

## State Estimation Using Multibody Models and Nonlinear Kalman Filters

**Roland Pastorino\***, **Dario Richiedei<sup>#</sup>**, **Javier Cuadrado\***, **Alberto Trevisani<sup>#</sup>**

\* Mechanical Engineering Laboratory  
University of La Coruña  
c/ Mendizábal s/n, 15403 Ferrol, Spain  
roland.pastorino@udc.es  
javicuad@udc.es

<sup>#</sup> DTG- Department of Management and Engineering  
Università degli Studi di Padova  
Stradella S. Nicola 3, 36100 Vicenza, Italy  
dario.richiedei@unipd.it  
alberto.trevisani@unipd.it

### ABSTRACT

The aim of this work is to provide a thorough research on the implementation of some nonlinear Kalman Filters (KF) using multibody (MB) models and to compare their performances in terms of accuracy and computational cost. The filters considered in this study are the extended KF (EKF) in its continuous form, the unscented KF (UKF) and the spherical simplex unscented KF (SSUKF). Two different MB formulations are taken into consideration to convert the differential algebraic equations (DAE) of the MB model into the ordinary differential equations (ODE) required by the filters: the state-space reduction method known as projection matrix–R method and the penalty formulation. Additionally, both implicit and explicit integration schemes are used to evaluate the impact of explicit integrators over implicit integrators, commonly employed in MB simulations, in terms of accuracy, stability and computational cost, commonly employed in MB simulations. However, state estimation through KFs is a closed-loop estimation correcting the model drift according to the difference between the predicted measurement and the actual measurement, what limits the interest in using implicit integrators. Performance comparisons of all the aforementioned nonlinear observers have been carried out in simulation on a 5-bar linkage. The mechanism parameters have been obtained from an experimental 5-bar linkage and the sensor characteristics from off-the-shelf sensors to reproduce a realistic simulation. The results should highlight useful clues for the choice of the most suitable filters, MB formulations and integration schemes.

### 1 INTRODUCTION

In recent years, the interest in state estimation in mechanical systems, and in particular in MB systems, has increased. As a matter of fact, the knowledge of the system state allows synthesizing effective state controllers, replacing expensive sensors with virtual sensors, improving reliability by making the controlled system robust in relation to sensor fault. On the one hand, many works in the literature address the synthesis of optimal observers for linear mechanical systems through the KF (linear Kalman Filter). On the other hand, when nonlinear mechanical systems are considered, only sub-optimal approaches based on the LKF (Linearized KF) have usually been adopted to ensure high-frequency and hard real-time estimation [1]. Indeed, up to now, the use of other types of nonlinear observers in MB systems has only been investigated marginally. This lack of use is mainly due to the difficulties in performing fast integration of the nonlinear equations of motion, which usually involve high frequency dynamics and severe nonlinearities. In [2], it is shown how the recent improvements in MB dynamics raise the possibility of employing complex models in real-time state observers. In this work, the estimation was performed through the EKF in its continuous form. Generally speaking, the EKF is the most widely used algorithm for nonlinear estimation. However, when nonlinearities are severe, EKF often gives unreliable or divergent estimates. In addition, the linearization requires a Jacobian matrix which could either be difficult to calculate or not exist. Implementation difficulties are particularly relevant if the system model is represented by DAE (Differential Algebraic Equations), as it is common in the MB field.

Recent developments in Kalman filtering algorithms make possible to overcome part of the aforementioned EKF shortcomings. The SPKFs (Sigma-Point Kalman Filters), also called LRKFs (Linear Regression Kalman Filters), take advantage of a set of deterministically calculated weighted samples (referred to as sigma-points, or regression points [4]) to capture at least the first and second order moments of the actual state probability distribution. This approximation made by the SPKFs is therefore more accurate than the EKF linearization, and it does not require the calculation of Jacobian matrices. Different sigma-point set definitions lead to different filter characteristics, allowing priority to be given to either estimation accuracy

or computational efficiency. The most relevant variants are the UKF, the CDKF (Central Difference KF) [4], the SSUKF [5] and their respective numerically stable forms, the so-called square-root forms. Therefore, a natural approach to overcome the EKF problems and to improve the estimation in MB systems would be to use SPKFs. To the best of the authors' knowledge, SPKFs have never been applied to the estimation of MB models.

## 2 NONLINEAR KALMAN FILTERS

The original formulation of the KF is intended only to linear systems. To apply the KF to nonlinear systems, the process and measurement models have to be approximated in some way. The differences between the KF variants for nonlinear systems arise mainly from distinct approximation approaches. In this section, two types of nonlinear Kalman filters are discussed: the EKF, which is the *de facto* KF for nonlinear systems, and the SPKFs. For the EKF, only the continuous-time form is taken into account while only the discrete-form is considered for the SPKFs.

### 2.1 THE KALMAN FILTER

The estimation problem for this filter consists in estimating the state of a linear stochastic system. Both continuous-time and discrete-time forms of the equations are described below. First, the continuous-time equations for the system dynamics and the measurement model are presented in Equations (1) and (2) respectively. In this filter, the system dynamics are described by *first order ordinary differential equations (ODE) with independent states*. For simplicity's sake, the process noises  $\mathbf{w}(t)$  and the measurement noises  $\mathbf{v}(t)$  have been considered additive white zero-mean Gaussian noises.

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t) + \mathbf{w}(t) \quad \mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(t)) \quad (1)$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)) + \mathbf{v}(t) \quad \mathbf{v}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}(t)) \quad (2)$$

where  $\mathbf{x}(t)$  represents the states of the system,  $\dot{\mathbf{x}}(t)$  is the time derivative of the state vector,  $\mathbf{f}$  is the dynamics system function,  $\mathbf{y}(t)$  contains the measurements and  $\mathbf{h}$  is the measurement sensitivity matrix. The differential equation for the recursive estimation of the states is presented in Equation (3).

$$\dot{\hat{\mathbf{x}}} = \mathbf{E}[\mathbf{f}(\hat{\mathbf{x}}(t), t) + \mathbf{w}(t)] + \bar{\mathbf{K}}(t)[\mathbf{y}(t) - \hat{\mathbf{y}}(t)] = \mathbf{F}\hat{\mathbf{x}}(t) + \bar{\mathbf{K}}(t)[\mathbf{y}(t) - \hat{\mathbf{y}}(t)] \quad (3)$$

where  $\mathbf{E}$  is the expectation,  $\bar{\mathbf{K}}(t)$  is the Kalman gain matrix,  $\hat{\mathbf{y}}(t)$  are the predicted measurements and  $\mathbf{F}(t)$  is the linearization of  $\mathbf{f}(t)$  (here  $\mathbf{F}(t) = \mathbf{f}'(t)$  as the latter is linear). The Kalman gain and the predicted measurements are calculated using Equations (4) and (5).

$$\bar{\mathbf{K}}(t) = \mathbf{P}(t)\mathbf{H}^T(t)\mathbf{R}^{-1}(t) \quad (4)$$

$$\hat{\mathbf{y}}(t) = \mathbf{E}[\mathbf{h}(\hat{\mathbf{x}}(t), t) + \mathbf{v}(t)] = \mathbf{H}\hat{\mathbf{x}}(t) \quad (5)$$

where  $\mathbf{H}$  is the linearization of  $\mathbf{h}$  (here  $\mathbf{H}(t) = \mathbf{h}'(t)$  as the latter is linear) and  $\mathbf{P}(t)$  is the covariance matrix of state estimation uncertainty which is obtained using differential Equation (6), also called *Riccati equation*.

$$\dot{\mathbf{P}} = \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T - \bar{\mathbf{K}}\mathbf{R}\bar{\mathbf{K}}^T + \mathbf{G}(t)\mathbf{Q}\mathbf{G}^T(t) \quad (6)$$

where  $\mathbf{G}(t)$  is the coupling matrix between the process noises and the states of the system. The case in which  $\mathbf{f}$  is a nonlinear function will be treated in 2.2.

The estimation problem can also be formulated in discrete-time form which is the most suitable form for hardware implementation as the estimation is divided in two parts depending on the availability of sensor information. The system dynamics and measurement model, which are now *linear difference equations with independent states*, are shown in Equations (7) and (8). The same assumptions mentioned above have been made for the process and measurement noises.

$$\mathbf{x}_{k+1} = \phi_k(\mathbf{x}_k) + \mathbf{w}_k \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \quad (7)$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \quad (8)$$

where  $k$  represents the time step and  $\phi_k$  is the state transition matrix. The estimates of the states and the covariance matrix of state estimation uncertainty are now calculated using Equation (9) and (10).

$$\hat{\mathbf{x}}_{k+1}^- = E[\phi_k(\hat{\mathbf{x}}_k) + \mathbf{w}_k] = \Phi_k \hat{\mathbf{x}}_k \quad (9)$$

$$\mathbf{P}_{k+1}^- = E[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^-)(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^-)^T] = \Phi_k \mathbf{P}_k \Phi_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \quad (10)$$

where  $\Phi_k$  is the linearization of  $\phi_k$  (here  $\Phi_k = \phi_k$  as  $\phi_k$  is linear),  $(-)$  indicates the *a priori* values of the variables (i.e. before the measurement information is used) and  $(+)$  indicates the *a posteriori* values (i.e. after the measurement information is used). To calculate the *a posteriori* estimates of the state, the estimation is split in two distinct parts: the time–update and the measurement–update parts. When no information is available from the sensors, the estimation relies only on the system’s model, this is the time–update. In this case the posterior state estimates and the posterior values of  $\mathbf{P}_k$  are obtained using Equation (11) and (12).

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^- \quad (11)$$

$$\mathbf{P}_{k+1} = \mathbf{P}_{k+1}^- \quad (12)$$

If some information is available from the sensors, the estimation relies now on both the system’s model and the sensors’ information; this is the measurement update, which provides a closed loop correction of the estimates. The *a posteriori* state estimates and the *a posteriori* covariance matrix are now obtained using Equation (13) and (14).

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \bar{\mathbf{K}}_{k+1}(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-) \quad (13)$$

$$\mathbf{P}_{k+1} = \mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \bar{\mathbf{K}}_{k+1} \mathbf{H}_{k+1} \mathbf{P}_{k+1}^- \quad (14)$$

The Kalman gain and the predicted measurements are calculated with Equation (15) and (16) respectively.

$$\begin{aligned} \bar{\mathbf{K}}_{k+1} &= E[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^-)(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-)^T] \times E[(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-)(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-)^T]^{-1} \\ &= \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T + \mathcal{R})^{-1} \end{aligned} \quad (15)$$

$$\hat{\mathbf{y}}_{k+1}^- = E[\mathbf{h}_{k+1}(\hat{\mathbf{x}}_{k+1}^-) + \mathbf{v}_k] = \mathbf{H}_{k+1} \hat{\mathbf{x}}_{k+1}^- \quad (16)$$

## 2.2 THE EXTENDED KALMAN FILTER

The EKF is the *de facto* KF for nonlinear systems. This filter propagates Gaussian random variables (GRVs) through the system dynamics represented by either the nonlinear continuous–time Equation (1) or the nonlinear discrete–time Equation (7). Unlike in the KF, here  $\mathbf{f}(t)$ ,  $\mathbf{h}(t)$ ,  $\phi_k$  and  $\mathbf{h}_k$  can be nonlinear functions. While the optimal quantities can be calculated in Equations (3), (5), (9) and (16) for linear systems, this is not the case for nonlinear systems. As a consequence, the optimal terms have to be approximated in some way. The EKF approximates first the continuous–time Equation (3) by Equation (17) where the derivative of the state vector is the function of the latter and then Equation (5) by Equation (18), where the predicted measurements are the function of the estimates. As for the KF, the Kalman gain is obtained using Equation (4) and the covariance matrix of the state estimation uncertainty using Equation (6) but the linearizations  $\mathbf{F}$  and  $\mathbf{H}$  have to be calculated as shown in Equation (19).

$$\dot{\hat{\mathbf{x}}} = E[\mathbf{f}(\hat{\mathbf{x}}(t), t) + \mathbf{w}(t)] + \bar{\mathbf{K}}(t)[\mathbf{y}(t) - \hat{\mathbf{y}}(t)] \simeq \mathbf{f}(\hat{\mathbf{x}}(t)) + \bar{\mathbf{K}}(t)[\mathbf{y}(t) - \hat{\mathbf{y}}(t)] \quad (17)$$

$$\hat{\mathbf{y}}(t) = E[\mathbf{h}(\hat{\mathbf{x}}(t), t) + \mathbf{v}(t)] \simeq \mathbf{h}(\hat{\mathbf{x}}(t)) \quad (18)$$

$$\dot{\mathbf{P}}(t) \simeq \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}\mathbf{F}^T(t) - \bar{\mathbf{K}}\mathcal{R}\bar{\mathbf{K}}^T + \mathbf{G}(t)\mathcal{Q}\mathbf{G}^T(t) \quad (19)$$

## 2.3 THE SIGMA–POINT KALMAN FILTERS

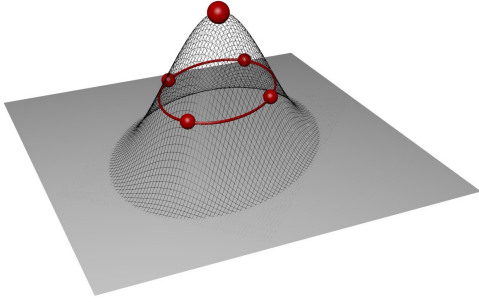
The SPKFs are KFs for nonlinear systems with a linearization approach substantially different from the one of the EKF [4]. Instead of first approximating the estimates as function of the prior estimates propagated through the nonlinear system function and then linearizing the dynamic equations to determine the covariances, a set of deterministically chosen weighted sample points is propagated through the nonlinear system function. The sample points, called *sigma–points*, capture at least the first two moments (mean and covariance) of the prior and posterior (i.e. after propagation through the nonlinear function) random variables. Two SPKF versions are briefly reminded hereafter: the UKF and the SSUKF. All the equations and the most important characteristics of these filters when used with MB models are presented.

### 2.3.1 The unscented Kalman filter

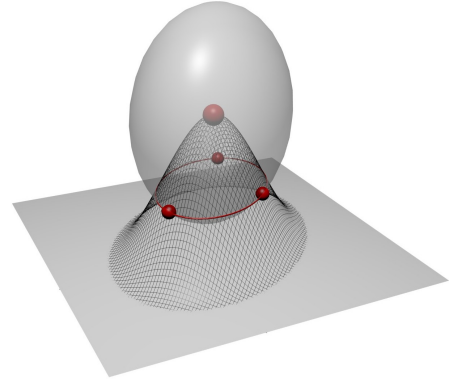
As previously mentioned, the time-update equations of the UKF differ substantially from the ones of the EKF. Firstly, the set of  $n_{sp}$  sigma-points for the UKF has to be calculated, where  $n_{sp} = (2L + 1)$  points ( $L$  is the dimension of the state vector). The zeroth point is the unchanged state estimate while the rest of points is calculated using the zeroth point and the square-root decomposition of the covariance matrix of state estimation uncertainty as demonstrated in Equation (20).

$$\chi_k(i) = \begin{cases} \hat{\mathbf{x}}_k & i = 0 \\ \hat{\mathbf{x}}_k + \gamma (\sqrt{\mathbf{P}_k})_i & i = 1, \dots, L \\ \hat{\mathbf{x}}_k - \gamma (\sqrt{\mathbf{P}_k})_i & i = L + 1, \dots, 2L \end{cases} \quad (20)$$

where  $\chi_k(i)$  is the  $i$ th sigma-point,  $\gamma = \sqrt{L + \lambda}$ ,  $\lambda = \alpha^2(L + \kappa)$ ,  $\alpha$  and  $\kappa$  are user-defined tuning parameters,  $\sqrt{\cdot}$  is the matrix square-root using lower triangular Cholesky decomposition and  $(\cdot)_i$  represents the  $i$ th column.  $0 < \alpha \leq 1$  is a scaling factor defining the extension of the spread of the sigma-points around the mean of the estimates.  $\kappa$  is another scaling factor, usually set to 0. Figure 1 exemplifies the sigma-point set for a 2-dimensional Gaussian random variable ( $n_{sp} = 5$ ). The size of the points represents their weights.



**Figure 1.** UKF: weighted sigma-points for a 2 dimensional Gaussian random variable



**Figure 2.** SSUKF: weighted sigma-points for a 2 dimensional Gaussian random variable

After that, each sigma-point is propagated through the system dynamics Equation (7) as illustrated in Equation (21). There are as many function evaluations as sigma-points. It is worth pointing out that these function evaluations are independent from each other allowing to parallelize the computations using several cores for example.

$$\chi_{k+1}^- = \phi_k(\chi_k) \quad (21)$$

The *a priori* state estimates and covariance matrix of state estimation uncertainty are approximated by taking the weighted mean and covariances of the propagated sigma-points as shown in Equations (22) and (23).

$$\hat{\mathbf{x}}_{k+1}^- = E[\phi_k(\hat{\mathbf{x}}_k) + \mathbf{w}_k] \simeq \sum_{i=0}^{n_{sp}-1} w_i^m \chi_{k+1}^-(i) \quad (22)$$

$$\hat{\mathbf{P}}_{k+1}^- = E[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^-)(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^-)^T] \simeq \sum_{i=0}^{n_{sp}-1} w_i^c (\chi_{k+1}^-(i) - \hat{\mathbf{x}}_{k+1}^-)(\chi_{k+1}^-(i) - \hat{\mathbf{x}}_{k+1}^-)^T \quad (23)$$

where  $w_0^m = \lambda / (L + \lambda)$ ,  $w_0^c = w_0^m + (1 - \alpha^2 + \beta)$ ,  $w_i^c = w_i^m = 1 / [2(L + \lambda)]$  for  $i = 1 \dots n_{sp} - 1$  and  $\beta$  is a scaling factor used to control the weighting of the zeroth sigma-point. Then, if no information is available from the sensors, Equations (11) and (12) are used.

If some information is available, the measurement-update equations, which also differ substantially from the ones of the EKF, are employed. The Kalman gain is obtained using the weighted covariances as shown

in Equation (24).

$$\begin{aligned}\bar{\mathbf{K}}_{k+1} &= \mathbb{E}[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^-)(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-)^T] \cdot \mathbb{E}[(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-)(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-)^T]^{-1} = \mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} \mathbf{P}_{\mathbf{y}_k \mathbf{y}_k} \\ &= \sum_{i=0}^{n_{sp}-1} w_i^c (\boldsymbol{\chi}_{k+1}^-(i) - \hat{\mathbf{x}}_{k+1}^-)(\boldsymbol{\mathcal{Y}}_{k+1}^-(i) - \hat{\mathbf{y}}_{k+1}^-)^T \cdot \sum_{i=0}^{n_{sp}-1} w_i^c (\boldsymbol{\mathcal{Y}}_{k+1}^-(i) - \hat{\mathbf{y}}_{k+1}^-)(\boldsymbol{\mathcal{Y}}_{k+1}^-(i) - \hat{\mathbf{y}}_{k+1}^-)^T\end{aligned}\quad (24)$$

Finally the predicted measurements are approximated by the weighted means of the estimates propagated through the measurement sensitivity matrix  $\mathbf{h}_k$ , as can be seen in Equations (25) and (26). The *a posteriori* covariance matrix is now obtained using Equation (27) and the *a posteriori* state estimates using Equation (13).

$$\hat{\mathbf{y}}_{k+1}^- = \mathbb{E}[\mathbf{h}_{k+1}(\hat{\mathbf{x}}_{k+1}^-) + \mathbf{v}_k] \simeq \sum_{i=0}^{n_{sp}-1} w_i^c \boldsymbol{\mathcal{Y}}_{k+1}^-(i) \quad (25)$$

$$\boldsymbol{\mathcal{Y}}_{k+1}^- = \mathbf{h}_{k+1}(\boldsymbol{\chi}_{k+1}) \quad (26)$$

$$\mathbf{P}_{k+1} = \mathbf{P}_{k+1}^- - \bar{\mathbf{K}}_{k+1} \mathbf{P}_{\mathbf{y}_k \mathbf{y}_k} \bar{\mathbf{K}}_{k+1}^T \quad (27)$$

### 2.3.2 The spherical simplex unscented Kalman filter

The structure and the equations of the SSUKF are similar to those of the UKF [5]. The first difference between the two filters is the rule adopted for the selection of the sigma-point set, and consequently their number:  $n_{sp} = (L + 2)$  for the SSUKF and  $n_{sp} = (2L + 1)$  for the UKF. As a consequence, the number of function evaluations is smaller for this filter, meaning that the computational cost is reduced. The sigma-point set for the SSUKF is calculated using Equation (28).

$$\boldsymbol{\chi}_i^j = \begin{cases} \begin{bmatrix} \boldsymbol{\chi}_0^{j-1} \\ \mathbf{0} \end{bmatrix} & \text{for } i = 0 \\ \begin{bmatrix} \boldsymbol{\chi}_i^{j-1} \\ 1 \\ -\frac{1}{\sqrt{j(j+1)w_1}} \end{bmatrix} & \text{for } i = 1, \dots, j \\ \begin{bmatrix} \mathbf{0}_{j-1} \\ 1 \\ \frac{1}{\sqrt{j(j+1)w_1}} \end{bmatrix} & \text{for } i = j + 1 \end{cases} \quad (28)$$

where  $j = 2, \dots, n$ ,  $w_0$  is the weight of the zeroth sigma-point and also a user-defined parameter that affects the fourth and higher moments of the sigma-points set ( $0 \leq w_0 \leq 1$ ),  $w_i$  are the weights of the rest of sigma-points ( $w_i = (1 - w_0)/(n + 1)$ ) and finally the initial values to calculate the set of sigma-points are:  $\boldsymbol{\chi}_0^1 = \mathbf{0}$ ,  $\boldsymbol{\chi}_1^1 = -\frac{1}{\sqrt{2w_1}}$  and  $\boldsymbol{\chi}_2^1 = \frac{1}{\sqrt{2w_1}}$ . All the points (apart from the zeroth point) lie on a hypersphere, the radius of which is  $\sqrt{L}/(1 - w_0)$ , as it is exemplified in Figure 2. The second difference with respect to the UKF is related to the weights used in Equations (22) (23) and (24) where only one set of weights is employed:  $w_i$  for  $i = 0, \dots, n_{sp} - 1$ .

## 3 MB FORMULATIONS

The dynamics of an MB system can be represented by the Lagrange's Equations (29) which lead to a set of *second order differential algebraic equations (DAE) of index 3*.

$$\begin{aligned}\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_q^T \boldsymbol{\lambda} &= \mathbf{Q} \\ \boldsymbol{\Phi} &= \mathbf{0}\end{aligned}\quad (29)$$

where  $\mathbf{M}$  is the positive semidefinite mass matrix,  $\ddot{\mathbf{q}}$  is the vector of dependent accelerations,  $\boldsymbol{\Phi}$  is the vector of constraints,  $\boldsymbol{\Phi}_q$  is the Jacobian matrix of the constraints,  $\boldsymbol{\lambda}$  are the Lagrange multipliers and  $\mathbf{Q}$  contains the external forces, the velocity-dependent inertia forces and those obtained from a potential. These equations are not independent from each other as they are linked by the constraint equations. However, with a

view to employ MB models in state estimation, the KF approach requires the equations to be independent from each other; in other words the equations of motion have to be ODEs. In this research, two different MB formulations have been employed to convert the DAEs into ODEs: the state-space reduction method known as the projection matrix–R method and the penalty formulation [3]. Both formulations eliminate the Lagrange multipliers related to the restrictions in such a way that the variables behave as if they were independent.

### 3.1 STATE–SPACE REDUCTION METHOD – MATRIX R METHOD

This method transforms the DAEs expressed in dependent variables into ODEs expressed in independent variables. Therefore, the ODE’s dimension ( $n_i$ ) is equal to the number of degrees of freedom (DOF) of the system. Equation (30), the first base equation of this method, relates the vector of dependent velocities  $\dot{\mathbf{q}}$  (dimension  $n_d$ ) to the independent vector  $\dot{\mathbf{z}}$  (dimension  $n_i$ ,  $n_i < n_d$ ).

$$\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{z}} \quad (30)$$

Matrix  $\mathbf{R}$  can be calculated first using the derivative of the constraint equations, as shown in Equation (31) and then expressing the dependent velocities as linear combinations of the independent ones.

$$\Phi_{\mathbf{q}}\dot{\mathbf{q}} = \mathbf{0} \quad (31)$$

The second base equation, Equation (32), is the time–derivative of Equation (30).

$$\ddot{\mathbf{q}} = \mathbf{R}\ddot{\mathbf{z}} + \dot{\mathbf{R}}\dot{\mathbf{z}} \quad (32)$$

Finally, the equations of motion, shown in Equation (33), are derived by premultiplying Equation (29) by  $\mathbf{R}^T$  and using Equation (31) to remove the Lagrange multipliers.

$$\ddot{\mathbf{z}} = (\mathbf{R}^T\mathbf{M}\mathbf{R})^{-1} \left[ \mathbf{R}^T(\mathbf{Q} - \mathbf{M}\ddot{\mathbf{R}}\dot{\mathbf{z}}) \right] = \bar{\mathbf{M}}^{-1}\bar{\mathbf{Q}} \quad (33)$$

The formulations in independent coordinates, like the one obtained through the matrix–R method, have the advantage of reducing considerably the number of equations to be integrated ( $n_i$ ). However this imposes an increase in the computational cost since the position and velocity problems have to be solved after each function evaluation.

### 3.2 PENALTY FORMULATION

This second method converts the DAEs into ODEs by introducing into the equations of motion the constraint equations as a dynamical system penalized by large factors. The rigid constraints in the DAEs are converted into constraints that can be violated in the ODEs, but only in a small amount in order to still represent the DAEs with negligible approximation errors. The Lagrange multipliers are approximated using the dynamical system demonstrated in Equation (34), meaning that the constraint forces depend on the constraint violations.

$$\boldsymbol{\lambda} \simeq \boldsymbol{\alpha}(\ddot{\Phi} + 2\zeta\boldsymbol{\omega}\dot{\Phi} + \boldsymbol{\omega}^2\Phi) \quad (34)$$

where the vectors  $\boldsymbol{\alpha}$ ,  $\boldsymbol{\zeta}$  and  $\boldsymbol{\omega}$  contain the penalty factors (usually  $> 10^7$ , its dimension depends on the type of constraints), the dimensionless damping ratios (usually  $\simeq 1$ ) and the natural frequencies (usually  $\simeq 10$  rad/s) for each constraint. Finally, the equations of motion (Equation (35)) are obtained by replacing the Lagrange multipliers of Equation (34) in Equation (29).

$$\ddot{\mathbf{q}} = (\mathbf{M} + \Phi_{\mathbf{q}}^T\boldsymbol{\alpha}\Phi)^{-1} \left[ \mathbf{Q} - \Phi_{\mathbf{q}}^T\boldsymbol{\alpha}(\dot{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} + 2\zeta\boldsymbol{\omega}\dot{\Phi} + \boldsymbol{\omega}^2\Phi) \right] \quad (35)$$

This formulation has a greater number of equations ( $n_d$ ) to be integrated than matrix R method ( $n_i$ ). Nevertheless it is not necessary to solve the position and velocity problems to pass from independent to dependent coordinates after each function evaluation.

#### 4 STATE ESTIMATION USING NONLINEAR KALMAN FILTERS WITH MB MODELS

In the previous sections, several types of nonlinear Kalman filters have been reminded as well as two different MB formulations. In order to employ MB models with these filters, the analytical equations of motion have to be reshaped or integrated before using them as system dynamics equations of the filters. Various approaches will be treated in this section

Two integration schemes have been considered: the trapezoidal rule (TR), an implicit scheme shown in Equation (36), and the second order explicit Runge–Kutta method (RK2) presented in Equation (37).

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \frac{\Delta t}{2}(\dot{\mathbf{v}}_k + \dot{\mathbf{v}}_{k+1}) \quad (36)$$

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \frac{\Delta t}{2}(\mathbf{k}_1 + \mathbf{k}_2) \quad \text{with} \quad \mathbf{k}_1 = \mathbf{f}(t_k, \mathbf{v}_k) \text{ and } \mathbf{k}_2 = \mathbf{f}(t_k + \Delta t, \mathbf{v}_k + \Delta t \mathbf{k}_1) \quad (37)$$

where  $\mathbf{v}(t) = \begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{bmatrix}$  and  $\mathbf{f}(t)$  is a system of *first-order ODE* equations coming from reshaping the *second-order ODE* equations obtained with either the matrix–R method or the penalty formulation and using variable duplication. Equation (38) demonstrates how  $\mathbf{f}(t)$  is calculated.

$$\ddot{\mathbf{q}} = \mathcal{F}(t, \mathbf{q}, \dot{\mathbf{q}}) \Rightarrow \dot{\mathbf{v}}(t) \equiv \begin{bmatrix} \dot{\mathbf{w}}(t) \\ \mathbf{w}(t) \end{bmatrix} = \begin{bmatrix} \mathcal{F}(t, \mathbf{q}, \mathbf{w}) \\ \dot{\mathbf{q}} \end{bmatrix} \equiv \mathbf{f}(\mathbf{v}(t), t) \quad (38)$$

If an MB formulation in independent coordinates is used,  $\mathbf{q}$  has to be replaced by  $\mathbf{z}$  in the previous equations. Implicit integrators are often employed in MB simulations to overcome stability problems and energy loss while explicit integrators are commonly used in applications requiring hard real–time.

##### 4.1 THE EXTENDED KALMAN FILTER WITH MB MODELS

In previous researches, the EKF in its continuous–time form has been employed with MB models [2]. The corresponding equations are briefly reminded in order to be compared with the other approaches presented in this paper. Equation (33) can be reshaped using variable duplication when defining the state vector (shown in Equation (39)) in order to convert the *second order ODE* (Equation (33)) into a *first order ODE* (Equation (40)). As a consequence, the latter equation and Equation (1) have now the same form meaning that the EKF equations in continuous–time can be applied. The resulting equations are then integrated by means of the TR.

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{z}(t) \\ \mathbf{w}(t) \end{bmatrix} \text{ where } \mathbf{w}(t) = \dot{\mathbf{z}}(t) \quad (39)$$

$$\dot{\mathbf{x}}(t) \equiv \begin{bmatrix} \dot{\mathbf{z}}(t) \\ \dot{\mathbf{w}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{w}(t) \\ \mathbf{M}^{-1}\mathbf{Q} \end{bmatrix} \equiv \begin{bmatrix} \mathbf{w}(t) \\ (\mathbf{R}^T\mathbf{M}\mathbf{R})^{-1} \left[ \mathbf{R}^T(\mathbf{Q} - \mathbf{M}\mathbf{R}\dot{\mathbf{z}}) \right] \end{bmatrix} \equiv \mathbf{f}(\mathbf{x}(t), t) \quad (40)$$

It is worth pointing out that all the elements of the state vector are independent by definition. Therefore, when the state vector contains positions and velocities, the Kalman corrections are applied to velocities and accelerations independently. This implies that the time–derivative relation between the position and the velocity as well as between the velocity and the acceleration is not thoroughly respected. This is not a problem since the filter tries to provide the best estimate of the velocity, which should overcome the effect of the derivation of the position estimation error. The same comment applies also to the estimate of the acceleration.

##### 4.2 THE UNSCENTED KALMAN FILTERS WITH MB MODELS

First, the approach to employ the projection matrix–R method with the UKF and the SSUKF will be described. In order to match the *nonlinear difference equations* of the filter (Equation 7), the equations of motion (Equation (33)) have to be integrated using either the TR (Equation (36)) or the RK2 method (Equation (37)). Subsequently, the choice of the variables to include in the state vector is a crucial point that will condition the implementation of the remaining equations of the filter. The first important comment is that, unlike in the EKF, the variable duplication is not imposed in this filter. As a consequence, there are no constraints for the selection of the states apart from the independence of the equations. As the projection

matrix–R method is a formulation in independent coordinates, the state vector could be made for instance of some of the following vectors:  $\mathbf{z}_k$ ,  $\dot{\mathbf{z}}_k$  or  $\ddot{\mathbf{z}}_k$ . Nevertheless, another important issue when choosing the state vector is to attempt to maintain the dimension of the state vector as small as possible in order to calculate the minimum number of sigma–points, thus reducing the overall computational cost. For this filter, the state vector has been chosen equal to the independent acceleration vector  $\ddot{\mathbf{z}}$ . In this way, the time–derivative relation between positions, velocities and accelerations is preserved. The equations of motion (Equation (7)) and the *a posteriori* state estimates (Equation (13)) corresponding to the new state vector are presented in Equations (41) and (42). A direct physical significance can be given to the Kalman corrections, namely  $\bar{\mathbf{K}}_{k+1}(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-)$  in Equation (42). Indeed as  $\mathbf{x}_k = \ddot{\mathbf{z}}_k$ , the Kalman corrections are also accelerations, meaning that these corrections can be understood as forces introduced to guide the system towards its real motion (partially given by the sensors’ information) after the information from a sensor is available.

$$\ddot{\mathbf{z}}_{k+1} = \phi_k(\ddot{\mathbf{z}}_k) + \mathbf{w}_k \quad (41)$$

$$\hat{\ddot{\mathbf{z}}}_{k+1} = \hat{\ddot{\mathbf{z}}}_{k+1}^+ = \hat{\ddot{\mathbf{z}}}_{k+1}^- + \bar{\mathbf{K}}_{k+1}(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-) \quad (42)$$

Once the state vector has been defined, the filter’s equations can be applied. First, the set of sigma–point is calculated using Equation (20) for the UKF or Equation (28) for the SSUKF. Then, each sigma–point is propagated through the nonlinear discrete–time system function (Equation (21)), in other words, a function evaluation of the MB model is performed for each sigma–point. The posterior independent acceleration vector  $\hat{\ddot{\mathbf{z}}}_{k+1}$  is obtained using Equation (22). Then, it is used to assess the posterior independent position ( $\mathbf{z}_{k+1}$ ) and velocity ( $\dot{\mathbf{z}}_{k+1}$ ) vectors by means of the TR. If no information from the sensors is available, the covariance matrix is obtained through Equation (10) and the time–update Equations (11) and (12) are applied. If any sensor data are available, the measurement–update Equations (24), (25), (26), (27) and (42) are applied.

After having placed a proposal to employ the projection matrix–R with UKFs, it is also of great interest to describe how to use the penalty approach, because of its lower computational cost. Only a few details differ from the previous explanation. First, in this case the dependent vector of coordinates has been augmented with the independent coordinate vector, as follows:  $\mathbf{q} = \begin{bmatrix} \mathbf{q}_d \\ \mathbf{q}_i \end{bmatrix}$ . However, as before, the state vector is taken as equal to the independent acceleration vector. Therefore, an extra step is necessary to update the remaining elements of  $\ddot{\mathbf{q}}$  (i.e.  $\ddot{\mathbf{q}}_d$ ) when  $\ddot{\mathbf{z}} \equiv \ddot{\mathbf{q}}_i$  is updated by the filter, as shown in Equation (43). The updates of  $\ddot{\mathbf{z}}$  take place in Equation (22) and Equation (42).

$$\ddot{\mathbf{q}}_k = \mathbf{R}_k \dot{\mathbf{z}}_k + \dot{\mathbf{R}}_k \dot{\mathbf{z}}_k \quad (43)$$

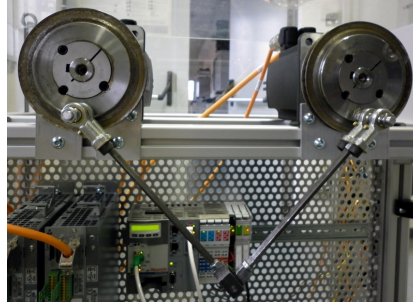
## 5 IMPLEMENTATION WITH A SIMPLE MECHANISM

A 5–bar linkage has been employed to exemplify the implementation and the performances of all the aforementioned nonlinear observers. The mechanism parameters have been obtained from the experimental 5–bar linkage shown in Figure (3) and the sensor characteristics from off-the-shelf sensors to reproduce a realistic simulation. As a first step, the motion of the real mechanism has been simulated but the magnitudes that correspond to sensor data are passed to the observers with their respective noises and sample rates. After that, some known errors (lengths, mass, inertia measuring errors for example) have to be considered for the dynamics system model of the filters, in order to allow the simulated real mechanism and the filter’s model to have similar but different behaviors. Employing a simulated real mechanism allows a comparison of all the states, including those that cannot be measured experimentally. As a consequence, all the variables of the filter’s model (i.e. the virtual sensors and the variables estimated through the state observer) can be compared to their exact magnitudes. In this way, comparisons of the filters’ performances are more genuine and comprehensive.

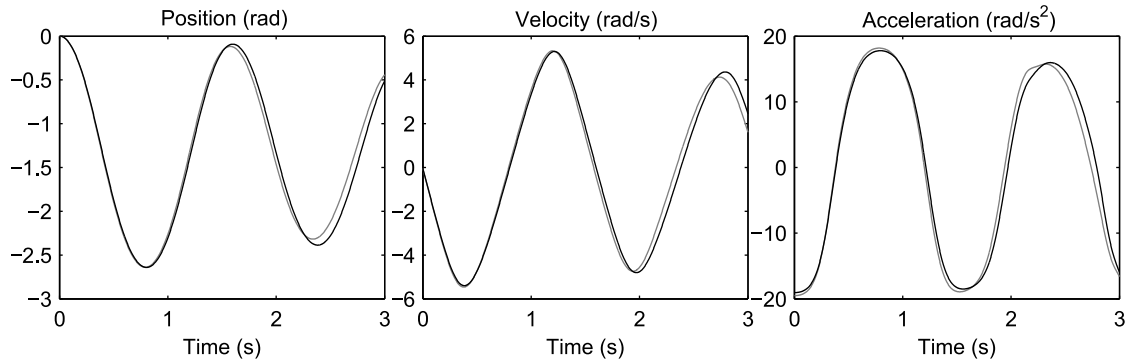
The motion selected to exemplify clearly the behavior of the filters should be as simple as possible. As a consequence, both pulleys start from an initial angle and then turn freely (i.e. without any inputs nor external forces) during the rest of the simulation. Figure (4) shows this motion for both the real mechanism and the dynamics system model used in the filters. The behavior discrepancies are clearly shown.

The corresponding figures for the different filters are not relevant since all the observers follow accurately the real mechanism. Instead, as the motion of the real mechanism is known exactly (as it is simulated), it





**Figure 3.** Photo of the 5-bar linkage



**Figure 4.** Angular positions, velocities and accelerations of the left pulley for the real mechanism (grey lines) and the dynamics system model of the filters (black lines)

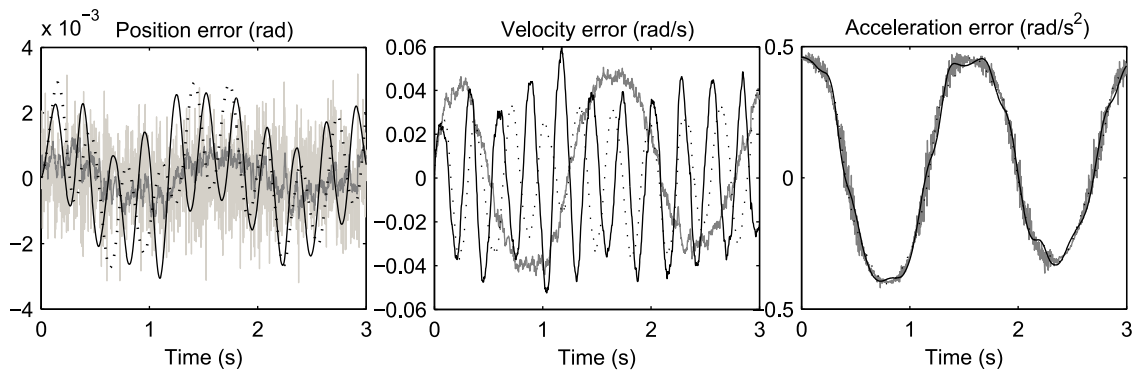
results more pertinent to represent the errors with respect to the real mechanism as shown in Figure (5) for the following sets: "EKF – matrix–R – TR", the "UKF – matrix–R – TR" and the "SSUKF – matrix–R – TR". The results for the remaining filters are not presented because of their similarity. The sensors for all the filters are the two pulley encoders. The CPU time for all the mentioned filters with respect to the filter with the lower computational cost are presented in Table (1). The root mean squared errors, where the error is the difference between the predicted measurements and the actual measurements, are also presented in Table (1). It is worth pointing out that the most accurate filter is the EKF, with an RMSE slightly superior to the noise standard deviation ( $10^{-3}$ ). However the rest of filters present also very similar results while correcting only the independent acceleration vector. It can be said that the SPKFs would be much more precise than the EKF if the velocities are estimated, at the expense of an additional computational cost.

**Table 1.** Comparison of CPU times and RMSE for several different filters

Filter	MB formu.	Integrator	CPU time	RMSE ( $10^{-3}$ )
SSUKF	penalty formu.	RK2	100%	2.13
EKF	matrix–R	TR	104%	1.36
SSUKF	matrix–R	RK2	119%	2.16
SSUKF	matrix–R	TR	183%	2.18
UKF	matrix–R	TR	216%	1.98

## 6 CONCLUSIONS

This work presents the theoretical development and practical implementation of nonlinear state observers using MB models. Several types of Kalman filters have been considered: the EKF in its continuous form



**Figure 5.** Sensor noise (light gray) and errors for the EKF (gray), the UKF (black) and the SSUKF (dashed lines)

and some SPKFs (UKF, SSUKF). Although these SPKFs outperform the accuracy of the EKFs by using better approximations of system nonlinearities, the sigma-points have brought about an additional computational cost. However, as implicit integrators have proved to improve observer accuracy insignificantly when used in closed-loop state estimation, explicit integrators help to reduce computational costs. Indeed, the difference between the predicted measurement and the actual measurement (i.e. the innovation) is adopted to correct the estimates and allows the prevention of numerical energy loss due to the integration. In the same way, the filters that allow the use of the penalty formulation have a lower computational load because of its simpler calculations than the ones of the matrix-R method. In light of the results of this study, the choice of the most suitable set of "filter, MB formulation and integrator" depends on the application requirements and is a trade-off between estimation accuracy and computational efficiency. Future works will be devoted to deeper investigations on wider classes of MB formulations and filters, and to the experimental validation of the proposed observers.

## ACKNOWLEDGMENTS

The authors would like to thank the Spanish Ministry of Science and Innovation and ERDF funds through the grant TRA2009-09314 for its support in this research.

## REFERENCES

- [1] Caracciolo, R.; Richiedei, D.; Trevisani, A.: Experimental validation of a model-based robust controller for multi-body mechanisms with flexible links. *Multibody System Dynamics*, Vol. 20, No. 2, pp. 129–145, 2008.
- [2] Cuadrado, J.; Dopico, D.; Barreiro, A.; Delgado, E.: Real-time state observers based on multibody models and the extended Kalman filter. *Journal of Mechanical Science and Technology*, Vol. 23, No. 4, pp. 894–900, 2009.
- [3] García de Jalón, J. and Bayo, E.: *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time challenge*. New-York: Springer-Verlag, 1994.
- [4] Julier, S.J.; Uhlmann, J.K.: Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, Vol. 92, No. 3, pp. 401–422, 2004.
- [5] Julier, S.J.: The spherical simplex unscented transformation. *Proceedings of the American Control Conference*, Vol. 3, pp. 2430–2434, 2003.