





# Accuracy and efficiency comparison of various nonlinear Kalman filters applied to multibody models

Emilio Sanjurjo  · Miguel Ángel Naya  ·  
José Luis Blanco-Claraco  ·  
José Luis Torres-Moreno  ·  
Antonio Giménez-Fernández

Received: 29 April 2016 / Accepted: 11 January 2017 / Published online: 2 February 2017  
© Springer Science+Business Media Dordrecht 2017

**Abstract** Multibody simulations are already used in many industries to speed up the development of new products. However, improvements in multibody formulations and the continuous increase in inexpensive computational power open new fields of applications for multibody simulations, such as using them as plant models for state observers. The present work introduces a novel state observer developed by combining a multibody model with an indirect Kalman filter. Together with other multibody-based Kalman filters already proposed in the literature, they have been applied to two mechanisms (four- and five-bar linkages) to assess their performance. The accuracy of the estimations and the computational cost are examined under several scenarios: using position or velocity sensors in different configurations, with different sampling rates, and consid-

ering different levels of errors in the models assumed by the filters. The aim of this work is to present different options to build a state observer based on a dynamic multibody model, and to provide guidance to select the most suitable option for future applications. All the methods have been implemented as a reusable MATLAB<sup>®</sup> toolkit which has been released as Open Source in <https://github.com/MBDS/mbde-matlab>.

**Keywords** Multibody dynamics · Kalman filter · State observer · Benchmark

## 1 Introduction

Being able to perform dynamic analyses of complex multibody systems (MBS) before manufacturing is key in achieving more efficient and competitive industries in sectors such as automotive or aeronautics. Doing so allows the manufacturer to study the expected dynamic behavior of products before building real prototypes, boosting the testing and development of new products and their associated electronic controllers.

However, simulating the dynamics of MBS has other farther-reaching applications, such as devising state observers. A state observer is a recursive Bayesian estimator [7, 18] aimed at providing a real-time best estimation of some parameters or states of a plant (e.g., a machine or a vehicle in the field of dynamics), by means of integrating a more or less accurate dynamical model with the history of sensor measurements. By employ-

---

E. Sanjurjo (✉) · M. Á. Naya  
Mechanical Engineering Laboratory, University of A  
Coruña, Escuela Politécnica Superior, Mendizábal s/n,  
15403 Ferrol, Spain  
e-mail: emilio.sanjurjo@udc.es

J. L. Blanco-Claraco · J. L. Torres-Moreno ·  
A. Giménez-Fernández  
Department of Engineering, Automatic Control, Robotics  
and Mechatronics Research Group, University of Almería,  
Agrifood Campus of International Excellence (ceiA3),  
Almería, Spain

J. L. Blanco-Claraco · J. L. Torres-Moreno ·  
A. Giménez-Fernández  
CIESOL, Joint Center University of Almería-CIEMAT,  
04120 Almería, Spain

ing accurate MBS models, the state of non-accessible parts of a mechanism can be inferred where installing a sensor would not be possible for economical or technical reasons.

Usually, when a state observer is needed, a simplified analytical model is specifically developed to be used as the model of the state observer (see, e.g., [1]). The scope of this paper is limited to the nonlinear variants of the Kalman filter, although other possibilities, such as the particle filter, have been successfully applied to multibody models [3]. Using a multibody model as the plant of a Kalman Filter provides a means to design state observers systematically. However, coupling both systems is not a trivial task, since the Kalman filter is formulated for first-order, linear, unconstrained systems, whereas the multibody models are, in general, second-order, nonlinear and constrained systems. This reason produced a high research effort to combine the generality and accuracy of a multibody model with a Kalman filter to obtain improved state observer formulations. A continuous extended Kalman filter was presented in [5], applied to a four-bar mechanism. A new version of the method was presented in [6] applied to an automobile, although the computational cost of the algorithm prevented it from running in real time. In [15] the unscented Kalman filter and the spherical simplex unscented Kalman filter were first used with multibody models and compared with the continuous extended Kalman filter. The particularity of the unscented Kalman filters that were proposed in this work is that the accelerations of the multibody model were used as states, even when the aim was to estimate the position and velocity of the system. In [12], a reduced multibody model is combined with an augmented Kalman filter to estimate states and input forces of a half car model, although the sensor configuration used led to a non-observable configuration. In [14] the kinematics of a multibody model is also combined with a Kalman filter. This approach is simpler than the dynamic filters cited previously and can be useful if force models are unknown or have a high uncertainty, but it requires more sensors and the frequency of the observer is limited by the sampling rate of the sensors used as inputs.

In the present work, four different methods for coupling multibody models and Kalman filters are presented. They are tested with two planar mechanisms (four-bar and five-bar linkages). Some of the methods presented here are variations of algorithms already

shown in [5, 15, 19], while the indirect formulation is new in the multibody literature. Both accuracy and computational cost of the estimators are evaluated, considering different sets of sensors, several sampling rates for the sensors, and two levels of modeling error of the plant.

In order to test the proposed methods, a four-bar mechanism has been selected because it is a simple closed-chain mechanism. In addition, a five-bar mechanism is also used, allowing to verify the variation of computational cost with the size of the model. All the methods presented here can be applied to both open-chain and closed-chain mechanisms; however, the mechanisms selected are closed chain because they allow to show all the benefits from incorporating a multibody model into a Kalman filter. Although the methods presented here are applied to planar mechanism, they are general and can be employed with any mechanism.

The different algorithms considered in this work have been implemented in MATLAB<sup>®</sup> as a reusable toolbox released as Open Source.<sup>1</sup> Some of the estimators presented here have already been implemented in C++ and installed in a test bench in [20], demonstrating that they can run faster than real time with an adequate implementation and provide accurate results in real conditions.

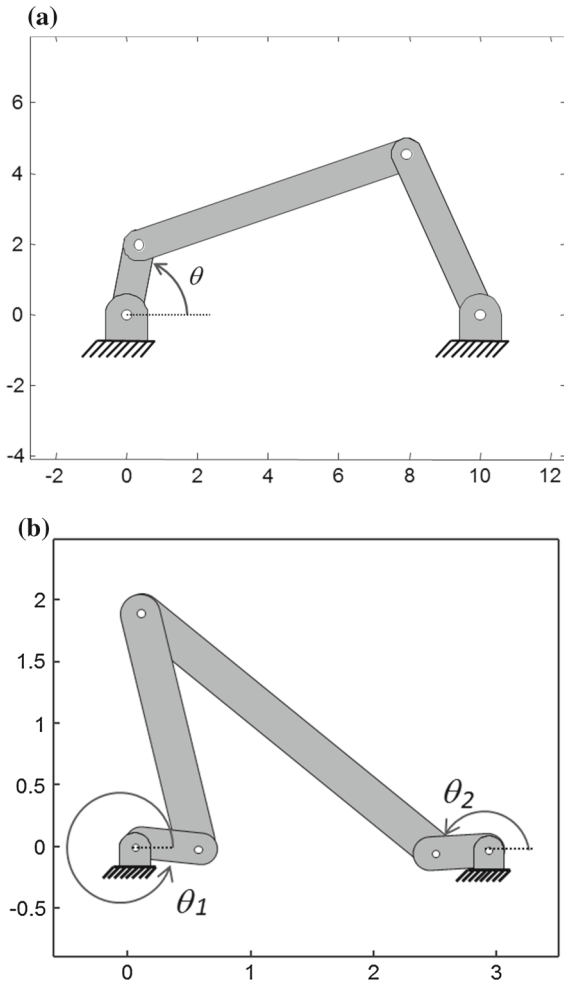
## 2 Methodology

In this work, a central question in MBS state observers is addressed, namely deciding which estimation algorithm is the best choice among the large number of Kalman filter based methods [2, 18].

Therefore, a benchmark consisting in two simple mechanisms (planar four-bar and five-bar linkages, shown in Fig. 1) has been performed with the aim of analyzing the accuracy and relative efficiency of the selected estimators. The main properties of the mechanism are provided in Tables 1 and 2.

For every test, a multibody simulation was performed using a model which was considered as the real mechanism, thus providing the ground truth to verify the results delivered by the observers. This simulation is also employed to build the signals from the sensors. To do this, perfect sensors are modeled, and then

<sup>1</sup> See <https://github.com/MBDS/mbde-matlab>.



**Fig. 1** Mechanisms employed in this work. **a** Four-bar linkage and **b** five-bar linkage

**Table 1** Properties of the four-bar linkage

	Crank	Coupler	Rocker	Ground element
Mass (kg)	2	8	5	–
Length (m)	2	8	5	10

**Table 2** Properties of the five-bar linkage

	Left crank	Left coupler	Right coupler	Right crank	Ground element
Mass (kg)	3	1	2	3	–
Length (m)	0.5	2.062	3.202	0.5	3

pseudo-random noise is added to their measurements. The sequence of the pseudo-random values of the noise is the same for all the tests carried out, enabling a fair comparison among the different methods.

A second multibody model is built, but modifying some of the properties, to simulate modeling error. Usually, the geometry of any machine or vehicle can be known with great accuracy. However, the level of accuracy in the determination of forces and mass distribution is often not so good, both producing acceleration errors. Therefore, the parameter which is intentionally modified is the acceleration of gravity, leading to an erroneous acceleration. Moreover, the initial position of the mechanism is also modified to simulate situations in which it is not exactly known. Finally, the state observer is built using the latter multibody model (the imperfect one), and corrected with the information provided by the noisy sensors built from the simulation of the first multibody model.

All the multibody simulations were run with a time step of  $5e-3$ s. The sensors considered in the test were position and velocity sensors in different configurations, and their sampling rates vary from 200 Hz (the frequency of the multibody simulations) to 10 Hz. Therefore, the measurements are not available at every integration time step when the sampling rate is lower than 200 Hz.

### 3 Design of the observers

The state observers considered in this work are based on the combination of a multibody model and a Kalman filter. The different mathematical structure of both systems produces that the coupling between them is not obvious, allowing different strategies which are described in the following sections. On the one hand, we deal with methods in independent coordinates. Among these methods, we consider the continuous and discrete extended Kalman filters, and the unscented

**Table 3** Summary of the methods

Method	Classification	State vector	Integrator
CEKF <sup>a</sup>	Continuous-time EKF	Indep. coord. and vel.	Trap. rule
DEKF <sup>b</sup>	Discrete-time EKF	Indep. coord. and vel.	Forward Euler
UKF <sup>c</sup>	Unscented KF	Indep. coord. and vel.	Any
errorEKF <sup>d</sup>	Discrete-time EKF, indirect formulation	Error in indep. coord. and vel.	Any for the MBS, Forward Euler for the KF

<sup>a</sup> First presented in [5]

<sup>b</sup> First presented in [20]

<sup>c</sup> Similar to the UKF presented in [15], but with different states. First presented in [20] in its current form

<sup>d</sup> Original contributions of this work

Kalman filter. On the other hand, we study the error-state Kalman filter, an indirect formulation in which the error of the multibody simulation is estimated and then the model is corrected. There are other options, such as methods in dependent coordinates, where the constraints are included as perfect measurements, or where the unconstrained estimation is projected over the constraints manifold [18], but these options have not been considered in this work for the sake of clarity. A summary of the methods is shown in Table 3. For ease of reference, the list of the principal symbols used is given in Table 4.

### 3.1 Independent coordinates methods

The Kalman filter assumes that its states are independent, and hence, the most natural way to combine a multibody model and a Kalman filter is by using the independent coordinates and velocities of the multibody model as the states of the Kalman filter.

#### 3.1.1 Continuous extended Kalman filter (CEKF)

This formulation is based on the one developed in [5] but adapted to deal with multirate (i.e., some or all the sensor's information might not be available at every time step). Due to the complexity of the dynamical model, approximate Jacobian matrices are employed in this method.

The main idea under this formulation is to adapt the multibody equations in order to fit the Kalman filter structure. Then, the resulting equations are integrated using the trapezoidal rule.

In its most basic form, the dynamics of a multibody system is described by the constrained Lagrangian equations:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \boldsymbol{\lambda} = \mathbf{Q} \\ \Phi = \mathbf{0} \end{cases} \quad (1)$$

where  $\mathbf{M}$  is the mass matrix,  $\ddot{\mathbf{q}}$  is the vector of dependent accelerations,  $\Phi$  is the constraints vector,  $\Phi_{\mathbf{q}}$  is the Jacobian matrix of the constraint equations with respect to the generalized coordinates  $\mathbf{q}$ ,  $\boldsymbol{\lambda}$  is the vector of Lagrange multipliers, and  $\mathbf{Q}$  is the vector of generalized forces.

As the multibody equations are expressed in the form of continuous-time differential-algebraic equations (DAEs), it seems natural to adopt the continuous-time version of the Kalman filter. To fit the Kalman filter structure, the system of DAEs must be transformed into a system of ordinary differential equations (ODEs). Two alternatives were explored in [5], the matrix-R method, and the penalty method [9], but the penalty method did not provide adequate results, because it could not impose the constraint equations fulfillment when the corrections provided by the Kalman filter were applied. Hence, only the version based on the matrix-R formulation is considered in the present work. The main idea behind this multibody formulation is to obtain a system of ordinary differential equations whose dimension  $g$  is equal to the number of degrees of freedom, starting with the identity  $\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{z}}$ , which relates dependent velocities  $\dot{\mathbf{q}}$  and independent velocities  $\dot{\mathbf{z}}$  through projection matrix  $\mathbf{R}$ . Accelerations can be then expressed as follows:

$$\ddot{\mathbf{q}} = \mathbf{R}\ddot{\mathbf{z}} + \dot{\mathbf{R}}\dot{\mathbf{z}} \quad (2)$$

Going back to Eq. (1), premultiplying by the transpose of  $\mathbf{R}$ , and keeping in mind that  $\Phi_{\mathbf{q}}\mathbf{R} = \mathbf{0}$ ,

$$\ddot{\mathbf{z}} = (\mathbf{R}^T \mathbf{M} \mathbf{R})^{-1} \left[ \mathbf{R}^T (\mathbf{Q} - \mathbf{M} \dot{\mathbf{R}} \dot{\mathbf{z}}) \right] := \bar{\mathbf{M}}^{-1} \bar{\mathbf{Q}} \quad (3)$$

**Table 4** List of symbols

$\bar{\mathbf{C}}$	Damping matrix
$f(), \tilde{\mathcal{F}}$	Continuous transition model, and its Jacobian matrix wrt. the states
$\mathbf{f}(), \mathbf{f}_x$	Discrete transition model, and its Jacobian matrix wrt. the states $\mathbf{x}$
$g$	Number of degrees of freedom
$\mathbf{h}(), \mathbf{h}_x$	Measurement model and its Jacobian matrix wrt. the states $\mathbf{x}$
$\mathbf{G}$	Kalman gain matrix
$\bar{\mathbf{K}}$	Stiffness matrix
$\mathbf{M}$	Mass matrix
$\bar{\mathbf{M}}$	Mass matrix projected over the set of independent coordinates
$m$	Number of constraints
$n$	Number of dependents coordinates
$\mathbf{o}$	Vector of measurements
$\mathbf{P}$	Covariance matrix of the state estimation uncertainty
$\mathbf{Q}$	Vector of generalized forces
$\bar{\mathbf{Q}}$	Vector of generalized forces projected over the dependent coordinates
$\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$	Vector of dependent positions, velocities, and accelerations
$\hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}$	Estimation of the vectors of dependent positions and velocities
$\mathbf{R}$	Velocity projection matrix, such that $\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{z}}$
$\mathbf{S}$	Covariance matrix of the innovation
$\mathbf{u}$	Vector of inputs
$\mathbf{x}, \dot{\mathbf{x}}$	States vector, and its time derivative
$\hat{\mathbf{x}}^-, \hat{\mathbf{x}}^+$	‘A priori’ state estimation, ‘a posteriori’ state estimation
$\mathbf{y}$	Innovations vector
$\mathcal{Y}_k(i)$	Innovation according to $i$ th sigma point of time step $k$
$\mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}}$	Vector of independent coordinates, velocities, and accelerations
$\hat{\mathbf{z}}, \hat{\dot{\mathbf{z}}}$	Estimation of the vectors of independent coordinates and velocities
$\Delta t$	Time step
$\boldsymbol{\lambda}$	Vector of Lagrange multipliers
$\boldsymbol{\Sigma}^P$	Covariance matrix of the plant noise in discrete filters
$\boldsymbol{\Sigma}_c^P$	Covariance matrix of the plant noise in continuous filters
$\boldsymbol{\Sigma}^S$	Covariance matrix of measurements noise
$\Phi, \Phi_q$	Vector of constraints, and its Jacobian wrt. the coordinates
$\chi_k(i)$	$i$ th sigma point of time step $k$

where  $\bar{\mathbf{M}}^{-1}$  and  $\bar{\mathbf{Q}}$  are the projected mass matrix and the projected vector of generalized forces, respectively. If now the filter state is defined as the vector  $\mathbf{x}^\top = [\mathbf{z}^\top, \dot{\mathbf{z}}^\top]$ , containing the vectors of independent positions  $\mathbf{z}$  and velocities  $\dot{\mathbf{z}}$ , it turns out that:

$$\begin{bmatrix} \hat{\dot{\mathbf{z}}} \\ \hat{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \hat{\dot{\mathbf{z}}} \\ \hat{\mathbf{z}} \end{bmatrix} \Rightarrow \hat{\mathbf{x}} = \mathbf{f}(\hat{\mathbf{x}}) \tag{4}$$

where the hat  $\wedge$  denotes estimated magnitudes. These equations perfectly fit the continuous extended Kalman filter equation, so they can be straightforwardly applied. In particular, the state-space transition matrix is obtained as the linearization:

$$\tilde{\mathcal{F}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0}_{g \times g} & \mathbf{I}_g \\ \frac{\partial(\bar{\mathbf{M}}^{-1}\bar{\mathbf{Q}})}{\partial \mathbf{z}} & \frac{\partial(\bar{\mathbf{M}}^{-1}\bar{\mathbf{Q}})}{\partial \dot{\mathbf{z}}} \end{bmatrix} \tag{5}$$

which can be approximated by:

$$\tilde{\mathcal{F}} \simeq \begin{bmatrix} \mathbf{0}_{g \times g} & \mathbf{I}_g \\ \tilde{\mathcal{F}}_{21} & \tilde{\mathcal{F}}_{22} \end{bmatrix} \tag{6}$$

$$\tilde{\mathcal{F}}_{21} = -\bar{\mathbf{M}}^{-1}\mathbf{R}^\top (\bar{\mathbf{K}}\mathbf{R} + 2\mathbf{R}_q\mathbf{R}\ddot{\mathbf{z}}) \tag{7}$$

$$\tilde{\mathcal{F}}_{22} = -\bar{\mathbf{M}}^{-1}\mathbf{R}^\top (\bar{\mathbf{C}}\mathbf{R} + \mathbf{M}\dot{\mathbf{R}}) \tag{8}$$

where  $\bar{\mathbf{K}} = -\frac{\partial \bar{\mathbf{Q}}}{\partial \mathbf{q}}$  and  $\bar{\mathbf{C}} = -\frac{\partial \bar{\mathbf{Q}}}{\partial \dot{\mathbf{q}}}$  are the stiffness and damping matrices, respectively, and  $\mathbf{R}_q = \frac{\partial \mathbf{R}}{\partial \mathbf{q}}$ . In this case, the size of the problem is  $2g$ , being  $g$  the number of degrees of freedom of the mechanism. Next, the CEKF correction stage is introduced [18], which fuses the sensor information into the filter, leading to:

$$\dot{\mathbf{z}} - \hat{\dot{\mathbf{z}}} + \mathbf{G}^z(\mathbf{h}(\mathbf{x}) - \mathbf{o}) = \mathbf{0}_{g \times 1} \tag{9}$$

$$\bar{\mathbf{M}}\ddot{\mathbf{z}} - \bar{\mathbf{Q}} + \bar{\mathbf{M}}\mathbf{G}^z(\mathbf{h}(\mathbf{x}) - \mathbf{o}) = \mathbf{0}_{g \times 1} \tag{10}$$

where  $\mathbf{G}^z$  and  $\mathbf{G}^z$  are the parts of the Kalman gain matrix correspondent to  $\mathbf{z}$  and  $\dot{\mathbf{z}}$ , respectively,  $\mathbf{h}(\mathbf{x})$  is the measurement model, and  $\mathbf{o}$  is the vector of measurements from sensors. The Kalman gain  $\mathbf{G}$  is calculated as follows:

$$\mathbf{G} = \mathbf{P}\mathbf{h}_x^\top \boldsymbol{\Sigma}^S \tag{11}$$

where  $\mathbf{P}$  is the covariance matrix of the state estimation uncertainty,  $\mathbf{h}_x$  is the Jacobian of the measurement model  $\mathbf{h}(\mathbf{x})$  with respect to the states  $\mathbf{x}$ , and  $\boldsymbol{\Sigma}^S$  is the covariance matrix of the measurement noise. The covariance matrix  $\mathbf{P}$  evolves following the next equation when the measurements are available:

$$\dot{\mathbf{P}} = \mathfrak{F}\mathbf{P} + \mathbf{P}\mathfrak{F}^\top - \mathbf{P}\mathbf{h}_x^\top \Sigma^S \mathbf{h}_x \mathbf{P} + \Sigma_c^P \tag{12}$$

being  $\Sigma_c^P$  the continuous covariance matrix of the plant noise. In the time steps in which the measurements are not available, the part related to them must be eliminated, leading to:

$$\dot{\mathbf{P}} = \mathfrak{F}\mathbf{P} + \mathbf{P}\mathfrak{F}^\top + \Sigma_c^P \tag{13}$$

Moreover, when the measurements are not available, the innovation  $\mathbf{h}(\mathbf{x}) - \mathbf{o}$  should be set to  $\mathbf{0}$  in Eqs. (9) and (10).

In order to integrate the result of the filter numerically, the implicit single-step trapezoidal rule can be used as the integration scheme:

$$\hat{\mathbf{z}}_{k+1} = \frac{2}{\Delta t} \hat{\mathbf{z}}_{k+1} - \left( \frac{2}{\Delta t} \hat{\mathbf{z}}_k + \hat{\mathbf{z}}_k \right) \tag{14}$$

$$\hat{\mathbf{z}}_{k+1} = \frac{2}{\Delta t} \hat{\mathbf{z}}_{k+1} - \left( \frac{2}{\Delta t} \hat{\mathbf{z}}_k + \hat{\mathbf{z}}_k \right) \tag{15}$$

Combining Eqs. (9) and (10) with Eqs. (14) and (15) leads to the following nonlinear system,

$$\begin{cases} \mathbf{g}_1(\hat{\mathbf{x}}_{k+1}) = \mathbf{0}_{g \times 1} \\ \mathbf{g}_2(\hat{\mathbf{x}}_{k+1}) = \mathbf{0}_{g \times 1} \end{cases} \Rightarrow \mathbf{g}(\hat{\mathbf{x}}_{k+1}) = \mathbf{0}_{2g \times 1} \tag{16}$$

This system can be iteratively solved, e.g., by means of the Newton–Raphson method, employing the following approximate Jacobian matrix:

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{2}{\Delta t} \mathbf{I}_g & & -\mathbf{I}_g \\ \mathbf{R}^\top \bar{\mathbf{K}} \mathbf{R} & \mathbf{R}^\top (\bar{\mathbf{C}} \mathbf{R} + \mathbf{M} \bar{\mathbf{R}}) & + \frac{2}{\Delta t} \bar{\mathbf{M}} \end{bmatrix} + \begin{bmatrix} \mathbf{G}^z \mathbf{h}_z & \mathbf{G}^z \mathbf{h}_z \\ \mathbf{M} \mathbf{G}^z \mathbf{h}_z & \mathbf{M} \mathbf{G}^z \mathbf{h}_z \end{bmatrix} \tag{17}$$

where the pair  $\mathbf{h}_z$  and  $\mathbf{h}_z$  are the position and velocity parts of the Jacobian of the measurement model.

The covariance matrix of the estimation error  $\mathbf{P}$  is integrated simultaneously and using the same integrator used to integrate the states. When a measurement is available at the present time step  $k$ , the time derivative of the covariance matrix is calculated using Eq. (12). If the measurement is not available at the considered time step, then Eq. (13) is employed instead.

### 3.1.2 Discrete extended Kalman filter (DEKF)

This is the discrete-time version of CEKF described above. A key difference between CEKF and the rest of estimators described from now on, which

work in discrete time steps, is that the filter formulation consists of two separated stages: state transition (also called prediction or time update) and state update (also called state correction or measurement update). The former relies on the transition model of the system (integration of dynamical equations) while the latter includes the information from sensors, or observations—this is in contrast to the CEKF, where both stages are seamlessly fused together.

The most generic form of the equations of the EKF in the prediction stage is described as:

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_k) \tag{18}$$

$$\mathbf{P}_k^- = \mathbf{f}_{\mathbf{x}k-1} \mathbf{P}_{k-1}^+ \mathbf{f}_{\mathbf{x}k-1}^\top + \Sigma^P \tag{19}$$

where  $\mathbf{f}(\cdot)$  and  $\mathbf{f}_x$  represent the transition model of the system and its Jacobian matrix with respect to the states  $\mathbf{x}$ , respectively, and  $\mathbf{u}$  is the vector of inputs. By considering now the same state vector as in Eq. (4), that is,  $\mathbf{x}^\top = [\mathbf{z}^\top, \dot{\mathbf{z}}^\top]$ , and using the Euler method as the numerical integrator for the resolution of the multibody equations, with time step  $\Delta t$ , the EKF transition model  $\mathbf{f}(\cdot)$  results:

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \mathbf{f}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_k) \\ &\Rightarrow \begin{bmatrix} \hat{\mathbf{z}}_k \\ \dot{\hat{\mathbf{z}}}_k \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{z}}_{k-1} + \Delta t \dot{\hat{\mathbf{z}}}_{k-1} \\ \dot{\hat{\mathbf{z}}}_{k-1} + \Delta t \ddot{\hat{\mathbf{z}}}_{k-1} \end{bmatrix} \end{aligned} \tag{20}$$

The input of this system is the acceleration vector  $\ddot{\mathbf{z}}_{k-1}$ , which is computed by solving the multibody equations of motion as in Eq. (3). Thus, it follows that the approximate Jacobian of the transition model  $\mathbf{f}_x$  is:

$$\begin{aligned} \mathbf{f}_x &\equiv \frac{\partial \mathbf{f}}{\partial \hat{\mathbf{x}}} = \frac{\partial}{\partial \{\hat{\mathbf{z}}, \dot{\hat{\mathbf{z}}}\}} \begin{bmatrix} \hat{\mathbf{z}} + \Delta t \dot{\hat{\mathbf{z}}} \\ \dot{\hat{\mathbf{z}}} + \Delta t \ddot{\hat{\mathbf{z}}} \end{bmatrix} \\ &\simeq \begin{bmatrix} \mathbf{I}_g & \Delta t \mathbf{I}_g \\ \mathbf{0}_{g \times g} & \mathbf{I}_g \end{bmatrix} \end{aligned} \tag{21}$$

where it is assumed that  $\frac{\partial \ddot{\mathbf{z}}}{\partial \dot{\mathbf{z}}} = \frac{\partial \ddot{\mathbf{z}}}{\partial \mathbf{z}} = \mathbf{0}$ .

Regarding the discrete plant covariance matrix  $\Sigma^P$  appearing in Eq. (19), the additional uncertainty of the new state  $\hat{\mathbf{x}}_k$  is attributed to unknown forces and parameters of the mechanism such as masses or inertia values.

The second stage of the DEKF method is the update, in which the sensor readings (when available) are incorporated to improve the estimate:

$$\mathbf{y}_k = \mathbf{o}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-) \tag{22}$$

$$\mathbf{S}_k = \mathbf{h}_{\mathbf{x}k} \mathbf{P}_k^- \mathbf{h}_{\mathbf{x}k}^\top + \Sigma_k^S \tag{23}$$

$$\mathbf{G}_k = \mathbf{P}_k^- \mathbf{h}_{\mathbf{x}k}^\top \mathbf{S}_k^{-1} \tag{24}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{G}_k \mathbf{y}_k \tag{25}$$

$$\mathbf{P}_k^+ = (\mathbf{I}_{2g} - \mathbf{G}_k \mathbf{h}_{\mathbf{x}k}) \mathbf{P}_k^- \tag{26}$$

where  $\tilde{\mathbf{y}}_k$  in Eq. (22) is the error or mismatch between the actual values and the expected sensor readings. The term  $\mathbf{S}_k$  in Eq. (23) is the innovation covariance matrix, which represents the uncertainty in the system state projected via the sensor function ( $\mathbf{h}_{\mathbf{x}k} \mathbf{P}_k^- \mathbf{h}_{\mathbf{x}k}^\top$ ) affected by the sensor Gaussian noise ( $\Sigma_k^S$ ). It is considered that the observation introduces useful information to constrain the estimation of the system state when the values of  $\mathbf{S}_k$  are small. The estimation of the mean and covariance is updated in Eqs. (25) and (26) respectively by evaluating the Kalman gain ( $\mathbf{G}_k$ ).

### 3.1.3 Unscented Kalman filter (UKF)

This method is a variant of the Kalman filters specially suitable for systems with strong nonlinearities in the transition and observation models [22]. The unscented Kalman filter is easier to implement than the conventional extended Kalman filters, since the calculation of the Jacobians of the state transition and measurement equations is not required. This method, among others, was previously applied to multibody models in [16] considering the independent accelerations of the multibody model as the states of the filter. The state vector of the UKF considered here, however, contains the independent coordinates and their velocities, that is,  $\hat{\mathbf{x}}^\top = [\hat{\mathbf{z}}^\top, \dot{\hat{\mathbf{z}}}^\top]$ .

The method comprises the same prediction and update stages than the DEKF. The differentiating feature of the UKF is the avoidance of the first order Taylor approximation in the propagation of the covariance matrix through the transition and observation functions. Instead, the unscented transformation is employed: A set of  $2l + 1$  samples  $\boldsymbol{\chi}(i)$  (usually called sigma points) is deterministically chosen from the Gaussian distributions, being  $l = 2g$  the length of the state vector:

$$\boldsymbol{\chi}_{k-1}(0) = \hat{\mathbf{x}}_{k-1}^+ \tag{27}$$

$$\boldsymbol{\chi}_{k-1}(i) = \hat{\mathbf{x}}_{k-1}^+ + \zeta \left( \sqrt{\mathbf{P}_{k-1}^+} \right)_i, \quad i = 1, \dots, l \tag{28}$$

$$\boldsymbol{\chi}_{k-1}(l+i) = \hat{\mathbf{x}}_{k-1}^+ - \zeta \left( \sqrt{\mathbf{P}_{k-1}^+} \right)_i, \quad i = 1, \dots, l \tag{29}$$

where  $\sqrt{\cdot}$  is the matrix square root using the lower triangular matrix of the Cholesky decomposition and  $(\cdot)_i$  stands for its  $i$ th column,  $\zeta = \sqrt{l + \lambda}$ ,  $\lambda = \alpha^2 (l + \kappa)$ ,  $\alpha$  and  $\kappa$  are user-defined tuning parameters, with  $0 < \alpha \leq 1$  and  $\kappa$  is usually set to 0. Then, these samples are transformed via the corresponding function (in this case, an integration step of the multi-body simulation):

$$\boldsymbol{\chi}_k(i) = \mathbf{f}(\boldsymbol{\chi}_{k-1}(i), \mathbf{u}_k(i)) \tag{30}$$

where the vector of inputs  $\mathbf{u}_k(i)$  of the  $i$ th sigma point is calculated by substituting  $\boldsymbol{\chi}_{k-1}(i)$  in Eq. (3). Both the forward Euler and the trapezoidal rule integration methods were considered in this work. Next, the mean and covariance of the resulting set are calculated:

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2l+1} W_i^m \boldsymbol{\chi}_k(i) \tag{31}$$

$$\mathbf{P}_k^- = \sum_{i=0}^{2l+1} W_i^c (\boldsymbol{\chi}_k(i) - \hat{\mathbf{x}}_k^-) (\boldsymbol{\chi}_k(i) - \hat{\mathbf{x}}_k^-)^\top + \Sigma^P \tag{32}$$

where  $W_0^m = \lambda / (l + \lambda)$ ,  $W_0^c = W_0^m + (1 - \alpha^2 + \beta)$ ,  $W_0^c = W_0^m = 1 / [2(l + \lambda)]$ , being  $\beta$  a secondary scaling factor used to emphasize the weighting on the zeroth sigma point for the covariance calculation. The selection of the weights  $W_i^m$  and  $W_i^c$  has been done according to the rules proposed in [11].

After the prediction stage, the update is accomplished. A new set of sigma points can be generated, or the existent sigma points  $\boldsymbol{\chi}_k(i)$  can be reused to save computational effort at the cost of sacrificing accuracy. This last option was used here. The measurement equation is applied to the samples, and the mean and covariance of the measurements are calculated with the same method applied in the prediction:

$$\mathbf{y}_k(i) = \mathbf{o}_k - \mathbf{h}(\boldsymbol{\chi}_k(i)) \tag{33}$$

$$\mathbf{y}_k = \sum_{i=0}^{2l+1} W_i^m \mathbf{y}_k(i) \tag{34}$$

$$\mathbf{P}_y = \sum_{i=0}^{2l+1} W_i^c \mathbf{y}_k(i) \mathbf{y}_k(i)^\top + \Sigma_k^S \tag{35}$$

Then, the Kalman gain matrix is calculated and employed to correct state and covariance matrix:

$$\mathbf{P}_{xy} = \sum_{i=0}^{2l+1} W_i^c (\boldsymbol{\chi}_k(i) - \hat{\mathbf{x}}_k^-) (-\mathcal{Y}_k(i))^T + \boldsymbol{\Sigma}_k^S \quad (36)$$

$$\mathbf{G}_k = \mathbf{P}_{xy} \mathbf{P}_y^{-1} \quad (37)$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{G}_k \mathbf{y}_k \quad (38)$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{h}_{\mathbf{x}k} \mathbf{P}_y \mathbf{h}_{\mathbf{x}k} \quad (39)$$

As shown in [22], this approach captures the correct posterior mean and covariance up to the third order of a Taylor series expansion, in contrast to the first order of DEKF and most other methods. In turn, its computational cost is in general higher than simpler methods.

### 3.2 Error-state extended Kalman filter (errorEKF)

A common problem of the previous EKF methods is that they require a tailor-made multibody algorithm, which is integrated with the Kalman filter algorithm. This problem is overcome by the UKF method, in which any multibody formulation or integrator can be used, but at the cost of running one multibody simulation for each sigma point. However, an error-state Kalman filter (also known as indirect Kalman filter) can combine the efficiency of the EKF while using the multibody simulation as a “black box.” This kind of indirect formulation is commonly used in inertial navigation and absolute position sensors fusion algorithms [17].

A simplified scheme of this method, referred to as errorEKF, is shown in Fig. 2. For each time step, this method first runs one step of the multibody simulation, obtaining the coordinates  $\mathbf{q}_{\text{MBS}}$  and velocities  $\dot{\mathbf{q}}_{\text{MBS}}$ . Next, the estimation process takes place on a state vector  $\mathbf{x}^T = [\Delta \mathbf{z}^T, \Delta \dot{\mathbf{z}}^T]$  which consists in the *error* in

position and velocity of the degrees of freedom of the mechanism, such that:

$$\mathbf{z} = \mathbf{z}_{\text{MBS}} + \Delta \mathbf{z} \quad (40)$$

$$\dot{\mathbf{z}} = \dot{\mathbf{z}}_{\text{MBS}} + \Delta \dot{\mathbf{z}} \quad (41)$$

where  $\mathbf{z}_{\text{MBS}}$  and  $\dot{\mathbf{z}}_{\text{MBS}}$  are values of  $\mathbf{z}$  and  $\dot{\mathbf{z}}$  predicted by the multibody model before the corrections are applied. The motivation of this approach is that a linearized estimator such as the EKF will suit well the problem of estimating a small displacement in the linearized neighborhood (tangent space) of the nonlinear manifold of the mechanism state space.

Accordingly, the propagation phase is performed following the next equations:

$$\hat{\mathbf{x}}_k^- = \mathbf{0} \quad (42)$$

$$\mathbf{P}_k^- = \mathbf{f}_{\mathbf{x}k-1} \mathbf{P}_{k-1}^+ \mathbf{f}_{\mathbf{x}k-1}^T + \boldsymbol{\Sigma}^P \quad (43)$$

These equations are the conventional equations for the propagation of the Kalman state, and the transition matrix  $\mathbf{f}_x$  is the same used in the DEKF method. Note that, since the errorEKF will always operate in the transformed state space of *errors*, its a priori estimate ( $\hat{\mathbf{x}}_k^-$ ) is always null, hence the zero in Eq. (42). In other words, the filter initially assumes that the multibody model made a perfect work in tracking the dynamical system at hand.

The equations for the correction phase of the filter are also similar to the ones found in the DEKF.

$$\mathbf{y}_k = \mathbf{o}_k - \mathbf{h}(\mathbf{q}_k, \dot{\mathbf{q}}_k) \quad (44)$$

$$\mathbf{S}_k = \mathbf{h}_{\mathbf{x}k} \mathbf{P}_k^- \mathbf{h}_{\mathbf{x}k}^T + \boldsymbol{\Sigma}_k^S \quad (45)$$

$$\mathbf{G}_k = \mathbf{P}_k^- \mathbf{h}_{\mathbf{x}k}^T \mathbf{S}_k^{-1} \quad (46)$$

$$\hat{\mathbf{x}}_k^+ = \mathbf{0} + \mathbf{G}_k \mathbf{y}_k \quad (47)$$

$$\mathbf{P}_k^+ = (\mathbf{I}_{2g} - \mathbf{G}_k \mathbf{h}_{\mathbf{x}k}) \mathbf{P}_k^- \quad (48)$$

The differences can be found in Eq. (22), which becomes Eq. (44), where the virtual measurements  $\mathbf{h}(\mathbf{q}_k, \dot{\mathbf{q}}_k)$  are built by using the coordinates of the multibody model instead of the states of the filter. The Jacobian of the measurement model  $\mathbf{h}_x$  has the same expression as in the DEKF, since the partial derivatives with respect to the errors in the states have the same value than the partial derivatives with respect to the states. Before the measurements are applied, the state is always null, so Eq. (25) is modified accordingly, becoming Eq. (47).

After the correction stage, the estimation of the position and velocity errors of the independent coordinates

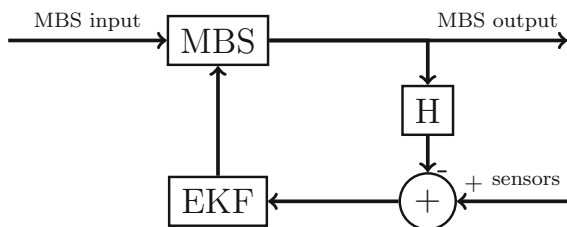


Fig. 2 Simplified diagram of the error-state Kalman filter applied to multibody simulations



are obtained. However, to correct the state of the multi-body system, the errors for all the coordinates must be obtained, so they must be projected over the constraints manifold.

An error in position means that an increment to the coordinates should be applied to get the position corrected. Such an increment must fulfill the velocity constraints:

$$\Phi_{\mathbf{q}} \Delta \mathbf{q} = \mathbf{0} \quad (49)$$

Therefore, the increments  $\Delta \mathbf{q}$  applied to the coordinates of the mechanism are calculated by solving the velocity problem [9]. If the Jacobian of the constraints  $\Phi_{\mathbf{q}}$  and the vector of position errors are split in their independent and dependent parts, Eq. (49) can be written as follows:

$$\Delta \mathbf{q} = [\Delta \mathbf{q}_{\text{INDEP}}^T \ \Delta \mathbf{q}_{\text{DEP}}^T]^T = [\Delta \mathbf{z}^T \ \Delta \mathbf{q}_{\text{DEP}}^T]^T \quad (50)$$

$$\Phi_{\mathbf{q}_{\text{DEP}}} \Delta \mathbf{q}_{\text{DEP}} = -\Phi_{\mathbf{q}_{\text{INDEP}}} \Delta \mathbf{z} \quad (51)$$

The estimated error of the degrees of freedom  $\Delta \hat{\mathbf{z}}$  is substituted into Eq. (51), and the resulting linear system is solved to obtain the estimated dependent errors  $\Delta \hat{\mathbf{q}}_{\text{DEP}}$ . Then, the resulting vector of estimated position errors  $\Delta \hat{\mathbf{q}}$  is fed back to the multibody simulation to obtain the position estimation  $\hat{\mathbf{q}}$ , as follows:

$$\hat{\mathbf{q}} = \mathbf{q}_{\text{MBS}} + \Delta \hat{\mathbf{q}} \quad (52)$$

This method is an approximation, and thus, a perfect fulfillment of the constraints at position level is not expected. However, as the corrections are performed every time step, the errors are usually acceptable for most applications. The main advantage of this method is that the velocity problem is linear, so this process is much faster than solving the position problem.

The correction of the velocity estimation is performed after the correction of the position is applied by solving other velocity problem, as follows:

$$\Phi_{\mathbf{q}_{\text{DEP}}} \hat{\mathbf{q}}_{\text{DEP}} = -\Phi_{\mathbf{q}_{\text{INDEP}}} [\dot{\mathbf{z}}_{\text{MBS}} + \Delta \hat{\dot{\mathbf{z}}}] \quad (53)$$

Therefore, the vector of estimated velocities yields:

$$\hat{\dot{\mathbf{q}}} = \left[ (\dot{\mathbf{z}}_{\text{MBS}} + \Delta \hat{\dot{\mathbf{z}}})^T \ \hat{\dot{\mathbf{q}}}_{\text{DEP}}^T \right]^T \quad (54)$$

Once the corrections are applied to the model, the expected error is  $\hat{\mathbf{x}}_k^+ = \mathbf{0}$ .

#### 4 Covariance matrices of plant and measurement noise

It is known that, when applying Kalman filters, the tuning of the parameters of the algorithm (covariance matrices of plant and measurement noise) is paramount. When dealing with nonlinear systems, even if everything else is correct in the algorithm, it can become unstable if the covariance matrices of the noise are not properly set.

In this work, the signals from the sensors are built from a multibody model, playing the role of actual mechanism, and then white Gaussian noise is generated and added to the sensor signal. Hence its properties are perfectly known.

The plant noise properties, however, are not known, since the errors introduced in the simulation are not additive white noise, but errors in the force models which produce deviations from the ideal behavior. In a real mechanism is even worst, because it is not known where the errors are, or it is not possible to correct them for any reason.

When the ground truth is not known, the only way to check that the filter is working properly is by checking the innovation sequence. It should behave like white noise. Although this criterion does not allow to find the correct absolute value of the covariances, it allows to find the correct relation between plant and measurement covariance matrices.

Despite that this work relies on simulation, meaning that the ground truth is available, the power spectral density of the innovation was checked to lead the process of getting the adequate covariance of plant noise.

The plant noise used for the test was different for the different modeling errors applied and for the different mechanisms. While considering specific mechanism and level of modeling error, all the discrete-time methods obtained the best results with the same plant noises, adjusted to produce an innovation sequence which behaves like white noise when the initial position of the mechanism is perfectly known and position sensors are employed at the same rate of the multibody simulation. However, smaller values of plant noise and initial covariance matrix of state estimation uncertainty  $\mathbf{P}_0$  had to be used in the CEKF to avoid the covariance matrix  $\mathbf{P}$  becoming negative definite.

#### 4.1 Structure of plant noise

When dealing with multibody models, usually the geometry is known accurately enough to model it properly. However, getting precise models of the forces and obtaining the actual distribution of mass is usually more complex in practice. Both these effects appear as errors in the accelerations. Then, the integration process and the multibody formulation may introduce more errors, but they are usually negligible compared to the previous ones. For this reason, only the acceleration terms are considered in the plant noise.

In the CEKF, which is the only method in continuous time analyzed in this work, the acceleration noise is straightforwardly introduced:

$$\Sigma_c^P = \begin{bmatrix} \mathbf{0}_{g \times g} & \mathbf{0}_{g \times g} \\ \mathbf{0}_{g \times g} & \sigma_z^2 \mathbf{I}_g \end{bmatrix} \quad (55)$$

However, in the discrete-time methods, the matrix of the covariance of the plant noise must be calculated from its continuous counterpart by integration:

$$\Sigma^P = \mathbf{f}_x(t_k, t_{k-1}) \mathfrak{J} \mathbf{f}_x^\top(t_k, t_{k-1}) \quad (56)$$

$$\mathfrak{J} = \int_{t_{k-1}}^{t_k} \mathbf{f}_x^{-1}(\tau, t_{k-1}) \Sigma_c^P \mathbf{f}_x^{-\top}(\tau, t_{k-1}) d\tau \quad (57)$$

This integration can be done using Van Loan's method [21], obtaining a covariance matrix of process noise with the following structure:

$$\Sigma^P = \begin{bmatrix} \sigma_z^2 \frac{\Delta t^3}{3} \mathbf{I}_g & \mathbf{0} \\ \mathbf{0} & \sigma_z^2 \frac{\Delta t^2}{2} \mathbf{I}_g \end{bmatrix} \quad (58)$$

## 5 Tests and results

Two different mechanisms, with two levels of modeling error, and with three different sets of sensors have been considered in this work. Different sampling rates for the sensors were considered in every configuration. The discrete-time methods were found to be more robust and accurate than the CEKF, but the latter is maintained here as a reference because it has been employed in several works previously (e.g., [5, 15]).

The sensors considered in the tests are gyroscopes (angular rate sensors) and encoders. Their measurements were built from a multibody simulation of the mechanism, and then a sequence of pseudo-random noise with a normal distribution with mean 0 and standard deviation of  $\pi/180$  rad was added. The units of

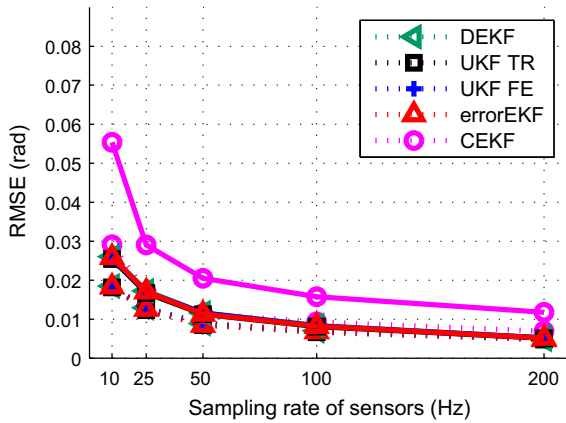
the noise are rad/s for the gyroscopes and rad for the encoders. The same sequence of pseudo-random noise was used for all the tests for a fair comparison. Slightly different results are expected with other sequences of noise. However, the relative behavior of the methods is the same with different sequences of noise.

The tests consist in the mechanisms starting from rest and falling under the action of their own weight for ten seconds. The errors introduced intentionally in the model are the value of the acceleration of gravity, representing an error in a force model which affects during all the simulation, and an error in the initial position of the mechanism. The initial position error is specially useful to show that the methods based only on gyroscopes can provide, in some configurations, accurate estimations of the position, which is not usually reported in the bibliography. Two levels of modeling error were considered:  $0.5 \text{ m/s}^2$  with an initial error of  $\pi/32$  rad, and  $1 \text{ m/s}^2$  with an initial error of  $\pi/16$  rad.

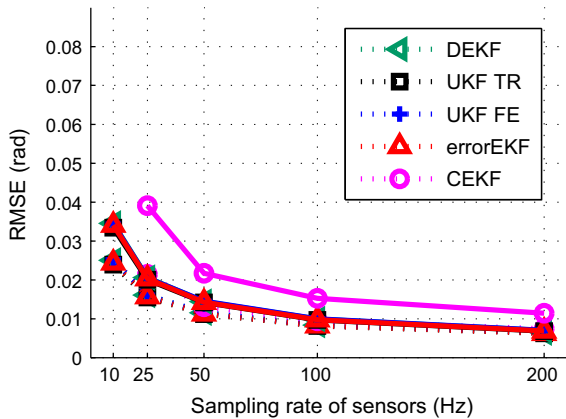
#### 5.1 Tests with position sensors

In these tests, we assume that encoders are installed on the cranks of the mechanisms. Encoders are usually the preferred option to measure rotational motion. They can provide accurate measurements at a high sampling rate. For this reason, if enough encoders are available, a state observer is not usually needed. However, not all the position sensors have such a high sampling rate or accuracy. Thus, in this case the encoders are employed to verify the behavior of the state observers when generic position sensors are available, but considering that they can have a slower sampling rate than actual encoders, because they could be other kinds of position sensors, such as a GPS, laser or ultrasonic distance sensors.

The results for the four-bar and five-bar linkages are shown in Figs. 3 and 4, respectively, where no significant differences can be observed. The worst accuracy is obtained from the CEKF. In addition, it is the least robust when the measurements are not available at every time step, being the only method which is not able to handle all the sampling rates tested in the case of the five-bar mechanism. All the other methods considered, namely DEKF, UKF with trapezoidal rule integrator (UKF TR), UKF with forward Euler integrator (UKF FE), and errorEKF provide the same accuracy. Moreover, they provide the same level of accuracy for both



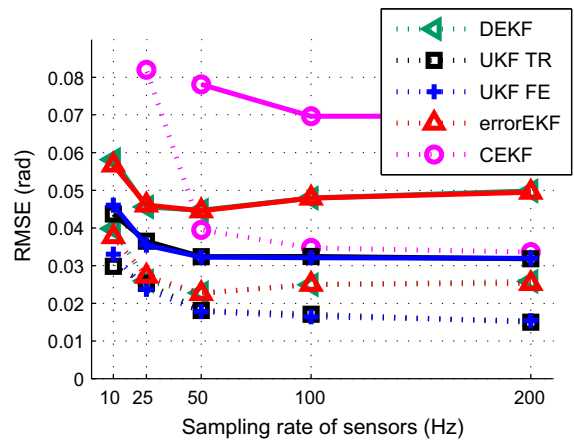
**Fig. 3** RMS error of the crank angle in the four-bar linkage with an encoder on the crank. *Dashed lines* represent the low modeling error test, while *solid lines* represent the high modeling error



**Fig. 4** RMS error of cranks angles in the five-bar linkage with encoders on the cranks. *Dashed lines* represent the low modeling error test, while *solid lines* represent the high modeling error. The CEKF method cannot handle all the sampling rates of the sensors tested with the other methods

levels of modeling error if the sampling rate of the measurements is high enough, but when the sensors have a slower sampling rate, the test with low modeling error provides better results. It is also interesting to remark that the RMS error of the sensor is  $\pi/180 \approx 0.0175$  rad. All the discrete-time methods get a smaller error if the sampling rate of the sensor is at least 50 Hz.

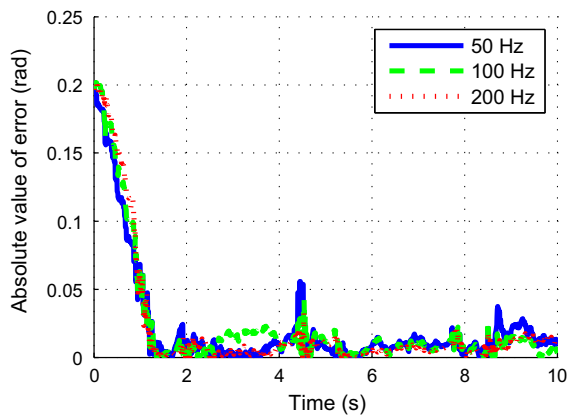
With this sensor configuration, the measurement model is linear. This is the reason why the methods based on the extended Kalman filter can get approximately the same accuracy level than the unscented Kalman filters.



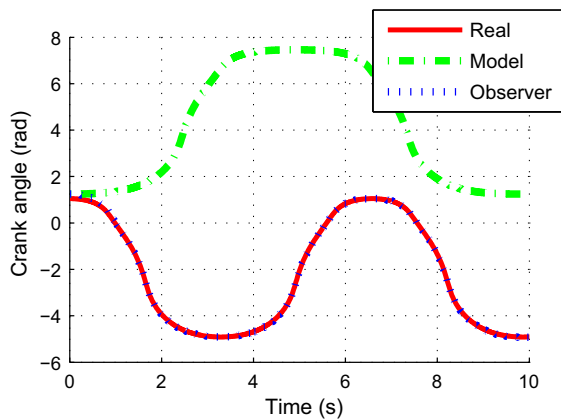
**Fig. 5** RMS error of the crank angle in the four-bar linkage with an angular rate sensor on the coupler link. *Dashed lines* represent the low modeling error test, while *solid lines* represent the high modeling error

### 5.2 Tests with velocity sensors

Two different configurations using only velocity sensors are considered in each mechanism, leading to different situations. In the first configuration, the angular rate sensors (gyroscopes) are fixed to the coupler link of the four-bar mechanism, and to both coupler links in the five-bar linkage. Since both mechanisms present a closed-loop topology, the velocity and the position of such elements are related, and hence the absolute position of the mechanism can be inferred from velocity measurements, even if the initial position is not accurately known. Here, both the geometry of the mechanism and the characteristics of the motion affect the accuracy of the state observer. The results for the four-bar linkage are shown in Fig. 5. Now the UKF methods show the best accuracy, independently of the type of integrator used. The next methods in terms of performance are the DEKF and the errorEKF. It seems that their performance degrades with higher sampling rates. However, this is not true. Indeed, at the beginning of the simulations the system is not observable with this sensor configuration, because the mechanism starts from rest. When the simulation starts, the first measurements are not coherent because of the error in the initial position, and this produces that the first measurements in this particular test take the state observer farther from the true solution instead of closer. For this reason, the test with the highest measurement frequency has to recover a greater error from the beginning. The abso-

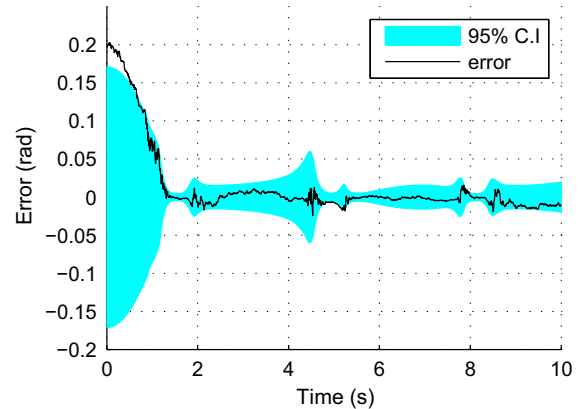


**Fig. 6** Absolute value of the error of the crank angle in the four-bar linkage with an angular rate sensor on the coupler link, with sampling rates of 200, 100, and 50 Hz. These tests were performed with the DEKF with  $\pi/16$  initial error, and  $1 \text{ m/s}^2$  of error in gravity acceleration



**Fig. 7** Angle of the crank of the four-bar mechanism for the real mechanism, the model without corrections, and the state observer with an angular rate sensor on the coupler link. This test was performed with the DEKF formulation, 200 Hz sampling rate for the sensor,  $\pi/16$  initial error, and  $1 \text{ m/s}^2$  of error in gravity acceleration

lute value of the error considering sampling rates of 50, 100, and 200 Hz is shown in Fig. 6 to illustrate this phenomenon. A plot of the crank angle is shown in Fig. 7, and its error during the whole test is displayed in Fig. 8, using the DEKF formulation with a sampling rate of 200 Hz. The confidence interval represented in Fig. 8 is calculated from the values of the covariance matrix of the state estimation uncertainty  $\mathbf{P}$ . The values of the diagonal of this matrix contain the variance of the correspondent states, from where the confidence interval can be calculated assuming a normal distribu-



**Fig. 8** Error of the crank angle in the four-bar linkage with an angular rate sensor on the coupler link, and its 95% CI. This test was performed with the DEKF formulation, 200 Hz sampling rate for the sensor,  $\pi/16$  initial error, and  $1 \text{ m/s}^2$  of error in gravity acceleration

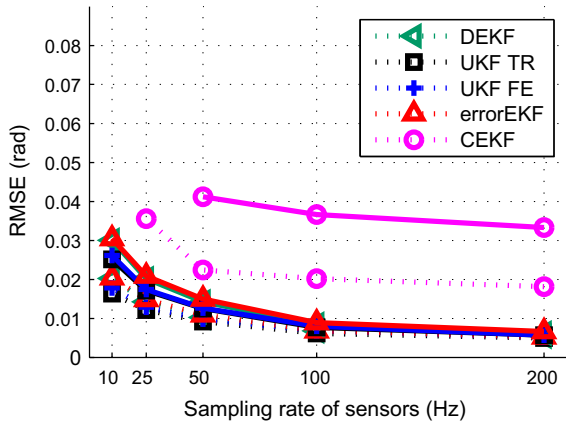
tion [13]. It can be seen how the confidence interval expands and shrinks depending on the motion of the mechanism, since it depends on both the velocity of the coupler bar and the relation between this velocity and the position of the crank.

The CEKF provided the worst results and showed also that it is the least robust, since it was again the only method which could not handle all the sampling rates.

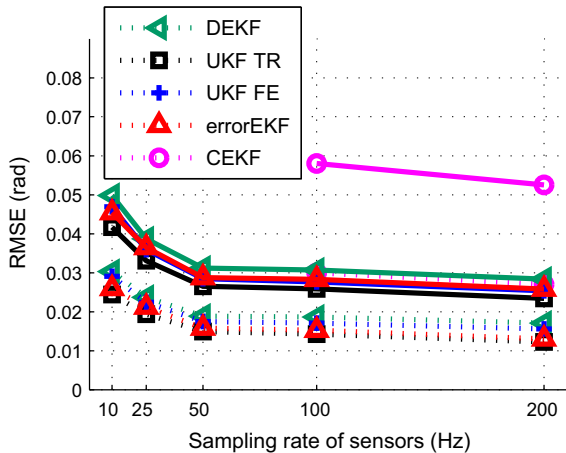
The same test has been repeated, but starting with the mechanism in motion, with an initial velocity of 5 rad/s in the crank. In this case, the system is observable from the beginning of the maneuver. The results, shown in Fig. 9, are clearly better than when starting from rest, and all the methods improved their results when the sampling rate of the measurements was increased.

The results from the tests with the five-bar mechanism are shown in Fig. 10. Again, the UKF methods show the best accuracy. However, in this test the errorEKF is closer to them than the DEKF. The CEKF obtained the worst results and robustness.

If this test is repeated with initial velocity (5 rad/s in each crank of the mechanism), the results shown in Fig. 11 are obtained. In this case, due to the geometry of the mechanism, it is more difficult to integrate when the velocity is increased. For this reason, the methods using the trapezoidal rule integration (UKF TR and errorEKF) provide better results than the methods which employ the forward Euler integration (UKF FE and DEKF). The CEKF still provides the worst



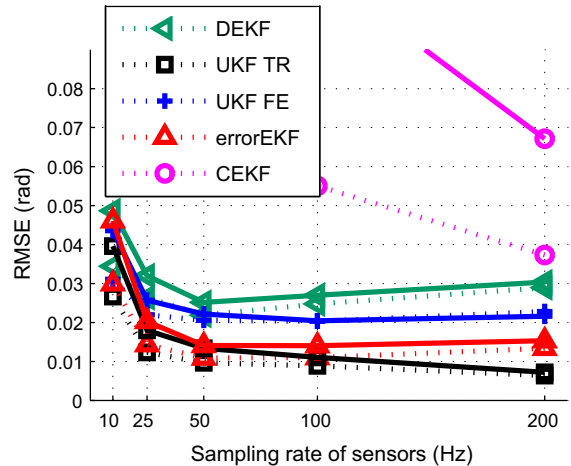
**Fig. 9** RMS error of the crank angle in the four-bar linkage with an angular rate sensor on the coupler link. Initial velocity of the crank of the mechanism is 5 rad/s. *Dashed lines* represent the low modeling error test, while *solid lines* represent the high modeling error



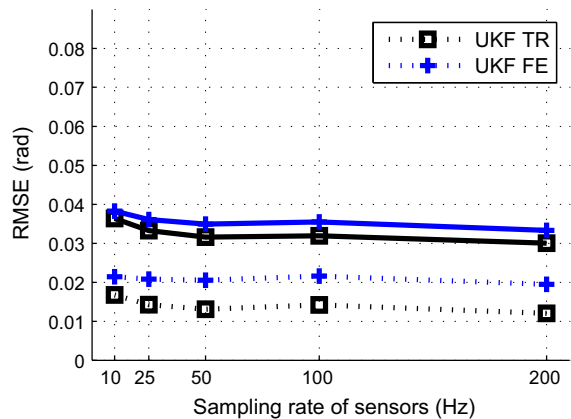
**Fig. 10** RMS error of the cranks angles of the five-bar linkage with angular rate sensors on the intermediate links. *Dashed lines* represent the low modeling error test, while *solid lines* represent the high modeling error

results, even using the trapezoidal rule as the integration scheme.

The second configuration tested with velocity sensors consists in two gyroscopes installed on the cranks of the mechanisms. This problem is more challenging than the previous one because the cranks can have any velocity at any position, and hence, there is no relationship between position and velocity. However, if the accelerations depend on the positions, the UKF methods can still provide information of the position from velocity measurements. The EKF methods cannot deal

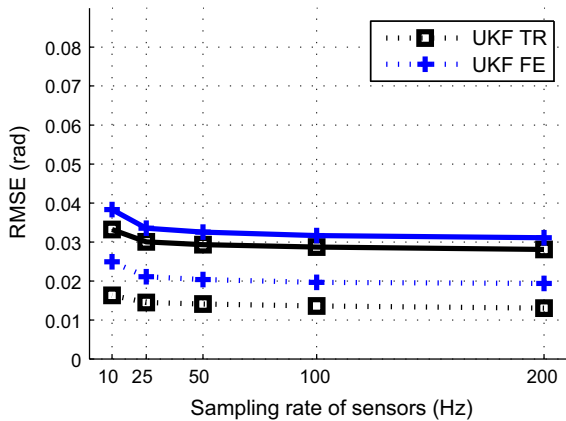


**Fig. 11** RMS error of the cranks angles of the five-bar linkage with angular rate sensors on the intermediate links. Initial velocity of the cranks was 5 rad/s. *Dashed lines* represent the low modeling error test, while *solid lines* represent the high modeling error

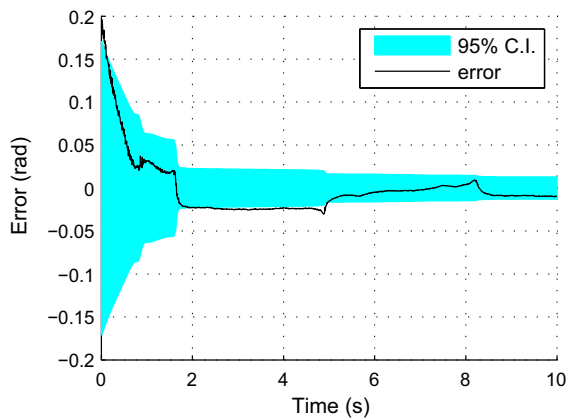


**Fig. 12** RMS error of the crank angle in the four-bar linkage with an angular rate sensor on the crank. *Dashed lines* represent the low modeling error test, while *solid lines* represent the high modeling error test

with this problem, so they are not considered here. In this case, the acceleration varies with the position of the cranks because the only actuating force is the gravity. The results are similar for both mechanisms, as shown in Fig. 12 for the four-bar linkage and in Fig. 13 for the five-bar linkage. In this configuration, the type of integrator considerably determines the accuracy of the methods, providing better results using the trapezoidal rule. This difference increases when the errors in the plant are smaller, because if the errors in force model



**Fig. 13** RMS error of the crank angles of the five-bar linkage with angular rate sensors on the cranks. *Dashed lines* represent the low modeling error test, while *solid lines* represent the high modeling error test



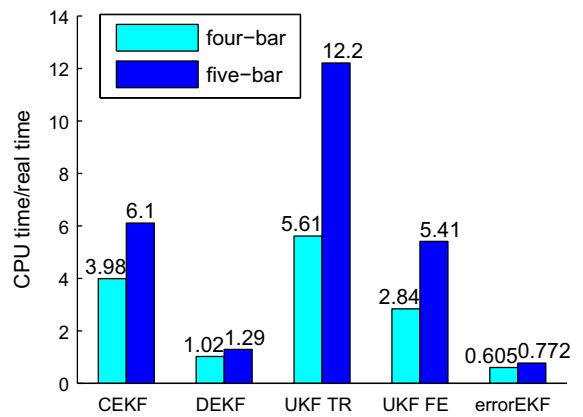
**Fig. 14** Error of the crank angle in the four-bar linkage with an angular rate sensor on the crank, and its 95% CI. This test was performed with 200 Hz sampling rate for the sensor,  $\pi/16$  initial error, and  $1 \text{ m/s}^2$  of error in gravity acceleration

are high enough, the integration error becomes negligible.

One of the tests, performed with the UKF method with the forward Euler integrator,  $1 \text{ m/s}^2$  of error in the acceleration of gravity and  $\pi/16$  rad of initial position error is shown in Fig. 14.

### 5.3 Computational cost

For a state observer to be useful, it must be run in real time. However, the tests presented here were run in MATLAB®, and the design of the software



**Fig. 15** Real-time factor of the different methods running with position sensors at 200 Hz

was intended to provide flexibility instead of efficiency, not being appropriated to be used as practical state observers. Nevertheless, the computational cost of these methods is evaluated to provide guidance in order to select a state observer for future applications. Every test takes a different time to be run, depending on the number and type of sensors, their sampling frequency, etc. However, as a comparison among the different formulations is intended here, only the results from the tests with position sensors at 200 Hz are depicted in Fig. 15. It can be seen that the two discrete-time extended Kalman filters are the fastest methods, being the errorEKF the only one which runs faster than real time. The CEKF method is much slower than its discrete counterparts. As for the unscented Kalman filters, the formulation employing the forward Euler integrator is about twice faster than the same formulation using the trapezoidal rule. The UKF with forward Euler integrator is even faster than the CEKF for the problems considered in this work. However, its computational cost grows faster as the number of degrees of freedom of the mechanism increases, so it is expected that it would be slower than the CEKF for problems of a higher number of degrees of freedom.

### 5.4 Observability analysis

The selection of the sensors needed to achieve the estimation of the desired magnitudes is a problem to be addressed when a state observer is designed. The study of the observability provides guidance in order to select the necessary sensors during the design process.

The state of a system is said to be observable if it is uniquely determined by the system model, its inputs and its outputs [7]. From the results obtained in the previous tests, the most interesting methods to be studied are the DEKF and the errorEKF for their efficiency, and the UKFs due to their accuracy. Therefore, only discrete-time methods are considered in this section.

The extended Kalman filters used here are based on nonlinear models. However, due to the discretization scheme employed (the forward Euler integrator), the resulting transition matrices are constant, and the nonlinearities of the transition models are only revealed in the calculation of the accelerations.

The sensor models, however, can be linear or not, depending on the kinds of sensors installed and their configuration. For example, the tests performed with the four-bar mechanism considering an encoder or a gyroscope on the first link lead to linear measurement models, while the test with a gyroscope on the coupler has a nonlinear measurement model.

As some of the cases have time-varying matrices, the concept of local observability is used here, determined by the rank of the following matrix [4]:

$$\mathbf{M}_{I_0} = \begin{bmatrix} \mathbf{h}_{xk} \\ \mathbf{h}_{xk+1} \mathbf{f}_{xk} \\ \mathbf{h}_{xk+2} \mathbf{f}_{xk+1} \mathbf{f}_{xk} \\ \vdots \\ \mathbf{h}_{xk+l-1} \mathbf{f}_{xk+l-2} \cdots \mathbf{f}_{xk} \end{bmatrix} \quad (59)$$

If  $\mathbf{M}_{I_0}$  is full rank, the system is observable at a given time spot. However, errors in the algorithm (not achieving a perfect fulfillment of the constraints, numerical errors, etc.) can make  $\mathbf{M}_{I_0}$  full rank when the system is actually non-observable. Moreover, the observability of the system can be weak, or progressively become non-observable in some conditions. For these reasons, instead of the rank of the matrix, the condition number of the matrix,  $\kappa(\mathbf{M}_{I_0})$  is used [7], since it is a quantitative assessment of the observability. The condition number is calculated as the relation between the maximum and the minimum singular value of  $\mathbf{M}_{I_0}$ :

$$\kappa(\mathbf{M}_{I_0}) = \frac{\sigma_{\max}}{\sigma_{\min}} \quad (60)$$

Since  $\kappa(\mathbf{M}_{I_0})$  can reach very high values, its logarithm is the magnitude to be studied. The smaller this magnitude is, the better observability the system has.

When using sampled methods, such as the UKF, the matrices  $\mathbf{f}_x$  and  $\mathbf{h}_x$  used to build  $\mathbf{M}_{I_0}$  are not available.

However, the UKF can be seen as a particular case of the linear regression Kalman filter [10], in which the matrices  $\mathbf{f}_x$  and  $\mathbf{h}_x$  are obtained as a statistical linear regression of several points propagated through the transition and measurement functions. Once these matrices are obtained from the transition and measurement updates of the UKF, they can be used to build the local observability matrix [8].

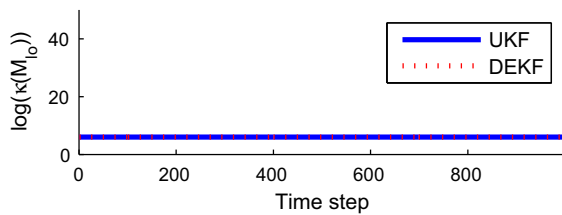
Using this method, the observability of the EKF with constant matrices can be examined before implementing the state observer. In the tests presented in this work with the four-bar mechanism, the case with the encoder and the case with the angular rate sensor on the crank can be analyzed this way. With the encoder, the system is always observable. However, if a gyroscope is used in the first link, the system is never observable.

Unfortunately, when the EKF methods with nonlinear measurement models or the UKF algorithms are employed, the matrices needed to perform the observability analysis are not available beforehand. Consequently, a simulation of the method under the expected working conditions has to be performed, limiting the interest of this observability analysis. However, the observability analysis is still useful to detect conditions in which the observability is weaker or temporarily lost under some working conditions.

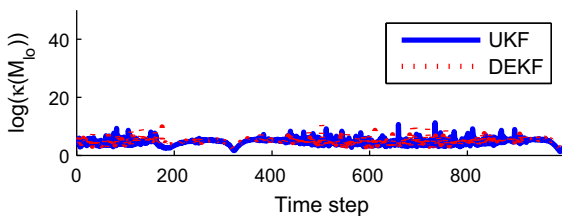
The observability of the errorEKF, DEKF and UKF was examined during a test with each one of the configurations tested. The results obtained from the DEKF and the errorEKF were similar, being observable with the encoders on the cranks, and with the gyroscopes on the coupler bars, and non-observable when the gyroscopes were installed on the cranks, both for the four-bar and the five-bar linkages. When the UKF is used, the system is observable with all the configurations tested. The results from the observability study of the four-bar mechanism are shown in Figs. 16, 17 and 18. As the behavior of the errorEKF and the DEKF is similar, only the results from the DEKF are shown for the sake of clarity.

## 6 Conclusions

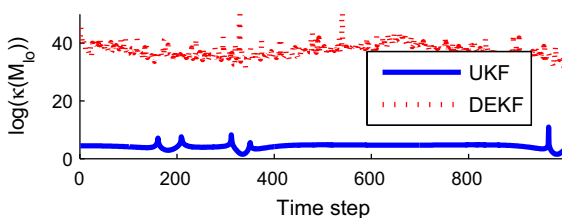
In this work, several state observers based on multibody models have been evaluated. The CEKF, DEKF and UKF methods were already evaluated in other papers, while the errorEKF is new in the multibody literature. The first three methods considered here are formulated



**Fig. 16** Result from the observability analysis of the test with the four-bar mechanism with an encoder on the crank



**Fig. 17** Result from the observability analysis of the test with the four-bar mechanism with a gyroscope on the coupler



**Fig. 18** Result from the observability analysis of the test with the four-bar mechanism with a gyroscope on the crank

in independent coordinates, namely the continuous extended Kalman filter (CEKF), the discrete extended Kalman filter (DEKF), and the unscented Kalman filter (UKF), which has been tested with two different integrators: forward Euler and the trapezoidal rule. The last method described is an error-state extended Kalman filter (errorEKF), in which the position and velocity errors of the multibody simulation are estimated and then fed back to the simulation. The methodology employed to tune the matrices of covariance of measurement and plant noise, and the shape of those matrices were also indicated.

All the presented methods have been tested in two mechanisms (four-bar and five-bar linkages), considering three different sensor configurations in each mechanism and 5 different sampling rates for the sensors. All the discrete-time methods were able to handle all the sampling rates considered, but the CEKF was not robust enough to deal with the lowest sampling rates,

and its accuracy degraded faster as the sampling rate decreased.

The tests with position sensors demonstrate that all the state observers improve the accuracy of the sensors, even if considering a sampling rate 4 times slower than the original sensor. The tests with velocity sensors demonstrate that, under certain circumstances, position information can be obtained from velocity sensors using a combination of a multibody model and an Kalman filter, even if the initial position assumed by the observer is wrong. Two different configurations were conducted in each mechanism with velocity sensors. In one of them, with the sensors on the couplers, the relation between the crank angle and the velocity measured by the sensors is time varying. In these tests, all the methods tested could provide position information. In the other configuration tested, the velocity sensors were installed on the cranks of the mechanisms. In this circumstance, the position of the crank and the velocity measured by the sensor are not related, and the methods based on the extended Kalman filter could not get position information from this sensor configuration. However, as the effect of the gravity varies with the position of the mechanism, the UKF could still provide position information in these conditions with both integrators tested.

The computational cost of the methods was also compared, being the errorEKF the fastest method, and also one of the less affected by the increase in size of the system from the four-bar to the five-bar linkage. The DEKF was the second one in efficiency. The UKF with trapezoidal rule integrator provided the best results in almost every test, but it was the slowest method. By using the forward Euler integrator, this method was executed about twice faster even when its accuracy was not significantly decreased in most of the tests. However, both methods are significantly slower in the five-bar mechanism than in the four-bar mechanism. The CEKF, although is less affected by the size of the problem than the UKF with forward Euler integrator, was slower than the latter in the tests performed in this work.

All the methods and multibody models tested here have been implemented in MATLAB<sup>®</sup> and made available as Open Source.<sup>2</sup>

<sup>2</sup> See <https://github.com/MBDS/mbde-matlab>.



**Acknowledgements** This work has been partially financed by the Spanish Ministry of Economy and Competitiveness and EU-ERDF funds under the projects ‘Observadores de estados y entradas basados en modelos multicuerpo detallados aplicados al control de vehículos’ (TRA2014-59435-P) and ‘ENERPRO’ (DPI2014-56364-C2-1-R). The research of Emilio Sanjurjo was funded by fellowship BES-2013-063598 of Spanish Government.

## References

- Aripin, M.K., Md Sam, Y., Danapalasingam, K.A., Peng, K., Hamzah, N., Ismail, M.F.: A review of active yaw control system for vehicle handling and stability enhancement. *Int. J. Veh. Technol.* **2014**, 1–15 (2014)
- Bishop, C.: *Pattern Recognition and Machine Learning*. Springer, New York (2006)
- Blanco-Claraco, J.L., Torres-Moreno, J.L., Giménez-Fernández, A.: Multibody dynamic systems as Bayesian networks: applications to robust state estimation of mechanisms. *Multibody Syst. Dyn.* **34**(2), 103–128 (2015)
- Chen, Z.: Local observability and its application to multiple measurement estimation. *IEEE Trans. Ind. Electron.* **38**(6), 491–496 (1991). doi:[10.1109/41.107106](https://doi.org/10.1109/41.107106)
- Cuadrado, J., Dopico, D., Barreiro, A., Delgado, E.: Real-time state observers based on multibody models and the extended Kalman filter. *J. Mech. Sci. Technol.* **23**(4), 894–900 (2009)
- Cuadrado, J., Dopico, D., Perez, J.A., Pastorino, R.: Automotive observers based on multibody models and the extended Kalman filter. *Multibody Syst. Dyn.* **27**(1), 3–19 (2012)
- Grewal, M., Andrews, A.: *Kalman Filtering: Theory and Practice Using MATLAB®*. Wiley, New York (2008)
- Huang, G.P., Mourikis, A.I., Roumeliotis, S.I.: A quadratic-complexity observability-constrained unscented Kalman filter for SLAM. *IEEE Trans. Robot.* **29**(5), 1226–1243 (2013)
- García de Jalón, J., Bayo, E.: *Kinematic and Dynamic Simulation of Multibody Systems: The Real Time Challenge*. Springer, New York (1994)
- Lefebvre, T., Bruyninckx, H., De Schuller, J.: Comment on “A new method for the nonlinear transformation of means and covariances in filters and estimators”. *IEEE Trans. Autom. Control* **47**(8), 1406–1409 (2002)
- Merwe, R.V.D., Wan, E.A., Julier, S.I.: Nonlinear estimation and sensor-fusion: applications to integrated navigation. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Providence, Rhode Island (2004)
- Naets, F., Pastorino, R., Cuadrado, J., Desmet, W.: Online state and input force estimation for multibody models employing extended kalman filtering. *Multibody Syst. Dyn.* **32**, 317–336 (2014)
- Navidi, W.: *Statistics for Engineers and Scientists*, 4th edn. McGraw-Hill Education, New York (2015)
- Palomba, I., Richiedei, D., Trevisani, A.: Kinematic state estimation for rigid-link multibody systems by means of nonlinear constraint equations. *Multibody Syst. Dyn.* 1–22 (2016). doi:[10.1007/s11044-016-9515-x](https://doi.org/10.1007/s11044-016-9515-x)
- Pastorino, R., Richiedei, D., Cuadrado, J., Trevisani, A.: State estimation using multibody models and non-linear Kalman filter. *Int. J. Non-Linear Mech.* **53**, 83–90 (2013)
- Pastorino, R., Richiedei, D., Cuadrado, J., Trevisani, A.: State estimation using multibody models and non-linear Kalman filters. *Int. J. Non-Linear Mech.* **53**, 83–90 (2013)
- Roumeliotis, S., Sukhatme, G., Bekey, G.: Circumventing dynamic modeling: evaluation of the error-state Kalman filter applied to mobile robot localization. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1656–1663. IEEE (1999)
- Simon, D.: *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley, Hoboken (2006)
- Torres, J., Blanco, J., Sanjurjo, E., Naya, M., Giménez, A.: Towards benchmarking of state estimators for multibody dynamics. In: *The 3rd Joint International Conference on Multibody System Dynamics. The 7th Asian Conference on Multibody Dynamics*, pp. 261–262. Busan, Korea (2014)
- Torres-Moreno, J.L., Blanco-Claraco, J.L., Giménez-Fernández, A., Sanjurjo, E., Naya, M.Á.: Online kinematic and dynamic-state estimation for constrained multibody systems based on IMUs. *Sensors* **16**(3), 333 (2016)
- Van Loan, C.F.: Computing integrals involving the matrix exponential. *IEEE trans. Autom. Control* **23**(3), 395–404 (1978)
- Wan, E.A., Merwe, R.V.D.: The unscented Kalman filter for nonlinear estimation. In: *Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158 (2000)

## Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”). Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval, sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

[onlineservice@springernature.com](mailto:onlineservice@springernature.com)