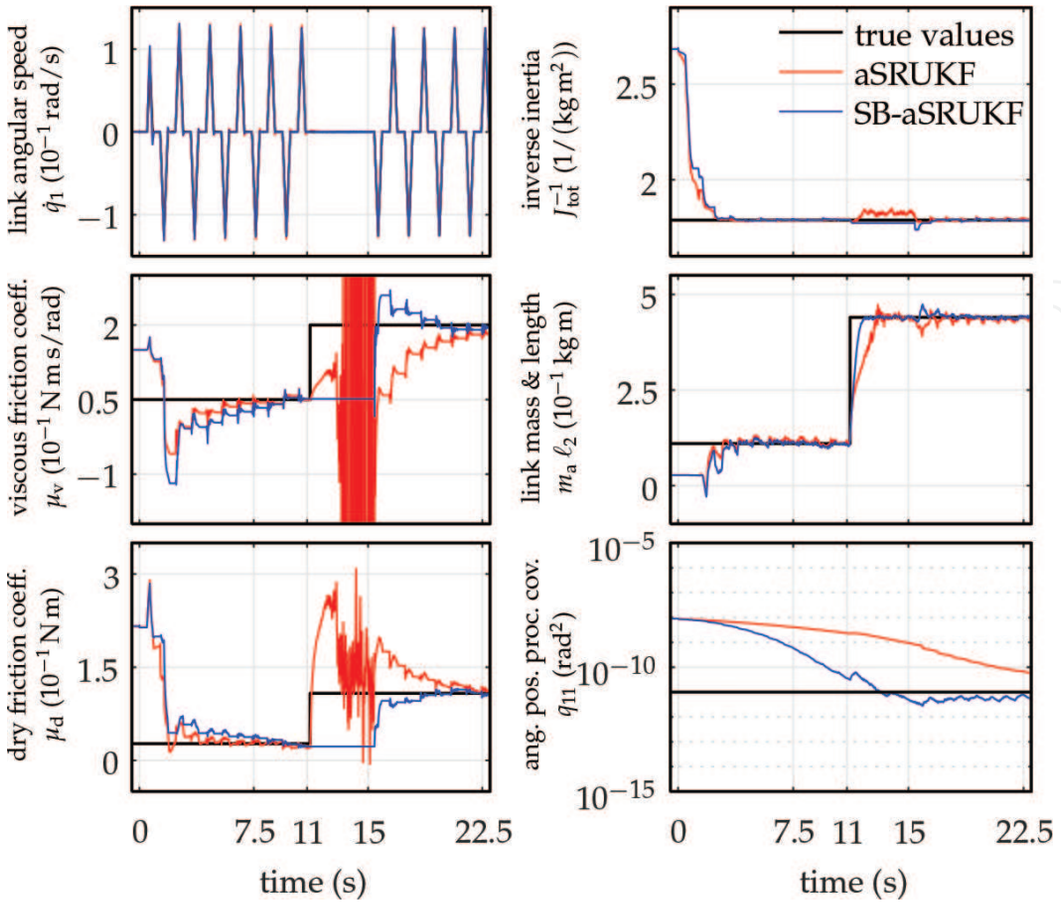

Figure 4. Planar one-link robot system: parameter and covariance estimation. The SB-aSRUKF uses SA to monitor the system excitation. The initial parameter $\boldsymbol{\theta}_{\text{init}}$ was randomly selected. The link position was held after ca. 11 s for about 4.5 s, and simultaneously the system parameters $\widehat{\mu}_{\text{v}}$, $\widehat{\mu}_{\text{d}}$, and $\widehat{m}_{\text{a}}$ were quadrupled.

An s-curve profile was selected as the desired position of the first link. The following states and parameters were estimated online:

$$
\begin{aligned}
x_1 &= q_1 && \rightarrow \text{link 1 position,}\\
x_2 &= \dot{q}_1 && \rightarrow \text{link 1 speed,}\\
x_3 &= q_2 && \rightarrow \text{link 2 position,}\\
x_4 &= \dot{q}_2 && \rightarrow \text{link 2 speed,}\\
\theta_{\text{a}} &= q_{11} && \rightarrow \text{process covariance (related to the link 1 ang.pos.),}\\
\theta_{\text{b}} &= q_{33} && \rightarrow \text{process covariance (related to the link 2 ang.pos.),}\\
\vartheta_1, \ldots, \vartheta_5 && && \rightarrow \text{pedubot minimal set of parameters.}
\end{aligned}
$$

The values θ_{a} and θ_{b}, which correspond to the process covariance values, were only estimated using the SB-aSRUKF. The viscous and Coulomb friction coefficients were identified offline and remained constant for both filters.

To simulate model and measurement uncertainties, noise distributions with the following covariance matrices were added to the system for the simulation:

$$
\begin{aligned}
\boldsymbol{Q} &= \text{diag}\left(\begin{pmatrix} 2{\cdot}10^{-10} & 1.5{\cdot}10^{-7} & 2{\cdot}10^{-10} & 1.5{\cdot}10^{-7} \end{pmatrix}\right),\\
\boldsymbol{R} &= \text{diag}\left(\begin{pmatrix} 5{\cdot}10^{-7} & 5{\cdot}10^{-7} & 1 \end{pmatrix}\right).
\end{aligned}
$$

The tuning parameters for the joint SRUKF were chosen as

$$\boldsymbol{Q}_{0_1} = \mathrm{diag}\big(\big(2{\cdot}10^{-10} \quad 1.5{\cdot}10^{-7} \quad 2{\cdot}10^{-10} \quad 1.5{\cdot}10^{-7} \quad 10^{-7} \quad 10^{-7} \quad 10^{-7} \quad 5{\cdot}10^{-10} \quad 10^{-10}\big)\big),$$

$$\boldsymbol{R}_{0_1} = \mathrm{diag}\big(\big(5{\cdot}10^{-7} \quad 5{\cdot}10^{-7} \quad 10\big)\big),$$

$$\alpha = 0.85, \quad \beta = 2, \quad \kappa = 3 - n_x - n_p,$$

$$\boldsymbol{P}_{0_1} = \mathrm{diag}\big(\big(1 \quad 1 \quad 1 \quad 1 \quad 10^{-3} \quad 10^{-5} \quad 10^{-5} \quad 5 \quad 1\big)\big),$$

and the parameters for the SB-aSRUKF were set as

$$\boldsymbol{Q}_{0_2} = \mathrm{diag}\big(\big(1.5{\cdot}10^{-7} \quad 1.5{\cdot}10^{-7}\big)\big), \quad \boldsymbol{P}_{xx,0_2} = \boldsymbol{I}^4,$$

$$\boldsymbol{R}_{0_2} = \mathrm{diag}\big(\big(5{\cdot}10^{-7} \quad 5{\cdot}10^{-7} \quad 10\big)\big),$$

$$\alpha = 0.85, \quad \beta = 2, \quad \kappa = 3 - n_x, \quad \lambda = 0.999,$$

$$\boldsymbol{\Theta}_{\text{init}} = \mathrm{diag}\big(\big(10^{-25} \quad 10^{-25} \quad 10^{-7} \quad 10^{-7} \quad 10^{-7} \quad 5{\cdot}10^{-10} \quad 10^{-10}\big)\big).$$

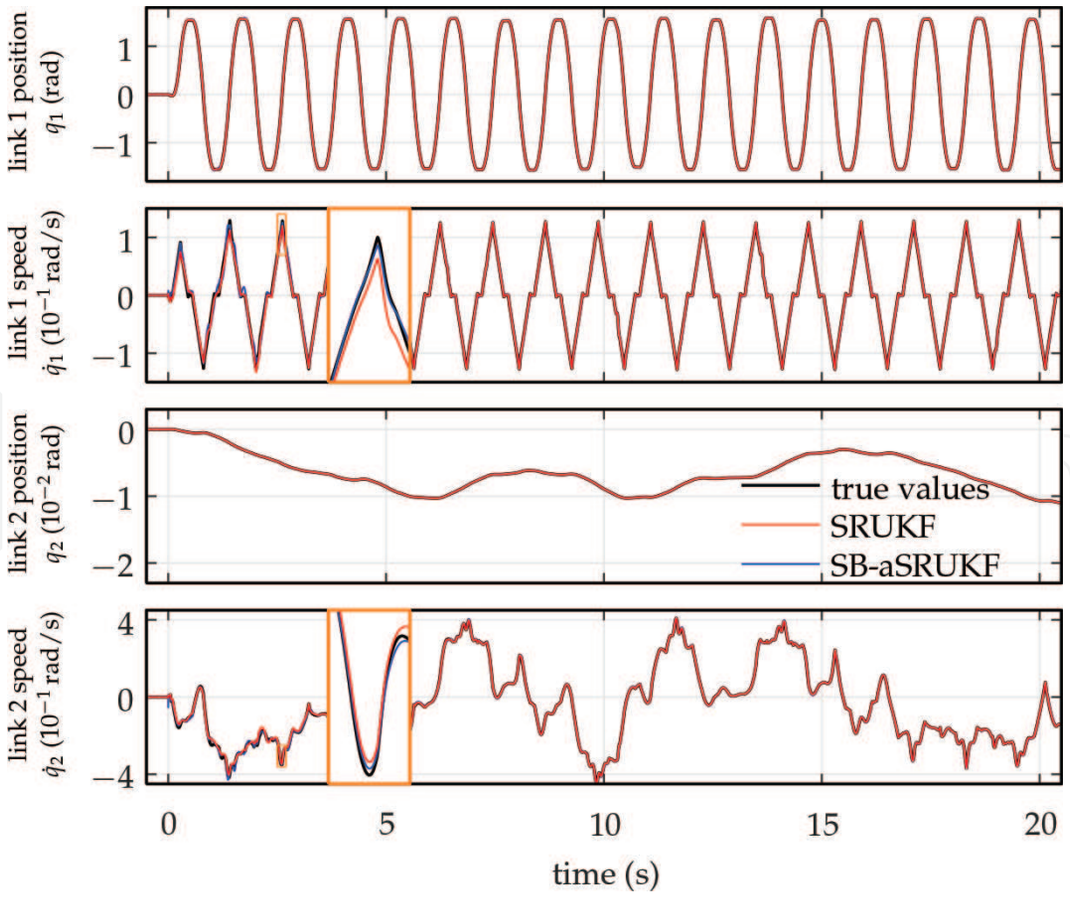

Figure 5. Pendubot: state estimation using the SB-aSRUKF and joint SRUKF. Both filters followed the dynamic of the true system states without any significant bias.

The filter initial system states were set to zero and the initial system parameters were set as the true values multiply by a random factor between $0 < \theta_{\text{factor},i} \leq 5$ as

$$\boldsymbol{x}_{\text{init}} = \mathbf{0},$$
$$\boldsymbol{\theta}_{\text{init}_1} = \left(1.5\,\widehat{\vartheta}_1\ 1.3\,\widehat{\vartheta}_2\ 1.5\,\widehat{\vartheta}_3\ 2\,\widehat{\vartheta}_4\ 2\,\widehat{\vartheta}_5\right)^{\mathrm{T}},$$
$$\boldsymbol{\theta}_{\text{init}_2} = \left(2{\cdot}10^{-10}\text{rad}^2 \quad 2{\cdot}10^{-10}\text{rad}^2 \quad (\boldsymbol{\theta}_{\text{init}_1})^{\mathrm{T}}\right)^{\mathrm{T}}.$$

The first four values of $\boldsymbol{P}_{0_1}$ tune the initial state estimation, while the last ones the parameter estimation. The first two values of $\boldsymbol{\Theta}_{\text{init}}$ tune the estimation of the covariance values θ_{a} and θ_{b} while the last values follow the index order of ϑ_i defined in Section 3. It should be noted that the settings related to the state and parameters estimation were equally tuned for both filters.

The state estimation of the pendubot is presented in **Figure 5**. The SB-aSRUKF was slightly faster to reach the true system states (cf. diagrams 1 and 4) and after ca. 5 s both filters followed the dynamic of the true system states without any significant bias.

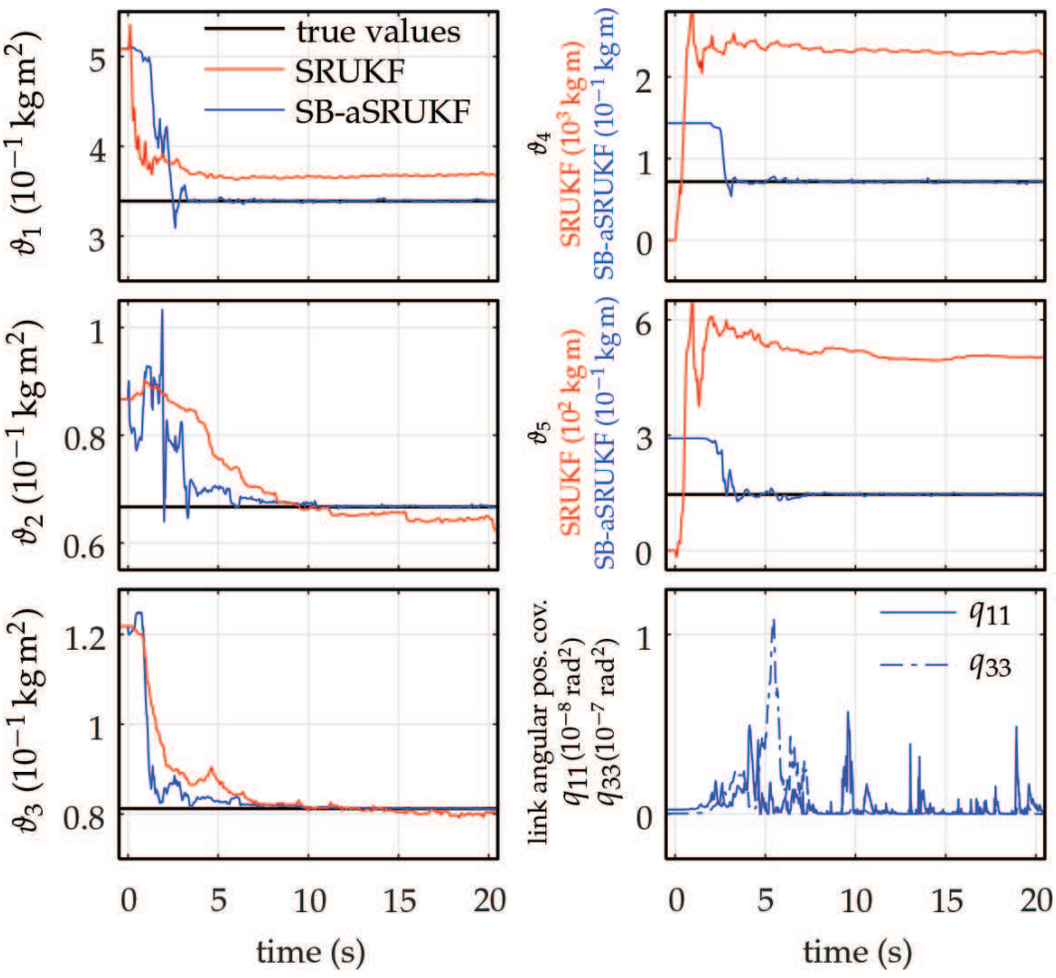

Figure 6. Pendubot: parameter and covariance estimation using the SB-aSRUKF and joint SRUKF. The SB-aSRUKF was configured to estimate the system parameters ϑ_1 to ϑ_5 and the process covariances corresponding to the link positions. The joint SRUKF estimates only the system parameters. It should be noted that the diagram scales for parameters ϑ_4 and ϑ_5 are different between the filters.

Figure 6 shows the parameter estimation of the pendubot for both filters. Using the same tuning parameter set, while the SB-aSRUKF estimated the ϑ_1 to ϑ_5 parameters without bias, the joint SRUKF estimated ϑ_1 to ϑ_3 with slight bias, and it was not able to estimate ϑ_4 and ϑ_5. These two parameters did not converge to the true system values. It should be noted that the diagram scales corresponding to parameters ϑ_4 and ϑ_5 are different between the filters. The parameter initialization and the tuning settings for the two filters were the same. The SB-aSRUKF outperforms the joint SRUKF for the parameter estimation of the pendubot.

5. Conclusions

In this chapter, some approaches for state, parameter and covariance estimation were discussed. Moreover, a sensitivity-based adaptive square-root unscented Kalman filter (SB-aSRUKF) was discussed and its performance was analysed. This filter estimates system states, parameters and covariances online. Additionally, sensitivity models were presented and used as system excitation monitor to prevent parameter and covariance estimation drifts while the system excitation was not sufficient.

Two testbeds based on an axis serial mechanism were modelled and used for testing the performance of the filter. Two experiments were performed in simulation on these two testbeds: a state sensitivity analysis and a comparison between the joint state and parameter square-root unscented Kalman filter (SRUKF) and the SB-aSRUKF. Simulation results show that the SB-aSRUKF outperforms the joint SRUKF with the same tuning configuration. While the joint SRUKF did not estimate two of the five parameters correctly, the SB-aSRUKF identified all the parameters. Moreover, the estimation of covariances reduces the parameter estimation bias. Configuring the right excitation thresholds for the system excitation monitor in Eq. (9) prevented parameter estimation drifts, while the input was not persistently exciting and did not only increased but also in some cases reduced the convergence time of the filter.

Author details

Mauro Hernán Riva*, Mark Wielitzka and Tobias Ortmaier

*Address all correspondence to: mauro.riva@imes.uni-hannover.de

Institute of Mechatronic Systems, Leibniz Universität Hanover, Hanover, Germany

References

[1] Wielitzka M, Dagen M, Ortmaier T. State estimation of vehicle's lateral dynamics using unscented Kalman filter. 53rd IEEE Conference on Decision and Control (CDC); Los Angeles, USA. 2014:5015-5020. DOI: 10.1109/CDC.2014.7040172

[2] Ding SX. Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools. 1st ed. Berlin, Germany: Springer; 2008. p. 493

[3] Simon D, Simon DL. Kalman filtering with inequality constraints for turbofan engine health estimation. IEE Proceedings - Control Theory and Applications. 2006;**153**(3):371-378. DOI: 10.1049/ip-cta:20050074

[4] Grewal MS, Andrews AP. Kalman Filtering: Theory and Practice with MATLAB. 4th ed. Hoboken, New Jersey, USA: Wiley-IEEE Press; 2014. p. 640

[5] Sorenson HW. Least-squares estimation: from Gauss to Kalman. IEEE Spectrum. 1970;**7**(7):63-68. DOI: 10.1109/MSPEC.1970.5213471

[6] Kalman RE. A new approach to linear filtering and prediction problems. Journal of Basic Engineering. 1960;**82**(1):35-45. DOI: 10.1115/1.3662552

[7] Anderson BDO, Moore JB. Optimal Filtering. 1st ed. Mineola, NY, USA: Dover Publications; 2006. p. 368

[8] Schmidt SF. Applications of state space methods to navigation problems. C. T. Leondes (Ed.) Advanced in Control Systems. 1966;**3**:293-340

[9] Simon D. Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches. 1st ed. Hoboken, New Jersey, USA: Wiley-Interscience; 2006. p. 526

[10] Julier SJ, Uhlmann JK. New extension of the Kalman filter to nonlinear systems. In: Signal Processing, Sensor Fusion, and Target Recognition VI. Orlando, FL, USA: International Society for Optics and Photonics; 1997. pp. 182-194. DOI: 10.1117/12.280797

[11] Ljung L, Soderstrom T. Theory and Practice of Recursive Identification. 1st ed. Cambridge, Massachusetts, USA: The MIT Press; 1983. p. 551

[12] Ljung L. System Identification: Theory for the User. 2nd ed. Prentice Hall: Upper Saddle River, NJ, USA; 1999. p. 672

[13] Saab SS, Nasr GE. Sensitivity of discrete-time Kalman filter to statistical modeling errors. Optimal Control Applications and Methods. 1999;**20**(5):249-259. DOI: 10.1002/(SICI)1099-1514(199909/10)20:5<249::AID-OCA659>3.0.CO;2-2

[14] Fitzgerald R. Divergence of the Kalman filter. IEEE Transactions on Automatic Control. 1971;**16**(6):736-747. DOI: 10.1109/TAC.1971.1099836

[15] Price C. An analysis of the divergence problem in the Kalman filter. IEEE Transactions on Automatic Control. 1968;**13**(6):699-702. DOI: 10.1109/TAC.1968.1099031

[16] Ljung L. Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems. IEEE Transactions on Automatic Control. 1979;**24**(1):36-50. DOI: 10.1109/TAC.1979.1101943

[17] Odelson B. Estimating Disturbance Covariances From Data For Improved Control Performance [dissertation]. Madison, WI, USA: University of Wisconsin-Madison; 2003. p. 309

[18] Odelson BJ, Rajamani MR, Rawlings JB. A new autocovariance least-squares method for estimating noise covariances. Automatica. 2006;**42**(6):303-308. DOI: 10.1016/j.automatica.2005.09.006

[19] Rajamani MR, Rawlings JB. Estimation of the disturbance structure from data using semidefinite programming and optimal weighting. Automatica. 2009;**45**(1):142-148. DOI: 10.1016/j.automatica.2008.05.032

[20] Riva MH, Díaz Díaz J, Dagen M, Ortmaier T. Estimation of covariances for Kalman filter tuning using autocovariance method with Landweber iteration. In: IASTED International Symposium on Intelligent Systems and Control; Marina del Rey, CA, USA. 2013. DOI: 10.2316/P.2013.807-009

[21] Rajamani MR. Data-based Techniques to Improve State Estimation in Model Predictive Control [dissertation]. Madison, WI, USA: University of Wisconsin-Madison; 2007. p. 262

[22] Riva MH, Beckmann D, Dagen M, Ortmaier T. Online parameter and process covariance estimation using adaptive EKF and SRCuKF approaches. In: 2015 IEEE Conference on Control Applications (CCA); Sydney, Australia. 2015. p. 1203-1210. DOI: 10.1109/CCA.2015.7320776

[23] Han J, Song Q, He Y. Adaptive unscented Kalman filter and its applications in nonlinear control. In: Moreno VM, Pigazo A, editors. Kalman Filter Recent Advances and Applications. Croatia: InTech; 2009. pp. 1-24. DOI: 10.5772/6799

[24] Wan EA, Van der Merwe R. The unscented Kalman filter for nonlinear estimation. In: IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium; Lake Louise, Canada. 2000. p. 153-158. DOI: 10.1109/ASSPCC.2000.882463

[25] Van der Merwe R, Wan EA. The square-root unscented Kalman filter for state and parameter-estimation. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 01); Salt Lake City, USA. 2001. p. 3461-3464. DOI: 10.1109/ICASSP.2001.940586

[26] Press WH, Flannery BP, Teukolsky SA, Vetterling WT. Numerical Recipes in C: The Art of Scientific Computing. 2nd ed. Cambridge, New York, USA: Cambridge University Press; 1992. p. 994

[27] Henderson HV, Searle SR. The vec-permutation matrix, the vec operator and Kronecker products: A review. Linear and Multilinear Algebra. 1981;**9**(4):271-288. DOI: 10.1080/03081088108817379

[28] Riva MH, Dagen M, Ortmaier T. Adaptive Unscented Kalman Filter for online state, parameter, and process covariance estimation. In: 2016 American Control Conference (ACC); Boston, MA, USA. 2016. p. 4513-4519. DOI: 10.1109/ACC.2016.7526063

[29] Eykhoff P. System Identification Parameter and State Estimation. 1st ed. Hoboken, New Jersey, USA: Wiley-Interscience; 1974. p. 555

[30] Riva MH, Dagen M, Ortmaier T. Comparison of covariance estimation using autocovariance LS method and adaptive SRUKF. In: 2017 American Control Conference (ACC); Seattle, WA, USA. 2017. p. 5780-5786. DOI: 10.23919/ACC.2017.7963856