REPOSITORIO UNIVERSIDADE CORUÑA

UNIVERSIDADE DA CORUÑA

**How to cite:**
E. Sanjurjo, D. Dopico, A. Luaces, M.Á. Naya, State and force observers based on multibody models and the indirect Kalman filter, Mechanical Systems and Signal Processing 106 (2018) 210–228. https://doi.org/10.1016/j.ymssp.2017.12.041.

© 2018. This manuscript version is made available under the CC-BY-NC-ND 4.0 license
https://creativecommons.org/licenses/by-nc-nd/4.0/

# State and force observers based on multibody models and the indirect Kalman filter

Emilio Sanjurjo [1], Daniel Dopico, Alberto Luaces, Miguel Ángel Naya

*Mechanical Engineering Laboratory, University of A Coruña, Escuela Politécnica Superior, Mendizábal s/n, Ferrol 15403, Spain*

**Abstract**

The aim of this work is to present two new methods to provide state observers by combining multibody simulations with indirect extended Kalman filters. One of the methods presented provides also input force estimation. The observers have been applied to two mechanism with four different sensor configurations, and compared to other multibody-based observers found in the literature to evaluate their behavior, namely, the unscented Kalman filter (UKF), and the indirect extended Kalman filter with simplified Jacobians (errorEKF).

The new methods have some more computational cost than the errorEKF, but still much less than the UKF. Regarding their accuracy, both are better than the errorEKF. The method with input force estimation outperforms also the UKF, while the method without force estimation achieves results almost identical to those of the UKF.

All the methods have been implemented as a reusable MATLAB® toolkit which has been released as Open Source in `https://github.com/MBDS/mbde-matlab`.

*Keywords:* Multibody dynamics, Kalman filter, State observer, Force estimation

## 1. Introduction

Multibody dynamics [1, 2, 3, 4] has become a standard tool to aid the development of new products, reducing development time and cost. Multibody modules are usually available within all the main CAD-CAE software suites, thus allowing to perform offline simulations. However, the increase in computational power and the improvement of multibody simulations allowed their execution in real time, which have been used both in human-in-the-loop (e.g. [5]) and hardware-in-the-loop applications (e.g. [6]), even on inexpensive single-board computers [7]. These new applications allowed the usage of multibody models in driving simulators, virtual reality applications, controllers development, and state observers, among others.

[1] Corresponding author, emilio.sanjurjo@udc.es

© 2018. This manuscript version is made available under the CC-BY-NC-ND 4.0 license
https://creativecommons.org/licenses/by-nc-nd/4.0/

During the last decade, the research on state observers based on multibody dynamics has been quite intense, usually in combination with nonlinear Kalman filters [8, 9], although other estimation techniques have been successfully applied to multibody models, such as particle filtering [10]. This research line started with a continuous extended Kalman filter combined with the equations of multibody dynamics in [11], and continued with the application of this method to the model of an automobile [12]. However, this method had a high computational cost, not allowing to run the observer of the automobile in real time. In [13] the unscented Kalman filters [14, 15] were combined with multibody models, allowing the implementation of multibody-based state observers without having to derive complicated Jacobian matrices for the transition and measurement matrices, since the unscented Kalman filter uses a set of deterministically chosen samples (so called sigma points in this method) to propagate the covariance matrices through nonlinear functions. However, the computational cost continued to be a problem, especially for systems with several degrees of freedom, even considering that the samples employed in the unscented transform are independent, and therefore the computations can be parallelized if several cores are available.

A model reduction technique was applied in [16] to estimate both states and inputs in multibody models, showing the application of the method to a planar half-car model. In [17] a new variation of an unscented Kalman filter and a discrete extended Kalman filter were applied to a four-bar mechanism using only measurements from inertial sensors. The results were validated experimentally, and the observers were run online in real time.

A different approach is used in [18], where a proportional-integral controller is used to calculate the forces applied to a multibody model to make it follow the signal provided by the sensors. The advantage of this approach is that the the equations of the multibody model do not need to be explicitly known, and therefore this method can be applied to models build using commercial software. State observers based on linear multibody models with unilateral constraints are presented in [19, 20, 21].

In [22] a kinematic state observer is presented. The inputs of the system are obtained from accelerometers, instead of using the dynamics of model. This approach is less sensitive to modeling errors than the previous ones, but it requires more sensors. This method was extended in [23] by adding a force estimation algorithm after the kinematic estimation is performed. There are other kinematic approaches, such as [24, 25] in which inertial measurements are combined with the constraints of the system to provide an improved position information.

The use of multibody dynamics has provided a systematic approach and an accurate base to develop state estimators since they include the actual geometry and the kinematic constraints of the systems under study. However, the difficulty of the combination of multibody dynamics with Kalman filters lies in their different mathematical structure, being the Kalman filters formulated for first order systems of ordinary differential equations (ODEs), while multibody dynamics is represented by a second order system of differential-algebraic equa-

tions (DAEs). In [26], several methods were presented and compared. One of the most accurate methods, and the most efficient was an indirect observer, referred to as errorEKF. This formulation allowed to run a state observer based on the multibody model of a complete car in real time [27], although the method uses simplified Jacobians, and hence the results can be improved. The description of the validation of the vehicle model used in [27] can be found in [28].

The results presented in [26] show that the errorEKF performs well in most of the tests studied, but the simplifications applied to calculate the Jacobian matrix of its plant lead to great estimation errors in some of the tests. In this work, the exact Jacobian matrix is developed, and the resulting method is accurate even in the cases studien in [26] in which the errorEKF failed to provide reliable results. Moreover, using the same terms, another method adding input force estimation is also presented. This characteristic was not present in any of the methods studied in [26]. The benefits of this method are twofold: on the one hand, it provides input force estimation, which might be the main aim of some applications. On the other hand, the force estimation is used to improve the predictions made by the multibody model, thus achieving more accuracy than the methods without force estimation. In addition, the method with force estimation provides a more natural way to incorporate corrections from acceleration measurements.

All the three methods (the original errorEKF with simplified Jacobian, its counterpart with exact Jacobian, and the augmented version with force estimation) are evaluated with position, velocity, and acceleration sensors. Two different sampling rates are considered for every set of sensors tested. An unscented Kalman filter (UKF) using the trapezoidal rule integration is also included in this study as the reference, since it was the most accurate method of those presented in [26].

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 has also been 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, closed-chain mechanisms were selected in this work because they allow to show all the benefits from incorporating a multibody model into a Kalman filter, such as using a reduced number of sensors (only one measurement per degree of freedom is required) while still obtaining information about the whole system. Although the methods presented here are applied to planar mechanisms, they are general and can be employed with 3D mechanisms.

From the results of the tests, we can say that the errorEKF with exact Jacobian matrix of the plant, referred to as errorEKF_EJ, provides almost the same results as the UKF with trapezoidal rule integration, but with a much lower computational cost. The method with force estimation, provides even better accuracy than the UKF. In addition, it provides the input force estimation, and it is the only method of those considered in this work which provides reliable results when considering accelerometers as the only sensors available. The computational cost is again much lower than that of the UKF.

The different algorithms considered in this work have been implemented in MATLAB® as a reusable toolbox released as Open Source [2].

## 2. Methodology

As said before, in this work, three indirect extended Kalman filters are evaluated when applied to mechanisms described by multibody models. To achieve a fair comparison, three multibody models are used for each test. The first multibody model represents the actual mechanism, thus providing the ground truth, and it will be referred to as the *real mechanism*. Sensor models are built from the kinematics of this model, but white noise is added to represent the noise properties of real sensors.

The second model, which is referred to as *model*, represents a model of the *real mechanism*. Therefore, the model is similar to the real mechanism, but some differences are introduced, simulating modeling errors. When building a multibody model, usually the geometry and the mass can be accurately determined, but the force models have much more uncertainty. For this reason, the errors introduced in the *model* with respect to the *real mechanism* are in the force models. It is worth pointing out that errors in the mass or the mass distribution will lead to a similar accuracy level regarding kinematic magnitudes, because both force and mass errors produce acceleration errors. However, a bias in force estimation is expected when the mass properties are not well characterized, but this case have not been considered in this work.

The third model is the one which will be run in combination with the indirect Kalman filter, and is referred to as *observer*. This multibody model is the same as the second one (i.e., with errors), but it is corrected with the information provided by the sensors installed on the *real mechanism*.

The three methods presented in this work are evaluated by applying this methodology to the two mechanisms shown in fig. 1 and using different sets of sensors at position, velocity, and acceleration level. The main properties of the mechanisms are provided in tables 1 and 2.

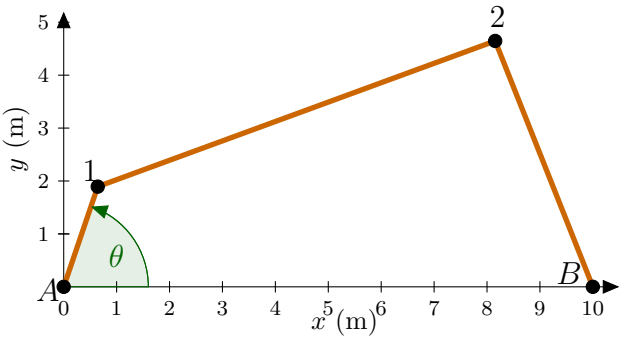
| | Crank | Coupler | Rocker | Ground element |
|---|---|---|---|---|
| Mass (kg) | 2 | 8 | 5 | — |
| Length (m) | 2 | 8 | 5 | 10 |

Table 1: Properties of the four-bar linkage.

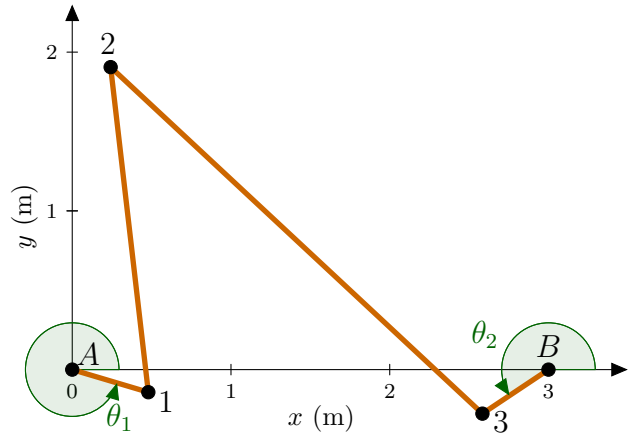
| | Left crank | Left coupler | Right coupler | Right crank | Ground element |
|---|---|---|---|---|---|
| Mass (kg) | 3 | 1 | 2 | 3 | — |
| Length (m) | 0.5 | 2.062 | 3.202 | 0.5 | 3 |

Table 2: Properties of the five-bar linkage.

[2] See `https://github.com/MBDS/mbde-matlab`




(a)




(b)

Figure 1: Mechanisms employed in this work. (a)Four-bar linkage. (b)Five-bar linkage.

## 3. Multibody dynamics: matrix R formulation

The state observers presented in the next sections can be applied to any multibody formulation and integrator. However, some terms of the matrix R formulation are used to propagate the covariance matrix of the estimation error and to map the sensor models over the independent coordinates. Therefore, the matrix R multibody formulation [1] is introduced in this section.

The dynamics of a multibody system can be described by the constrained Lagrangian equations:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} + {\mathbf{\Phi}_\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} &= \mathbf{Q} \\ \mathbf{\Phi} &= \mathbf{0} \end{cases} \tag{1}$$

where $\mathbf{M}$ is the mass matrix, $\ddot{\mathbf{q}}$ is the vector of dependent accelerations, $\mathbf{\Phi}$ is the constraints vector, $\mathbf{\Phi_q}$ 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.

Since this system of differential-algebraic equations (DAEs) is difficult to solve in this form, it is usually converted to an equivalent system of equations which is easier to solve. One option is to transform this system of DAEs into a system of ODEs using the projection method known as matrix R [1]. The formulation is described hereafter, but the interested reader is referred to [1] for further details.

First, the vector of dependent velocities $\dot{\mathbf{q}}$ is expressed as a function of the independent velocities $\dot{\mathbf{z}}$ as follows:

$$\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{z}} - \mathbf{S}\boldsymbol{\Phi}_t \tag{2}$$

where the term $\mathbf{R}\dot{\mathbf{z}}$ represents the general solutions of the homogeneous velocity equation 2, and the term $-\mathbf{S}\boldsymbol{\Phi}_t$ represents a particular solution of the complete equation.

The accelerations $\ddot{\mathbf{q}}$ can be written as:

$$\ddot{\mathbf{q}} = \mathbf{R}\ddot{\mathbf{z}} + \mathbf{S}\mathbf{c} \tag{3}$$

were $\mathbf{c} = -\dot{\boldsymbol{\Phi}}_t - \boldsymbol{\Phi}_{\mathbf{q}}\dot{\mathbf{q}}$ . Similarly, the virtual dependent velocities can be expressed as a function of the virtual independent velocities:

$$\dot{\mathbf{q}}^* = \mathbf{R}\dot{\mathbf{z}}^* \tag{4}$$

Combining eqs. (1), (3) and (4), and taking into account that the velocities $\dot{\mathbf{z}}$ are independent, the equations of motion of the multibody model can be expressed as a system of ODEs:

$$\left(\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\right)\ddot{\mathbf{z}} = \mathbf{R}^{\mathrm{T}}\left(\mathbf{Q} - \mathbf{M}\mathbf{S}\mathbf{c}\right) \tag{5}$$

which can be rewritten as:

$$\bar{\mathbf{M}}\ddot{\mathbf{z}} = \bar{\mathbf{Q}} \tag{6}$$

being $\bar{\mathbf{M}} = \mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}$, and $\bar{\mathbf{Q}} = \mathbf{R}^{\mathrm{T}}\left(\mathbf{Q} - \mathbf{M}\mathbf{S}\mathbf{c}\right)$.

## 4. Design of the Observers

The state observers presented in this work are based on the well known extended Kalman filter [8, 9]. However, given the mathematical structure of the multibody equations, the indirect filtering approach is used. In the indirect filters, commonly used in multisensor integration for navigation [29, 30], the variables of interest are not directly estimated. Instead, a dynamical model (in this case, a multibody model) is run without modifications, and then the indirect filter estimates the drift of the model by comparing its output with the sensors available. After every measurement, the estimated error is applied to the multibody model to correct it. Although any multibody formulation and integrator can be used with this approach, some terms of the matrix R

| Method | Classification | State vector | Integrator |
|---|---|---|---|
| errorEKF[a] | Discrete-time EKF with simplified Jacobian matrix, indirect formulation | Error in indep. coord. and velocities | Any for the MBS, Forward Euler for the KF |
| errorEKF_EJ[b] | Discrete-time EKF with exact Jacobian matrix, indirect formulation | Error in indep. coord. and velocities | Any for the MBS, Forward Euler for the KF |
| errorEKF_FE[b] | Discrete-time EKF with complete Jacobian matrix and input force estimation, indirect formulation | Error in indep. coord., velocities and accelerations | Any for the MBS, Forward Euler for the KF |

[a] First presented in [26].
[b] Original contributions of this work.

Table 3: Summary of the methods.

formulation are used to propagate the covariance matrix of the estimation error and for mapping the sensor models into the independent coordinates space.

This approach, first applied to multibody models in [26], has shown advantages regarding both easiness of implementation and computational efficiency. In this work, the multibody formulation used to run the multibody simulations was the augmented Lagrangian of index 3 with velocity and acceleration projections, combined with the trapezoidal rule as the integrator [31, 32, 33].

A summary of the methods presented in this work is shown in Table 3. For ease of reference, the list of the principal symbols used is given in Table 4.

### *4.1. Error-State Extended Kalman Filter (errorEKF)*

This method was already described in [26], but it is presented hereafter for the convenience of the reader.

With the indirect estimation, also known as error-state estimation, the dynamical system (in this case, the multibody model) is run without modifications, and hence any multibody formulation or integrator can be employed, including methods with implicit integrators, dependent coordinates, projections, etc (see, e.g. [34]). Even previously implemented multibody simulations can be used with minimal changes to the code. After one time step of the multibody simulation has been run, the indirect observer estimates the error committed by the multibody model in tracking the real system. Then, the error is corrected, and a new time step is run.

A simplified diagram 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 $\mathbf{q}_{MBS}$ and velocities $\dot{\mathbf{q}}_{MBS}$. Next, the estimation process takes place on a state vector $\mathbf{x}^{\mathrm{T}} = \left[\Delta\mathbf{z}^{\mathrm{T}}, \Delta\dot{\mathbf{z}}^{\mathrm{T}}\right]$ which consists in the *error* in position and velocity of the degrees of freedom of the mechanism, such that:

$$\mathbf{z} = \mathbf{z}_{MBS} + \Delta\mathbf{z} \tag{7}$$

$$\dot{\mathbf{z}} = \dot{\mathbf{z}}_{MBS} + \Delta\dot{\mathbf{z}} \tag{8}$$

| | |
|---|---|
| $\mathbf{C}$ | Damping matrix in dependent coordinates, $\mathbf{C} = -\frac{\partial \mathbf{Q}}{\partial \dot{\mathbf{q}}}$ |
| $\bar{\mathbf{C}}$ | Damping matrix in independent coordinates $\bar{\mathbf{C}} = -\frac{\partial \bar{\mathbf{Q}}}{\partial \dot{\mathbf{z}}}$ |
| $\mathbf{f}(), \mathbf{f_x}$ | Discrete transition model, and its Jacobian matrices wrt. the states $\mathbf{x}$ |
| $\mathbf{G}$ | Kalman gain matrix |
| $g$ | Number of degrees of freedom |
| $\mathbf{h}(), \mathbf{h_x}, \mathbf{h_q}$ | Measurement model and its Jacobian matrices wrt. the states $\mathbf{x}$, and the dependent coordinates $\mathbf{q}$ |
| $\mathbf{K}$ | Stiffness matrix in dependent coordinates $\mathbf{K} = -\frac{\partial \mathbf{Q}}{\partial \mathbf{q}}$ |
| $\bar{\mathbf{K}}$ | Stiffness matrix in independent coordinates $\bar{\mathbf{K}} = -\frac{\partial \bar{\mathbf{Q}}}{\partial \mathbf{z}}$ |
| $\mathbf{M}$ | Mass matrix |
| $\bar{\mathbf{M}}$ | Mass matrix projected over the set of independent coordinates |
| $m$ | Number of constraints |
| $n$ | Number of dependent 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 independent coordinates |
| $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ | Vector of dependent positions, velocities, and accelerations |
| $\hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}, \hat{\ddot{\mathbf{q}}}$ | Estimation of the vectors of dependent positions, velocities and accelerations |
| $\mathbf{R}$ | Velocity projection matrix, such that $\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{z}} - \mathbf{S}\mathbf{\Phi}_t$ |
| $\mathbf{S}$ | Transformation matrix for rheonomous constraints |
| $\mathbf{x}$ | States vector |
| $\hat{\mathbf{x}}^-, \hat{\mathbf{x}}^+$ | 'A priori' state estimation, 'a posteriori' state estimation |
| $\mathbf{y}$ | Innovations vector |
| $\mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}}$ | Vector of independent coordinates, velocities, and accelerations |
| $\hat{\mathbf{z}}, \hat{\dot{\mathbf{z}}}, \hat{\ddot{\mathbf{z}}}$ | Estimation of the vectors of independent coordinates, velocities and accelerations |
| $\Delta t$ | Time step |
| $\boldsymbol{\lambda}$ | Vector of Lagrange multipliers |
| $\mathbf{\Sigma}^P$ | Covariance matrix of the plant noise in discrete filters |
| $\mathbf{\Sigma}_c^P$ | Covariance matrix of the plant noise in continuous filters |
| $\mathbf{\Sigma}^S$ | Covariance matrix of measurements noise |
| $\mathbf{\Sigma}$ | Covariance matrix of the innovation |
| $\mathbf{\Phi}, \mathbf{\Phi_q}$ | Vector of constraints, and its Jacobian wrt. the coordinates |

Table 4: List of the main symbols used

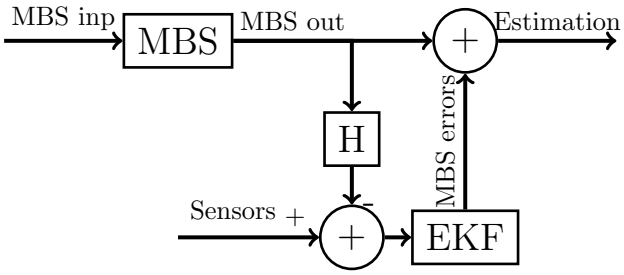



Figure 2: Simplified flow diagram of a time step of the errorEKF.

where $\mathbf{z}_{MBS}$ and $\dot{\mathbf{z}}_{MBS}$ are values of positions and velocities of the degrees of freedom predicted by the multibody model before the corrections are applied, and $\mathbf{z}$ and $\dot{\mathbf{z}}$ are their actual values.

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

$$\hat{\mathbf{x}}_k^- = \mathbf{0} \tag{9}$$

$$\mathbf{P}_k^- = \mathbf{f}_{\mathbf{x}k-1}\mathbf{P}_{k-1}^+\mathbf{f}_{\mathbf{x}k-1}^{\mathrm{T}} + \mathbf{\Sigma}^P \tag{10}$$

These equations are the conventional equations for the propagation of the discrete Kalman filter [8, 9] , and the transition matrix $\mathbf{f}_{\mathbf{x}}$, which represents a forward Euler integration, is as follows:

$$\mathbf{f}_{\mathbf{x}} = \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{11}$$

In this method, the variation of the acceleration error with position and velocity errors has been neglected for the sake of simplicity, thus leading to a constant matrix.

Note that, since the errorEKF operates in the transformed state space of errors, its a priori estimate ($\hat{\mathbf{x}}_k^-$) is always null, hence the zero in Eq. (9). 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 a conventional extended Kalman filter.

$$\mathbf{y}_k = \mathbf{o}_k - \mathbf{h}(\mathbf{q}_k, \dot{\mathbf{q}}_k) \tag{12}$$

$$\mathbf{\Sigma}_k = \mathbf{h}_{\mathbf{x}k}\mathbf{P}_k^-\mathbf{h}_{\mathbf{x}k}^{\mathrm{T}} + \mathbf{\Sigma}_k^S \tag{13}$$

$$\mathbf{G}_k = \mathbf{P}_k^-\mathbf{h}_{\mathbf{x}k}^{\mathrm{T}}\mathbf{S}_k^{-1} \tag{14}$$

$$\hat{\mathbf{x}}_k^+ = \mathbf{0} + \mathbf{K}_k\mathbf{y}_k \tag{15}$$

$$\mathbf{P}_k^+ = (\mathbf{I}_{2g} - \mathbf{K}_k\mathbf{h}_{\mathbf{x}k})\mathbf{P}_k^- \tag{16}$$

The difference can be found in Eq. (12), in which 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 an equivalent conventional Kalman filter, 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, as shown in Eq. (15).

After the correction stage, the estimation of the position and velocity errors of the independent coordinates is obtained. However, to correct the state of the multibody system, the errors for all the coordinates must be obtained, so they must be projected over the constraints manifold.

Although the position problem could be solved, it is an iterative problem, and hence it is computationally expensive. However, since the corrections are expected to be small, a linearization of the position problem can be solved instead, as follows:

$$\mathbf{\Phi_q} \Delta \mathbf{q} = \mathbf{0} \tag{17}$$

Therefore, the increments $\Delta\mathbf{q}$ applied to the coordinates of the mechanism are calculated by solving Eq. (17). If the Jacobian of the constraints $\mathbf{\Phi_q}$ and the vector of position errors are split in their independent and dependent parts, Eq. (17) can be written as follows:

$$\Delta \mathbf{q} = \begin{bmatrix} \left(\Delta \mathbf{q}^i\right)^{\mathrm{T}} & \left(\Delta \mathbf{q}^d\right)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} \Delta \mathbf{z}^{\mathrm{T}} & \left(\Delta \mathbf{q}^d\right)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{18}$$

$$\mathbf{\Phi_q}^d \Delta \mathbf{q}^d = -\mathbf{\Phi_q}^i \Delta \mathbf{z} \tag{19}$$

The estimated error of the degrees of freedom $\Delta\hat{\mathbf{z}}$ is substituted into Eq. (19), and the resulting linear system is solved to obtain the estimated dependent errors $\Delta\hat{\mathbf{q}}^d$. 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:

$$\hat{\mathbf{q}} = \mathbf{q}_{MBS} + \Delta \hat{\mathbf{q}} \tag{20}$$

Since this method is an approximation, a perfect fulfillment of the constraints at position level is not expected. However, since the corrections are performed every time step, the errors are usually acceptable for most applications. The main advantage of this method is that it avoids solving the position problem iteratively. Nevertheless, if the highest possible accuracy is required, the exact position problem must be solved.

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

$$\mathbf{\Phi_q}^d \hat{\dot{\mathbf{q}}}^d = -\mathbf{\Phi}_t - \mathbf{\Phi_q}^i \left[\dot{\mathbf{z}}_{MBS} + \Delta \hat{\dot{\mathbf{z}}}\right] \tag{21}$$

Therefore, the vector of estimated velocities yields:

$$\hat{\dot{\mathbf{q}}} = \begin{bmatrix} \left(\dot{\mathbf{z}}_{MBS} + \Delta \hat{\dot{\mathbf{z}}}\right)^{\mathrm{T}} & \left(\hat{\dot{\mathbf{q}}}^d\right)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{22}$$

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

*4.2. ErrorEKF with exact Jacobian (errorEKF_EJ)*

In the previous method, the dependency of the accelerations with the positions and velocities was neglected when calculating the Jacobian of the plant matrix $\mathbf{f_x}$. In some cases, the accuracy achieved with these simplifications might fulfill the requirements of a certain application, and the resultant filter is simple to implement. However, depending on the type and number of sensors, these simplifications may lead to big errors, which can be fixed by including the complete Jacobian matrix of the transition model. In particular, when the relationship between the states of the filter and the sensors models is kinematic, the simplified version of the filter provides accurate results. In contrast, when only the variation of the forces of the model affects the measurement of the sensors, the complete version of the Jacobian matrix is required.

As in the previous method, the integrator assumed to integrate the error is the forward Euler. Then, the complete Jacobian matrix of the transition matrix can be written as follows:

$$\mathbf{f_x} = \left[\begin{array}{c|c} \mathbf{I} + \frac{1}{2}\frac{\partial \Delta \ddot{\mathbf{z}}}{\partial \mathbf{z}}\Delta t^2 & \mathbf{I}\Delta t + \frac{1}{2}\frac{\partial \Delta \ddot{\mathbf{z}}}{\partial \dot{\mathbf{z}}}\Delta t^2 \\ \hline \frac{\partial \Delta \ddot{\mathbf{z}}}{\partial \mathbf{z}}\Delta t & \mathbf{I} + \frac{\partial \Delta \ddot{\mathbf{z}}}{\partial \dot{\mathbf{z}}}\Delta t \end{array}\right] \tag{23}$$

where

$$\frac{\partial \Delta \ddot{\mathbf{z}}}{\partial \mathbf{z}} = -\left(\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\right)^{-1}\bar{\mathbf{K}} - \left(\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\right)^{-1}\frac{\partial\left(\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\right)}{\partial \mathbf{z}}\left(\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\right)^{-1}\bar{\mathbf{Q}} \tag{24}$$

$$\frac{\partial\left(\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\right)}{\partial \mathbf{z}} = \left(\mathbf{R}_{\mathbf{z}}^{\mathrm{T}}\mathbf{M}\mathbf{R} + \mathbf{R}^{\mathrm{T}}\mathbf{M}_{\mathbf{z}}\mathbf{R} + \mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}_{\mathbf{z}}\right) \tag{25}$$

$$\frac{\partial \Delta \ddot{\mathbf{z}}}{\partial \dot{\mathbf{z}}} = -\left(\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\right)^{-1}\bar{\mathbf{C}} \tag{26}$$

Most of the terms needed to implement these expressions have been developed in [35], but are included hereafter for the convenience of the reader.

$$\bar{\mathbf{K}} = -\frac{\partial \bar{\mathbf{Q}}}{\partial \mathbf{z}} = -\left(\bar{\mathbf{Q}}_{\mathbf{q}} - \bar{\mathbf{Q}}_{\dot{\mathbf{q}}}\mathbf{S}\dot{\mathbf{\Phi}}_{\mathbf{q}}\right)\mathbf{R} \tag{27}$$

$$\bar{\mathbf{C}} = -\frac{\partial \bar{\mathbf{Q}}}{\partial \dot{\mathbf{z}}} = -\bar{\mathbf{Q}}_{\dot{\mathbf{q}}}\mathbf{R} \tag{28}$$

$$\bar{\mathbf{Q}}_{\mathbf{q}} = \frac{\partial \bar{\mathbf{Q}}}{\partial \mathbf{q}} = \mathbf{R}_{\mathbf{q}}^{\mathrm{T}}\left(\mathbf{Q} - \mathbf{M}\mathbf{S}\mathbf{c}\right) - \mathbf{R}^{\mathrm{T}}\left(\mathbf{K} + \mathbf{M}_{\mathbf{q}}\mathbf{S}\mathbf{c} + \mathbf{M}\frac{\partial \mathbf{S}\mathbf{c}}{\partial \mathbf{q}}\right) \tag{29}$$

$$\bar{\mathbf{Q}}_{\dot{\mathbf{q}}} = \frac{\partial \bar{\mathbf{Q}}}{\partial \dot{\mathbf{q}}} = -\mathbf{R}^{\mathrm{T}}\left(\mathbf{C} + \mathbf{M}\mathbf{S}\mathbf{c}_{\dot{\mathbf{q}}}\right) \tag{30}$$

$$\mathbf{R}_{\mathbf{q}}^{\mathrm{T}} = -\mathbf{R}^{\mathrm{T}}\mathbf{\Phi}_{\mathbf{qq}}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}} \tag{31}$$

$$\frac{\partial \mathbf{Sc}}{\partial \mathbf{q}} = \mathbf{S}\left(-\mathbf{\Phi_{qq}}\mathbf{Sc} + \mathbf{c_q}\right) \tag{32}$$

$$\mathbf{c_q} = -\dot{\mathbf{\Phi}}_{\mathbf{qq}}\dot{\mathbf{q}} - \dot{\mathbf{\Phi}}_{t\mathbf{q}} \tag{33}$$

$$\mathbf{c}_{\dot{\mathbf{q}}} = -\dot{\mathbf{\Phi}}_{\mathbf{q}\dot{\mathbf{q}}}\dot{\mathbf{q}} - \dot{\mathbf{\Phi}}_{\mathbf{q}} - \dot{\mathbf{\Phi}}_{t\dot{\mathbf{q}}} = -\mathbf{\Phi_{qq}}\dot{\mathbf{q}} - \dot{\mathbf{\Phi}}_{\mathbf{q}} - \mathbf{\Phi}_{t\mathbf{q}} = -2\dot{\mathbf{\Phi}}_{\mathbf{q}} \tag{34}$$

### *4.3. ErrorEKF with force estimation (errorEKF_FE)*

This indirect state observer can be also extended to estimate acceleration and forces. When there are no force sensors, the force estimation is limited to one force for each degree of freedom of the mechanism. If the unknown forces to be estimated come from sources of different nature, this information could be employed to estimate their individual effect, thus estimating more forces than degrees of freedom, but this technique would be specific for every problem and it is beyond the scope of this work.

To estimate the input forces, the first step is to add the input acceleration error of the degrees of freedom to the states vector, as follows:

$$\mathbf{x} = \begin{bmatrix} \Delta\mathbf{z}^{\mathrm{T}} & \Delta\dot{\mathbf{z}}^{\mathrm{T}} & \Delta\ddot{\mathbf{z}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{35}$$

The model used for the acceleration error is a random walk, i.e., it is assumed that it is constant, and the only source of variation is the noise. The other states are integrated using the forward Euler method. Therefore, the Jacobian matrix of the transition model is as follows:

$$\mathbf{f_x} = \left[\begin{array}{c:c:c} \mathbf{I} + \frac{1}{2}\frac{\partial \Delta\ddot{\mathbf{z}}}{\partial \mathbf{z}}\Delta t^2 & \mathbf{I}\Delta t + \frac{1}{2}\frac{\partial \Delta\ddot{\mathbf{z}}}{\partial \dot{\mathbf{z}}}\Delta t^2 & \frac{1}{2}\mathbf{I}\Delta t^2 \\ \hdashline \frac{\partial \Delta\ddot{\mathbf{z}}}{\partial \mathbf{z}}\Delta t & \mathbf{I} + \frac{\partial \Delta\ddot{\mathbf{z}}}{\partial \dot{\mathbf{z}}}\Delta t & \mathbf{I}\Delta t \\ \hdashline \mathbf{0} & \mathbf{0} & \mathbf{I} \end{array}\right] \tag{36}$$

The blocks in position (3,1) and (3,2) in eq. (36) are set to $\mathbf{0}$ because the evolution of the acceleration error is unknown. The terms $\frac{\partial \Delta\ddot{\mathbf{z}}}{\partial \mathbf{z}}$ and $\frac{\partial \Delta\ddot{\mathbf{z}}}{\partial \dot{\mathbf{z}}}$ are included in the blocks (1,1), (1,2), (2,1), and (2,2), as in the previous method, to account for the effect of the acceleration error through the force models which are modeled in the multibody model.

The rest of the algorithm is analogous to the methods described previously: during the prediction stage, the evolution of the error is null, and only the covariance matrix is updated. Then, the measurements from sensors are applied, thus obtaining an estimate of the error of the multibody model in tracking the real mechanism at position, velocity and acceleration levels.

Finally, the multibody model is corrected before the following time step. Position and velocity corrections are applied as in the methods previously described. However, the correction at acceleration level is slightly different. First,

the acceleration correction is applied in an analogous way to the position and velocity corrections: the accelerations of the degrees of freedom of the mechanism predicted by the multibody model are corrected with the estimations provided by the filter, and then, the accelerations of the remaining coordinates are obtained by solving the acceleration problem, as follows [1]:

$$\mathbf{\Phi_q}\hat{\ddot{\mathbf{q}}}^d = -\dot{\mathbf{\Phi}}_t - \dot{\mathbf{\Phi}}_\mathbf{q}\dot{\mathbf{q}} - \mathbf{\Phi_q}\left(\ddot{\mathbf{z}}_{MBS} + \Delta\hat{\ddot{\mathbf{z}}}\right) \tag{37}$$

The final estimated acceleration vector results as follows:

$$\hat{\ddot{\mathbf{q}}} = \left[\left(\ddot{\mathbf{z}}_{MBS} + \Delta\hat{\ddot{\mathbf{z}}}\right)^{\mathrm{T}} \quad \left(\hat{\ddot{\mathbf{q}}}^d\right)^{\mathrm{T}}\right]^{\mathrm{T}} \tag{38}$$

Once the accelerations have been corrected, the generalized forces $\Delta\mathbf{Q}$ which would have to be applied to the mechanism to avoid the estimated acceleration error are calculated. As there are more coordinates than degrees of freedom, there are infinite possibilities to calculate a vector $\Delta\mathbf{Q}$ which produces the desired effect. In this work, it is assumed that all the unknown forces are applied to the degrees of freedom, such that:

$$\Delta\mathbf{Q}_i^d = \mathbf{0} \tag{39}$$

$$\Delta\mathbf{Q}_i^i = \Delta\mathbf{Q}_{i-1}^i + \bar{\mathbf{M}}\Delta\hat{\ddot{\mathbf{z}}} \tag{40}$$

$$\Delta\mathbf{Q} = \left[\left(\Delta\mathbf{Q}^i\right)^{\mathrm{T}} \quad \left(\Delta\mathbf{Q}^d\right)^{\mathrm{T}}\right]^{\mathrm{T}} \tag{41}$$

where the subindex $i$ refers to the time step. The force $\Delta\mathbf{Q}$ is added to the vector of the generalized forces of the model, and it is applied to the integration of the following time step. Therefore, the expected acceleration error of the multibody model for the next time step is null. Note that these forces are calculated taking into account only the acceleration error of the degrees of freedom. The velocity-dependent accelerations must not be included because they are implicitly corrected when the velocity correction is applied.

## 5. Sensor models

In this paper we considered position sensors (encoders), angular velocity sensors (gyroscopes), and acceleration sensors (accelerometers). In this section we provide generic expressions for mapping the relationship between the whole set of positions $\mathbf{q}$, velocities $\dot{\mathbf{q}}$, and acceleration $\ddot{\mathbf{q}}$, and models of the sensors.

In general, the model of a sensor depends on positions, velocities and accelerations, such that $\mathbf{h} = \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$. Since the state observer estimates the error of the independent coordinates, the model of the sensors must be mapped over the independent coordinates taking into account only kinematic relations, i.e., the effect of the forces of the model is not accounted for when doing the partial

derivatives needed in this section. The kinematic relations for gyroscopes and accelerometers are provided in appendices A and B.

The partial derivative of the sensor model with respect to the positions, velocities , and accelerations can be written as follows:

$$\mathbf{h_z} = \frac{\partial \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})}{\partial \mathbf{z}} = \frac{\partial \mathbf{h}}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial \mathbf{z}} + \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}} \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{z}} + \frac{\partial \mathbf{h}}{\partial \ddot{\mathbf{q}}} \frac{\partial \ddot{\mathbf{q}}}{\partial \mathbf{z}} \tag{42}$$

$$\mathbf{h}_{\dot{\mathbf{z}}} = \frac{\partial \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})}{\partial \dot{\mathbf{z}}} = \frac{\partial \mathbf{h}}{\partial \mathbf{q}} \cancelto{0}{\frac{\partial \mathbf{q}}{\partial \dot{\mathbf{z}}}} + \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}} \frac{\partial \dot{\mathbf{q}}}{\partial \dot{\mathbf{z}}} + \frac{\partial \mathbf{h}}{\partial \ddot{\mathbf{q}}} \frac{\partial \ddot{\mathbf{q}}}{\partial \dot{\mathbf{z}}} \tag{43}$$

$$\mathbf{h}_{\ddot{\mathbf{z}}} = \frac{\partial \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})}{\partial \ddot{\mathbf{z}}} = \frac{\partial \mathbf{h}}{\partial \mathbf{q}} \cancelto{0}{\frac{\partial \mathbf{q}}{\partial \ddot{\mathbf{z}}}} + \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}} \cancelto{0}{\frac{\partial \dot{\mathbf{q}}}{\partial \ddot{\mathbf{z}}}} + \frac{\partial \mathbf{h}}{\partial \ddot{\mathbf{q}}} \frac{\partial \ddot{\mathbf{q}}}{\partial \dot{\mathbf{z}}} \tag{44}$$

where the next expressions can be used:

$$\frac{\partial \mathbf{q}}{\partial \mathbf{z}} = \frac{\partial \dot{\mathbf{q}}}{\partial \dot{\mathbf{z}}} = \frac{\partial \ddot{\mathbf{q}}}{\partial \ddot{\mathbf{z}}} = \mathbf{R} \tag{45}$$

$$\frac{\partial \ddot{\mathbf{q}}}{\partial \mathbf{z}} = \mathbf{R}_{\mathbf{z}} \ddot{\mathbf{z}} + \frac{\partial \mathbf{Sc}}{\partial \mathbf{q}} \mathbf{R} + \mathbf{Sc}_{\dot{\mathbf{q}}} \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{z}} \tag{46}$$

$$\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{z}} = -\mathbf{S}\dot{\mathbf{\Phi}}_{\mathbf{q}} \mathbf{R} \tag{47}$$

$$\frac{\partial \ddot{\mathbf{q}}}{\partial \dot{\mathbf{z}}} = \mathbf{Sc}_{\dot{\mathbf{q}}} \mathbf{R} = -2\mathbf{S}\dot{\mathbf{\Phi}}_{\mathbf{q}} \mathbf{R} = 2\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{z}} \tag{48}$$

## 6. 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 the *real mechanism*, and then white Gaussian noise is generated and added to the sensor signal. Hence noise properties are perfectly known, and are used to build the covariance matrices of measurement noise. The same sequence of pseudorandom noise was used in all the tests to ensure a fair comparison among the different methods.

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 know where the errors are, or it is not possible to correct them for any reason.

When the ground truth is not know, the only way to check that the filter is working well 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. Unfortunately, this method works well only for the state observers without force estimation. When the input force estimation is added, the estimated force is applied to the plant, and hence the plant is modified. This makes the plant noise of this method with force estimation more difficult to adjust, and it has to be tuned by trial and error.

### *6.1. 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 is usually more complex in practice. These effects appear as errors in the acceleration. 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 state observers used in this work.

In the methods with position and velocity estimation, the covariance matrix of the plant noise $\mathbf{\Sigma}^P$ must be calculated from its continuous counterpart $\mathbf{\Sigma}_c^P$ by integration:

$$\mathbf{\Sigma}_c^P = \left[\begin{array}{c:c} \mathbf{0} & \mathbf{0} \\ \hdashline \mathbf{0} & diag\left(\boldsymbol{\sigma}_{\ddot{\mathbf{z}}}^2\right) \end{array}\right] \tag{49}$$

$$\mathbf{\Sigma}^P = \mathbf{f_x}(t_k, t_{k-1})\mathfrak{J}{\mathbf{f_x}}^{\mathrm{T}}(t_k, t_{k-1}) \tag{50}$$

$$\mathfrak{J} = \int_{t_{k-1}}^{t_k} {\mathbf{f_x}}^{-1}(\tau, t_{k-1})\mathbf{\Sigma}_c^P{\mathbf{f_x}}^{-\mathrm{T}}(\tau, t_{k-1})d\tau \tag{51}$$

Where $diag\left(\boldsymbol{\sigma}_{\ddot{\mathbf{z}}}^2\right)$ is a diagonal matrix containing the variance of all the components of the continuous plant noise at acceleration level.

To avoid the recalculation of the noise matrix every time step, only the constant terms of the Jacobian matrix of the plant have been employed in this work. This integration can be done using Van Loan's method [36], obtaining a covariance matrix of process noise with the following structure for the state observers without force estimation:

$$\mathbf{\Sigma}^P = \left[\begin{array}{c:c} \frac{\Delta t^3}{3} diag\left(\boldsymbol{\sigma}_{\ddot{\mathbf{z}}}^2\right) & \frac{\Delta t^2}{2} diag\left(\boldsymbol{\sigma}_{\ddot{\mathbf{z}}}^2\right) \\ \hdashline \frac{\Delta t^2}{2} diag\left(\boldsymbol{\sigma}_{\ddot{\mathbf{z}}}^2\right) & \Delta t\, diag\left(\boldsymbol{\sigma}_{\ddot{\mathbf{z}}}^2\right) \end{array}\right] \tag{52}$$

When dealing with the method with input estimation, the acceleration noise can be directly introduced in the covariance matrix of the discrete plant noise,

as follows:

$$\mathbf{\Sigma}^P = \left[\begin{array}{c|c|c} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & diag\left(\boldsymbol{\sigma}^2_{\ddot{\mathbf{z}},D}\right) \end{array}\right] \tag{53}$$

where $diag\left(\boldsymbol{\sigma}^2_{\ddot{\mathbf{z}},D}\right)$ is a diagonal matrix containing the variance of all the components of the discrete plant noise at acceleration level.

Both $diag\left(\boldsymbol{\sigma}^2_{\ddot{\mathbf{z}}}\right)$ and $diag\left(\boldsymbol{\sigma}^2_{\ddot{\mathbf{z}},D}\right)$ are adjusted by trial and error, but the difference between them is that $diag\left(\boldsymbol{\sigma}^2_{\ddot{\mathbf{z}}}\right)$ is independent of the simulation time step because it is a continuous variance , while $diag\left(\boldsymbol{\sigma}^2_{\ddot{\mathbf{z}},D}\right)$ must be increased or decreased proportionally to the multibody time step if it is modified, since it is a discrete variance. For more information about the relation between the continuous-time and the discrete-time covariances, see [9].

## 7. Tests and results

For the tests, the mechanisms are in the vertical plane, and the gravity is the only actuating force. The differences introduced to simulate modeling errors are a wrong value of the acceleration of the gravity, and an initial position error. The wrong value of the gravity affects to the plant during the whole simulation, just as any other unmodeled force would do.

The initial position error provides a non-formal demonstration of the system observability : if the position error is corrected, it means that the configuration being considered is observable under the test conditions. The reader interested in a more formal study of the observability of these methods is referred to [26].

Two levels of modeling error were used in these tests. The low modeling error was 0.5 $m/s^2$ in gravity and $\pi/32$ rad in initial position, while the high modeling error considered was 1 $m/s^2$ in gravity and $\pi/16$ rad in initial position.

Although there are many possible sensor configurations, in this work we considered only four possibilities for each mechanism: position sensors measuring the crank angle of the mechanisms (encoders), angular rate sensors (gyroscopes) attached to the couplers of the mechanisms, angular rate sensors attached to the cranks, and pairs of accelerometers attached to each of the cranks. One of the sensitive axes of ear pair of accelerometers is parallel to the crank to which it is attached, and the other is perpendicular to it. The accelerometers are attached at the end of the cranks, at the points where they meet the couplers. All these sensor configurations are shown in fig. 3 for the four-bar mechanism. The configurations tested in the five-bar linkage are analogous to those of the four-bar mechanism, but duplicating the number of sensors because the five-bar linkage has two degrees of freedom.

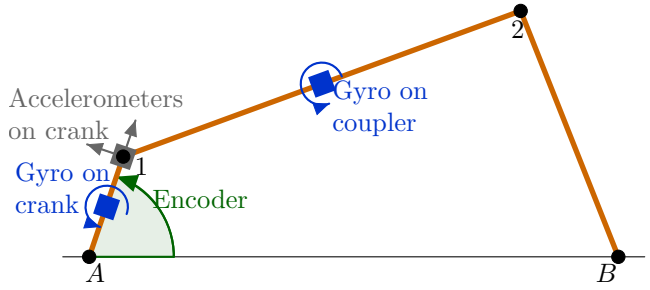



Figure 3: Schematic of the different sets of sensors used on the four-bar mechanism

| | errorEKF<br>errorEKF_EJ | errorEKF_FE |
|---|---|---|
| Low modeling error | 6.109e-2 | 4.072e-4 |
| High modeling error | 9.163e-2 | 2.291e-3 |

Table 5: Plant noise used for the four-bar mechanism. Value of $\boldsymbol{\sigma}^2_{\ddot{\mathbf{z}}}$ for errorEKF and errorEKF_EJ, and value of $\boldsymbol{\sigma}^2_{\ddot{\mathbf{z}},D}$ for errorEKF_FE. Units are rad$^2$/s$^4$.

The models used for the gyroscopes and accelerometers are provided in appendices A and B.

To simulate actual measurements, white Gaussian noise was added to the measurements. The standard deviation of the noise added to position measurements was 1.745e-2 rad, which is approximately equivalent to a 360 pulses per revolution encoder sampled at 1x, or to a 90 ppr encoder sampled at 4x. The noise properties for gyroscopes and accelerometers were obtained from actual measurements taken from an Invensense MPU-9250, a low-cost IMU, obtaining a standard deviation of 9.839e-4 rad/s for the noise of the gyroscopes, and 5.638e-2 m/s$^2$ for noise of the accelerometers.

All the multibody simulations were run with a time step of 5e-3 s, providing information about the mechanism at 200 Hz. The sampling rates considered for the sensors were 200 Hz (i.e., one measurement every time step of the multibody simulation) and 25 Hz (i.e., one measurement available every 8 simulation time steps).

The values of the plant noise used for the different tests are shown in tables 5 and 6.

### *7.1. Accuracy evaluation*

The results obtained when considered the gravity and initial position error of the crank angle with the four-bar mechanism are shown in figs. 4 to 6, where fig. 4 shows the rms error of the different tests at position level (i.e.: crank angle error), fig. 5 displays the rms error at velocity level (i.e.: crank angular rate error), and fig. 6 shows the rms error at acceleration level (i.e: crank angular

| | errorEKF<br>errorEKF_EJ | errorEKF_FE |
|---|---|---|
| Low modeling error | 3.054e-1 | 6.109e-2 |
| High modeling error | 5.498e-1 | 1.100e-1 |

Table 6: Plant noise used for the five-bar mechanism. Value of the components $\boldsymbol{\sigma}^2_{\ddot{\mathbf{z}}}$ for errorEKF and errorEKF_EJ, and value of the components $\boldsymbol{\sigma}^2_{\ddot{\mathbf{z}},D}$ for errorEKF_FE. Units are $\mathrm{rad}^2/\mathrm{s}^4$.

acceleration error). In addition to the three methods described in this work, the results from the unscented Kalman filter with trapezoidal rule integration (UKF TR) were also included as a reference, since it was the method which provided the best results in [26].

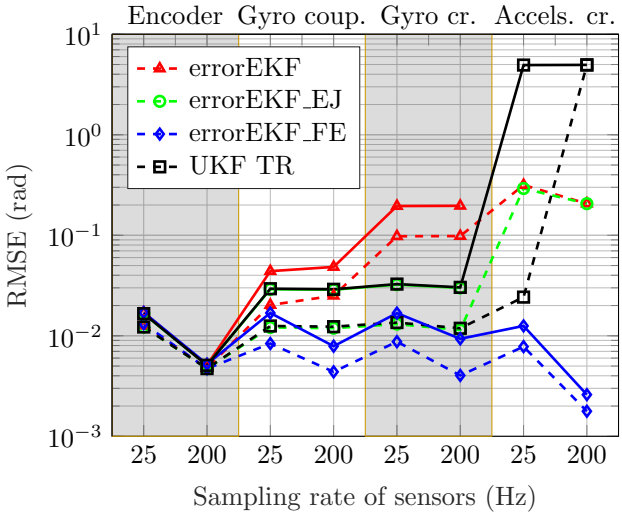



Figure 4: Position error provided by the observers for the four-bar mechanism with gravity and initial position error, evaluated as the rms of the crank angle error. Dashed lines represent the low modeling error (0.5 $\mathrm{m/s}^2$ gravity error and $\pi/32$ rad initial position error) and continuous lines the high modeling error (1 $\mathrm{m/s}^2$ gravity error and $\pi/16$ rad initial position error).

The errorEKF with exact Jacobian matrices (errorEKF_EJ) provides results almost identical to those of the UKF TR in most of the tests. The errorEKF with simplified Jacobian is in most of the tests the method which performs worst. In particular, when the sensor used is a gyroscope on the crank, the system is not observable [26], and hence the initial position error is not corrected.

To understand better the behavior of the different observers in this particular configuration, the angle of the crank for the whole maneuver is shown in fig. 7. The difference between the observer and the real mechanism is almost inappreciable in this plot. Therefore, the actual tracking error of the different methods is shown in fig. 8. It can be seen that the initial position error is not corrected when using the errorEKF, but it is properly estimated with the other

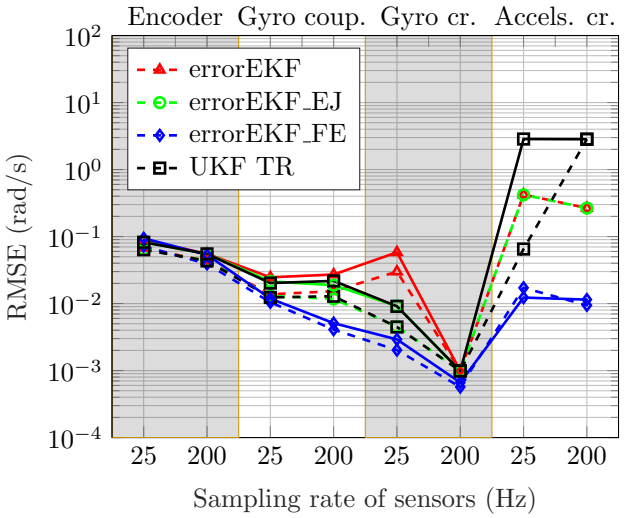



Figure 5: Velocity error provided by the observers for the four-bar mechanism with gravity and initial position error, evaluated as the rms of the crank angular rate error. Dashed lines represent the low modeling error (0.5 m/s$^2$ gravity error and $\pi/32$ rad initial position error) and continuous lines the high modeling error (1 m/s$^2$ gravity error and $\pi/16$ rad initial position error).

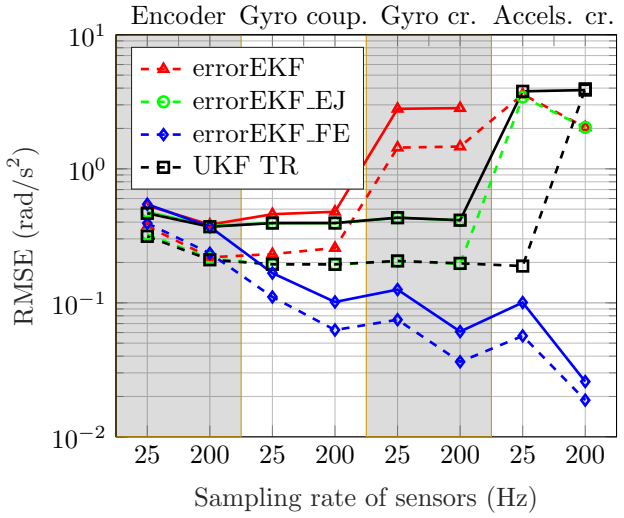



Figure 6: Acceleration error provided by the observers for the four-bar mechanism with gravity and initial position error, evaluated as the rms of the crank angular acceleration error. Dashed lines represent the low modeling error (0.5 m/s$^2$ gravity error and $\pi/32$ rad initial position error) and continuous lines the high modeling error (1 m/s$^2$ gravity error and $\pi/16$ rad initial position error).

methods. The convergence to the true value is faster with the errorEKF_FE than with the errorEKF_EJ. The confidence intervals are consistent with the actual estimation errors in all the methods: it is constant in the errorEKF, but if shrinks in the other methods when the error is corrected. This kind of plots provide useful clues about the observability of the system without the need of a more formal observability analysis.

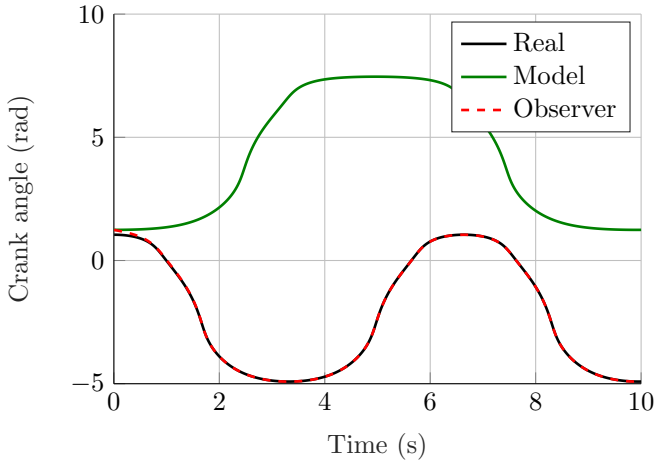



Figure 7: Angle of the crank of the four-bar mechanism obtained with the *real* mechanism, the *model* with the high modeling error, and the *observer*, in this case the errorEKF_EJ with a gyroscope on the crank providing data at 200 Hz.

The fact that this configuration is not observable with the errorEKF method is because the terms relating the variation of the acceleration with respect to position and velocity are those omitted in the Jacobian matrix of the plant in this method. This kind of behavior is expected in all the situations in which the variation of the output of the sensor with positions and/or velocities occurs only through force models, but not through the kinematics of the system.

When using accelerometers, none of the methods without force estimation provided reliable results. These results can be improved by artificially increasing the values of the covariance matrix of the measurement noise, but this was not done here to compare all the methods under the same conditions.

The method with force estimation (errorEKF_FE) provides the best results in most of the situations, but its accuracy degrades faster than the other methods when the sampling rate of the sensors is reduced. When using position sensors, all the methods provide a similar level of accuracy.

If the observers are applied to the five-bar mechanism, the results shown in figs. 9 to 11 are obtained. The trends are quite similar to those obtained with the four-bar mechanism, being the best results provided by the errorEKF_FE, and the worst behavior provided by the errorEKF, while the UKF TR and the errorEKF_EJ perform almost to the same level of accuracy in all the tests, except when considering gyroscopes on the couplers. In this last case, the UKF

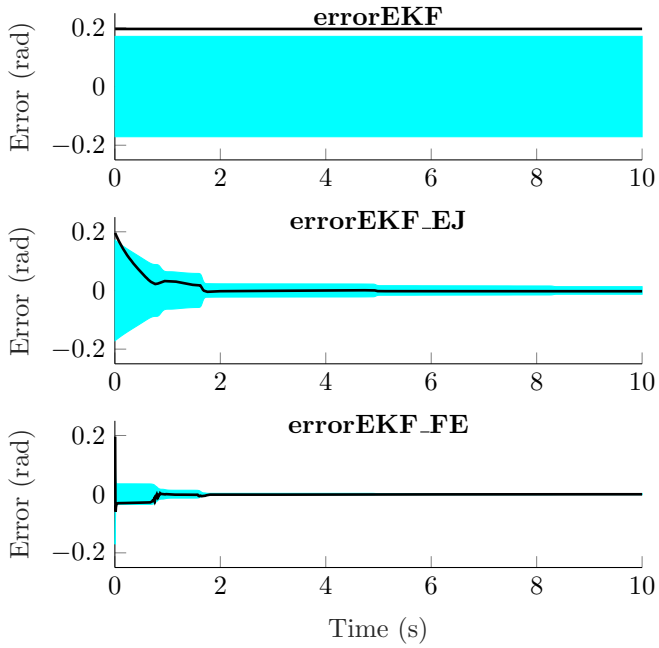



Figure 8: Tracking error of the angle of the crank of the four-bar mechanism with the different observers with the high modeling error and a gyroscope on the crank providing data at 200 Hz. The light blue shadow represents the 95 % confidence interval, computed as $\pm 1.96\sigma$, where $\sigma$ is the standard deviation calculated from the corresponding value of the covariance matrix of the state estimation uncertainty $\mathbf{P}$ (see, e.g. [37]).

TR performs a little better than the erroEKF_EJ. Again, the errorEKF_FE is the only method which provides reliable results when the sensors employed are accelerometers exclusively, but in contrast, it does not achieve better accuracy than the other methods when only position sensors are used.

The force estimation obtained with the errorEKF_FE method is shown in fig. 12. Only the four-bar mechanism with the high modeling error is shown for the sake of conciseness. The algorithm estimates the torque which would have to be applied to the crank to compensate for the gravity error. It can be seen that the force estimation using only position sensors (i.e., encoders) is quite noisy, it shows an appreciable delay, and a remarkable overshooting is produced when the force changes suddenly. The root mean squared error (RMSE) in force estimation is 8.678 Nm, while the root mean squared value of the reference torque is 9.6228 Nm (i.e. this would be the RMSE if no force estimation were performed at all). This little improvement with respect to the total force error explains why the errorEKF_FE does not provide better kinematic estimations than the methods without force estimation when position sensors are used.

The force estimation with velocity measurements is less noisy than the previous one. When the angular rate sensor is installed on the coupler, a correct

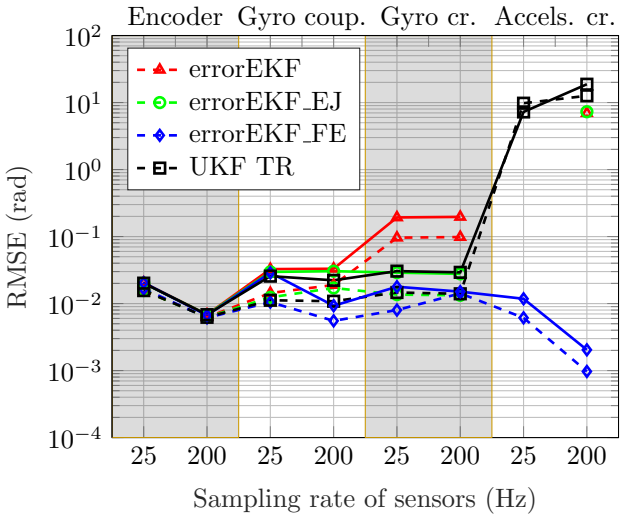



Figure 9: Position error provided by the observers for the five-bar mechanism with gravity and initial position error, evaluated as the rms of the norm of the two crank angle errors. Dashed lines represent the low modeling error (0.5 m/s$^2$ gravity error and $\pi/32$ rad initial position error) and continuous lines the high modeling error (1 m/s$^2$ gravity error and $\pi/16$ rad initial position error).

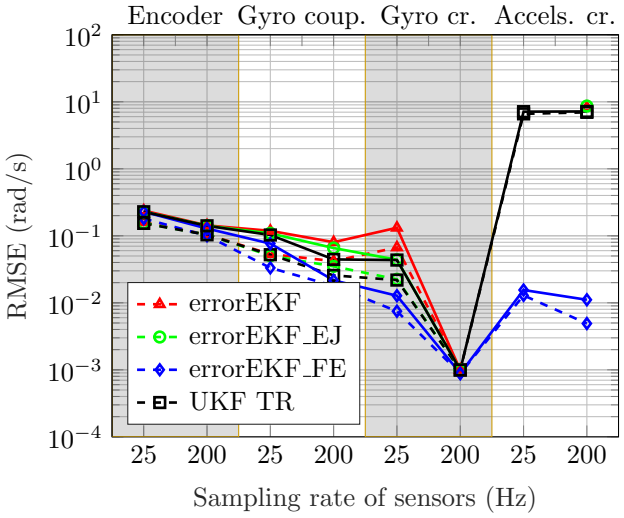



Figure 10: Velocity error provided by the observers for the five-bar mechanism with gravity and initial position error, evaluated as the rms of the norm of the two crank angular rate errors. Dashed lines represent the low modeling error (0.5 m/s$^2$ gravity error and $\pi/32$ rad initial position error) and continuous lines the high modeling error (1 m/s$^2$ gravity error and $\pi/16$ rad initial position error).

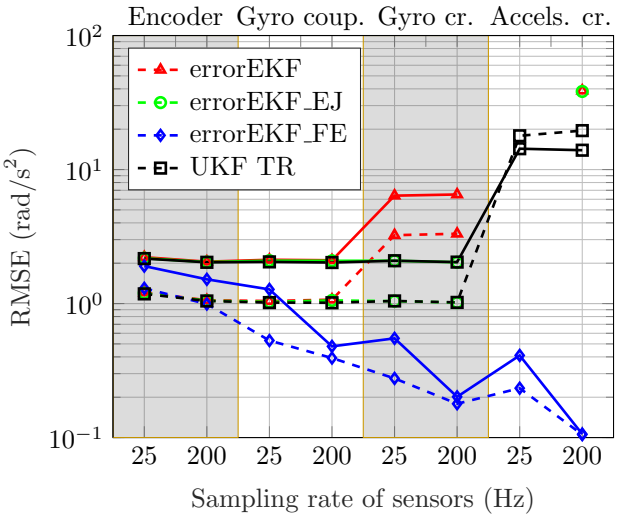



Figure 11: Acceleration error provided by the observers for the five-bar mechanism with gravity and initial position error, evaluated as the rms of the norm of the two crank angular acceleration errors. Dashed lines represent the low modeling error (0.5 m/s$^2$ gravity error and $\pi/32$ rad initial position error) and continuous lines the high modeling error (1 m/s$^2$ gravity error and $\pi/16$ rad initial position error).

force estimation is achieved since the beginning of the simulation, and the final RMSE is 2.794 Nm. When the gyroscope is installed on the crank, the force estimation is noticeably less noisy, but during the first second the force estimation is not correct, and hence the final RMSE for the whole maneuver is 2.082 Nm.

Finally, the best force estimation is obtained when the measurements come from a pair of accelerometers installed on the crank, with a RMSE of 0.893 Nm.

### *7.2. Computational cost*

A state observer has to run in real time in order to be used in any controller. Therefore, both the algorithm and the implementation have to be adequate for this purpose. The implementation presented here was thought as a benchmark, prioritizing the flexibility instead of the performance. Moreover, this comparison was made in MATLAB®, which is not suitable for hard real-time applications. For this reason, the real-time factors presented here must be taken only as a reference to compare among the different methods. The results obtained from the configuration with position sensor for both four-bar and five-bar mechanisms are presented in fig. 13. With other sampling rates, or with other sensors, the computational cost is slightly different, but the relation among the different algorithms is the same. It can be seen that the fastest method is the errorEKF with simplified Jacobians. However, the errorEKF with exact Jacobian (errorEKF_EJ) and the errorEKF with force estimation (errorEKF_FE) have only

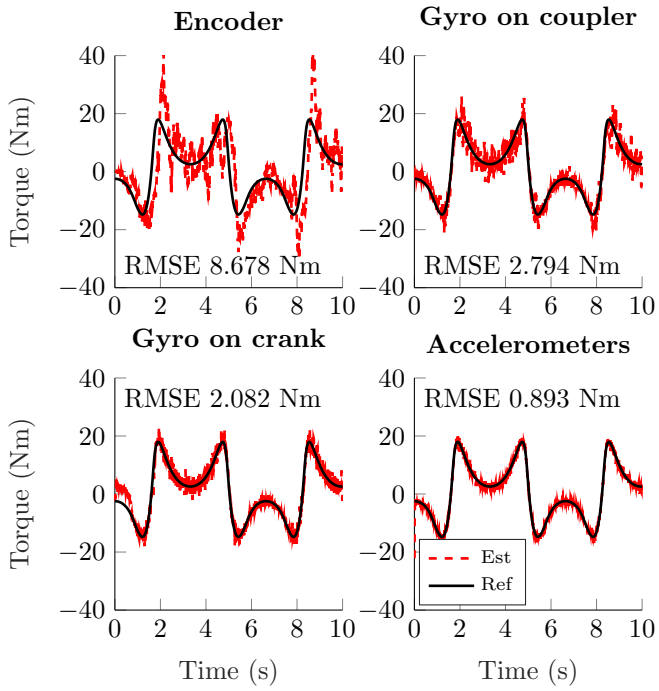



Figure 12: Force estimation for the four-bar mechanism with a modeling error of 1 m/s$^2$ in gravity and $\pi/16$ rad in initial position, sensors measurements at 200 Hz.

a little penalization. The unscented Kalman filter, however, is clearly slower than the other methods, and its computational cost increases faster with the size of the mechanism.

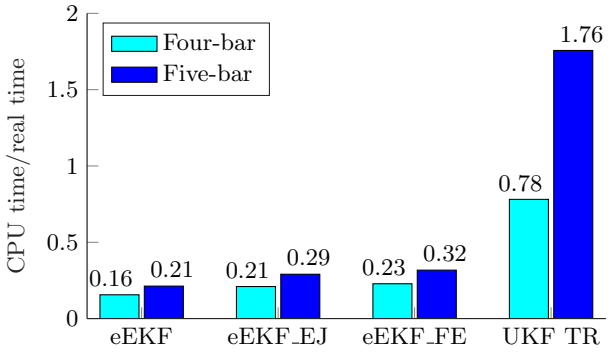



Figure 13: Real-time factor of the different methods running with position sensors at 200 Hz.

*7.3. Discussion*

From the results shown we can see that the method which provides the best results in most of the configurations tested is the method with force estimation (errorEKF_FE). Moreover, it is the only method which works reliably when only accelerometers are used as sensors. The other methods (errorEKF and errorEKF_EJ) can also be made to work with accelerometers if the covariance matrix of the measurement noise is increased for the measurements coming from accelerometers, but this was not done in the tests presented here.

If acceleration and force estimation are not desired, quite accurate results can be obtained in most configurations from the methods without force estimation. We found them easier to tune, they preserve better their accuracy when the sampling rate of the measurements is decreased, and they also have a slightly lower computational cost than the method with force estimation. From these two methods, the estimator with exact Jacobian works usually better, but the difference is small in most of the configurations. In other configurations, however, the method with exact Jacobian achieves an observable system, while the method with simplified Jacobian does not. Although the computational cost of both methods is quite similar, the method with exact Jacobian is significantly more difficult to implement due to the partial derivatives needed to calculate the Jacobian, which can become difficult to calculate, in particular if complex force models are involved.

A final consideration is the observability study: the errorEKF with simplified Jacobian has a constant transition matrix. If the measurement matrix is also constant, the observability study can be done before running any test. In the examples considered here, this happens when encoders or gyroscopes in cranks are used. For the other methods, the transition matrix is variable, and hence the observability has to be studied with methods for time-varying systems [26]. The observability study of the errorEKF can be useful for the other methods, because if a system is observable with the errorEKF, it will be also observable with the other methods, since they use more dynamic information from the system. The opposite is not true: a system can be observable with the methods with exact Jacobian, even if it is not when using the simplified method, as it happens in the tests studied here when using gyroscopes on the cranks of the mechanisms.

## 8. Conclusions and future work

This work presents two new state observers based on multibody models and the indirect Kalman filter. These methods were tested by applying them to two closed-chain mechanisms, and considering four different sets of sensors in each mechanism. Two different sampling rates for the sensors were also considered. In order to compare these new methods to other previously developed algorithms, the tests included also the errorEKF (presented in [26]) and the UKF with trapezoidal rule integration, as presented in [17].

One of the new methods, referred to as errorEKF_EJ, is a version of the previous errorEKF but using the exact Jacobian matrix of the plant. The other

method presented in this work, referred to as errorEKF_FE, follows the same structure, but it also provides input force estimation.

The errorEKF_EJ improves the estimation accuracy of the errorEKF with only a slight increase in computational cost, achieving almost the same level of accuracy as the UKF with trapezoidal rule integration, but with a much lower computational cost.

The errorEKF_FE provides even better results, and it is the only method of those considered in this work which can provide reliable estimations using only accelerometers as sensors. However, the quality of the estimations degrade faster than the other methods when the sampling rate of the sensors is reduced. The computational cost of this method is a little higher than that of the errorEKF_EJ, but still far from that of the UKF.

All the methods and multibody models tested here have been implemented in MATLAB®, and made available as Open Source[3].

In future works we will apply these methods to the multibody model of a vehicle prototype described in [28], which includes the challenge of developing the Jacobian matrices of the plant for complex forces such as the tire models, and the brake model. Moreover, we are currently studying hardware platforms for on board inplementations in automotive applications [38].

## Acknowledgements

This work has been partially financed by the Spanish Ministry of Economy and Competitiveness (MINECO) 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 'Diseño y control óptimo eficiente de sistemas multicuerpo basado en técnicas de análisis de sensibilidad' (DPI2016-81005-P), and by the Galician Government through grant ED431B2016/031.

## References

[1] J. García de Jalón, E. Bayo, Kinematic and Dynamic Simulation of Multibody Systems: The Real Time Challenge, Springer-Verlag, 1994.

[2] E. J. Haug, Computer Aided Kinematics and Dynamics of Mechanical Systems, Allyn and Bacon, 1989.

[3] O. A. Bauchau, Flexible multibody dynamics, Springer, 2011.

[4] A. A. Shabana, Computational Dynamics, 3rd Edition, Wiley, 2009.

[5] D. Dopico, A. Luaces, M. Gonzalez, J. Cuadrado, Dealing with multiple contacts in a human-in-the-loop application, Multibody System Dynamics 25 (2) (2011) 167–183. doi:10.1007/s11044-010-9230-y.

[3] See https://github.com/MBDS/mbde-matlab

[6] S.-S. Kim, W. H. Jeong, Real-time multibody vehicle model with bush compliance effect using quasi-static analysis for HILS, Multibody System Dynamics 22 (4) (2009) 367–382. doi:10.1007/s11044-009-9160-8.

[7] R. Pastorino, F. Cosco, F. Naets, W. Desmet, J. Cuadrado, Hard real-time multibody simulations using ARM-based embedded systems, Multibody System Dynamics 37 (1) (2016) 127–143. doi:10.1007/s11044-016-9504-0.

[8] D. Simon, Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches, Wiley, Hoboken, 2006.

[9] M. Grewal, A. Andrews, Kalman filtering: theory and practice using MATLAB®, Wiley, New Jersey, 2008.

[10] J. L. Blanco-Claraco, J. L. Torres-Moreno, A. Giménez-Fernández, Multibody dynamic systems as Bayesian networks: Applications to robust state estimation of mechanisms, Multibody System Dynamics 34 (2) (2015) 103–128.

[11] J. Cuadrado, D. Dopico, A. Barreiro, E. Delgado, Real-time state observers based on multibody models and the extended Kalman filter, Journal of Mechanical Science and Technology 23 (4) (2009) 894–900.

[12] J. Cuadrado, D. Dopico, J. A. Perez, R. Pastorino, Automotive observers based on multibody models and the extended Kalman filter, Multibody System Dynamics 27 (1) (2012) 3–19.

[13] R. Pastorino, D. Richiedei, J. Cuadrado, A. Trevisani, State estimation using multibody models and non-linear Kalman filter, International Journal of Non-Linear Mechanics 53 (2013) 83–90.

[14] S. Julier, J. Uhlmann, H. Durrant-Whyte, A new approach for filtering nonlinear systems, in: Proceedings of the American Control Conference, Vol. 3, 1995, pp. 1628–1632.

[15] S. Julier, J. Uhlmann, A new extension of the Kalman filter to nonlinear systems, International Symposium on Aerospace/Defense Sensing, Simulation and Controls (1997) 182–193.

[16] F. Naets, R. Pastorino, J. Cuadrado, W. Desmet, Online state and input force estimation for multibody models employing extended kalman filtering, Multibody System Dynamics 32 (2014) 317–336.

[17] J. L. Torres-Moreno, J. L. Blanco-Claraco, A. Giménez-Fernández, E. Sanjurjo, M. Á. Naya, Online Kinematic and Dynamic-State Estimation for Constrained Multibody Systems Based on IMUs, Sensors 16 (3) (2016) 333.

[18] C. Cuesta, P. Luque, D. A. Mántaras, State estimation applied to non-explicit multibody models, Nonlinear Dynamics 86 (3) (2016) 1673–1686. doi:10.1007/s11071-016-2985-9.

[19] M. Baumann, R. Leine, Synchronization-based state observer including position jumps for impacting multibody systems, in: Proceedings of the IEEE Conference on Decision and Control, Vol. 54rd IEEE, Osaka, Japan, 2015, pp. 5556–5562. doi:10.1109/CDC.2015.7403090.

[20] M. Baumann, R. Leine, Synchronization-based state observer for impacting multibody systems using switched geometric unilateral constraints, in: Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics 2015, Multibody Dynamics 2015, Barcelona, Spain, 2015, pp. 585–596.

[21] M. Baumann, R. I. Leine, A synchronization-based state observer for impact oscillators using only collision time information, International Journal of Robust and Nonlinear Control 26 (12) (2016) 2542–2563. doi:10.1002/rnc.3460.

[22] I. Palomba, D. Richiedei, A. Trevisani, Kinematic state estimation for rigid-link multibody systems by means of nonlinear constraint equations, Multibody System Dynamics (2016) 1–22.

[23] I. Palomba, D. Richiedei, A. Trevisani, Two-stage approach to state and force estimation in rigid-link multibody systems, Multibody System Dynamics 39 (1-2) (2017) 115–134. doi:10.1007/s11044-016-9548-1.

[24] A. Szczesna, P. Pruszowski, Model-based extended quaternion Kalman filter to inertial orientation tracking of arbitrary kinematic chains, SpringerPlus 5 (1) (2016) 1965. doi:10.1186/s40064-016-3653-8.

[25] J. Žumer, J. Slavič, M. Boltežar, Minimization of the positional errors for an accurate determination of the kinematic parameters of a rigid-body system with miniature inertial sensors, Mechanism and Machine Theory 81 (2014) 193–208. doi:10.1016/j.mechmachtheory.2014.07.008.

[26] E. Sanjurjo, M. Á. Naya, J. L. Blanco-Claraco, J. L. Torres-Moreno, A. Giménez-Fernández, Accuracy and efficiency comparison of various nonlinear Kalman filters applied to multibody models, Nonlinear Dynamicsdoi:10.1007/s11071-017-3354-z.

[27] E. Sanjurjo, State observers based on detailed multibody models applied to an automobile, Phd. thesis, University of A Coruña (2016).
URL http://hdl.handle.net/2183/17963

[28] R. Pastorino, E. Sanjurjo, A. Luaces, M. Á. Naya Villaverde, W. Desmet, J. Cuadrado, Validation of a Real-Time Multibody Model for an X-by-Wire Vehicle Prototype Through Field Testing, Journal of Computational and Nonlinear Dynamics 10 (3) (2015) 031006.

[29] P. D. Groves, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Artech House, London, 2008.

[30] S. Roumeliotis, G. Sukhatme, G. Bekey, 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, IEEE, 1999, pp. 1656–1663.

[31] J. Cuadrado, R. Gutierrez, M. Á. Naya Villaverde, P. Morer, A comparison in terms of accuracy and efficiency between a MBS dynamic formulation with stress analysis and a non-linear FEA code, International Journal for Numerical Methods in Engineering 51 (9) (2001) 1033–1052.

[32] J. Cuadrado, D. Dopico, M. Á. Naya Villaverde, Penalty, semi-recursive and hybrid methods for MBS real-time dynamics in the context of structural integrators, Multibody System Dynamics 12 (2) (2004) 117–132.

[33] D. Dopico, F. González, J. Cuadrado, J. Kövecses, Determination of Holonomic and Nonholonomic Constraint Reactions in an Index-3 Augmented Lagrangian Formulation With Velocity and Acceleration Projections, Journal of Computational and Nonlinear Dynamics 9 (4) (2014) 041006.

[34] J. Cuadrado, D. Dopico, M. Á. Naya Villaverde, M. González, Real-Time Multibody Dynamics and Applications, in: Simulation Techniques for Applied Dynamics, no. 507, 2008, pp. 247–311. doi:10.1007/978-3-211-89548-1_6.

[35] D. Dopico, Y. Zhu, A. Sandu, C. Sandu, Direct and Adjoint Sensitivity Analysis of Ordinary Differential Equation Multibody Formulations, Journal of Computational and Nonlinear Dynamics 10 (1) (2014) 011012.

[36] C. F. Van Loan, Computing Integrals Involving the Matrix Exponential 23 (3) (1978) 395–404.

[37] W. Navidi, Statistics for Engineers and Scientists, 4th Edition, McGraw-Hill Education, New York, 2015.

[38] A. J. Rodríguez, R. Pastorino, M. Á. Naya, E. Sanjurjo, Virtual sensing on an automotive embedded heterogeneous platforms, in: 15th EAEC European Automotive Congress, Madrid, Spain, 2017.

## Appendix A Gyroscope model

In this work, the gyroscopes (angular rate sensors) are modeled from two points $i$ and $j$ of a solid as shown in fig. A.1.

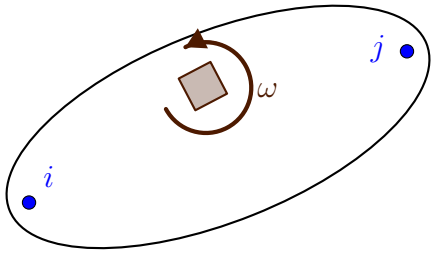



Figure A.1: Gyroscope model attached to a solid defined by points $i$ and $j$.

The angular rate $\omega$ sensed by this sensor can be written as follows:

$$\omega = \frac{\left(\dot{x}_j - \dot{x}_i\right)\left(y_i - y_j\right) + \left(\dot{y}_j - \dot{y}_i\right)\left(x_j - x_i\right)}{d_{ij}^2} \tag{A.1}$$

where $d_{ij}^2 = \left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2$. Note that the angular rate of a rigid body is the same at any point, and hence the gyroscope output is independent of its position. For a model of a three-axis gyroscope installed on a 3D solid, refer to [27].

## Appendix B Accelerometer model

The accelerometers are modeled from two points $i$ and $j$ of a solid as shown in fig. B.1.

The acceleration $\mathbf{a}$ of the point where the accelerometer is attached can be written as follows:

$$\mathbf{a} = \mathbf{a}_i + \alpha\left|\mathbf{r}\right|\mathbf{u}_\mathbf{r}^\perp - \omega^2\left|\mathbf{r}\right|\mathbf{u}_\mathbf{r} \tag{B.1}$$

where $\mathbf{a}_i$ is the acceleration of the point $i$, angular velocity $\omega$ is calculated using eq. (A.1), and angular acceleration $\alpha$ is calculated as follows:

$$\alpha = \frac{\left(\ddot{x}_j - \ddot{x}_i\right)\left(y_i - y_j\right) + \left(\ddot{y}_j - \ddot{y}_i\right)\left(x_j - x_i\right)}{d_{ij}^2} \tag{B.2}$$

However, to calculate the actual accelerometer output $a_{acc}$, first the acceleration of the gravity $\mathbf{g}$ must be subtracted, because accelerometers cannot measure it, and finally, the acceleration have to be projected over the sensitive axis of the accelerometer $\mathbf{u}_{acc}$, as follows:

$$a_{acc} = \left(\mathbf{a} - \mathbf{g}\right) \cdot \mathbf{u}_{acc} \tag{B.3}$$

If the general expression for a three-axis accelerometer installed on a 3D body is desired, refer to [27].

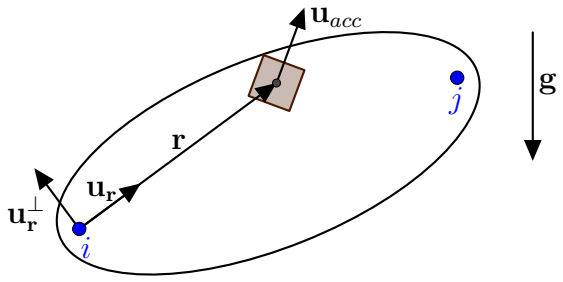



Figure B.1: Accelerometer model attached to a solid defined by points $i$ and $j$.