The 3rd Joint International Conference on Multibody System Dynamics
The 7th Asian Conference on Multibody Dynamics
June 30-July 3, 2014, BEXCO, Busan, Korea

Towards Benchmarking of State Estimators for Multibody Dynamics

José-Luis Torres#, José-Luis Blanco#, Emilio Sanjurjo*, Miguel-Ángel Naya*, Antonio Giménez#

# Escuela Politécnica Superior
University of Almería
La Cañada de San Urbano, 04120 Almería, Spain
[jltmoreno,jlblanco,agimfer]@ual.es

* Laboratorio de Ingeniería Mecánica
University of La Coruña
Mendizábal s/n, 15403 Ferrol, Spain
[emilio.sanjurjo,minaya]@udc.es

ABSTRACT

Multibody simulations are already used in many industries to speed up the development of new products. However, improvements in multibody formulations and the continuous increase of computational power open new fields of applications for the multibody simulations, such as using them as a basis for developing state observers. This work introduces novel state observers developed by combining a multibody model with some probabilistic estimators from the family of Kalman filters. Together with other multibody-based state observers already proposed in the literature, have been benchmarked by applying them to a planar four-bar linkage with different levels of sensor noise and modeling errors. Then, the accuracy of the estimations as well as the computational costs are examined. It will be shown that the Discrete Iterated Extended Kalman filter with perfect measurement, a method applied to multibody state estimation for the first time in this work, provides one of the best trade-offs between efficiency and accuracy.

1 INTRODUCTION

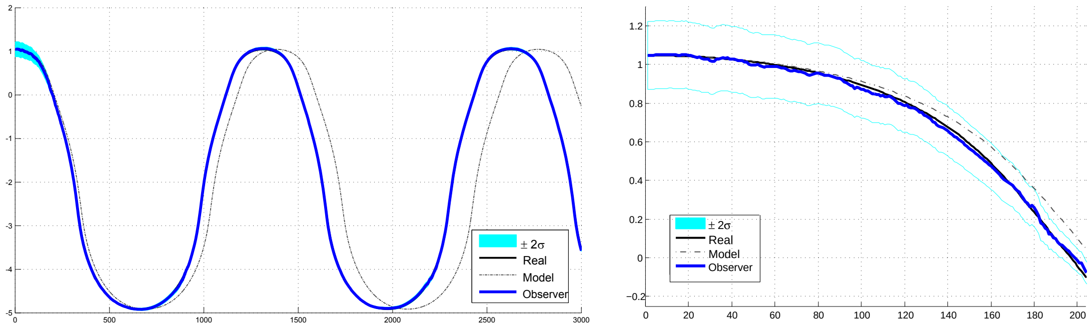
Being able of performing dynamic analyses of complex Multibody Systems (MBS) before manufacturing is key in achieving more efficient and competitive industries in sectors such as automotive or aeronautics. Doing so allows the manufacturer to study the expected dynamic behavior of products before building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

However, simulating the dynamics of MBS has other farther-reaching applications, such as devising state observers. A *state observer* is a recursive Bayesian estimator [9] aimed at providing a real-time best estimation of some parameters of a machine or vehicle, by means of integrating a more or less accurate multibody model with the history of sensor measurements. The parameters of interest typically describe the dynamic state (position, velocities and accelerations) of an interesting part of the model, as it is the case in the present study. By employing accurate MBS models, we are able to indirectly *infer* the state of non-accessible parts of a mechanism where installing a sensor would not be practical; typical sample results are illustrated in Figure 1. Examples of this problem are estimating tire-ground interaction forces in vehicle dynamics, or the parameters (e.g. length, mass) of a poorly-modeled part of the vehicle. In fact, previous works have already explored the possibility of improving the position and orientation estimate for a real vehicle [3]. These examples demonstrate the potential of state observers and, indeed, require much future research.

In this paper we chose to benchmark the behavior of state observers when used in one particular task with potential commercial applications: replacing existing complex and bulky devices (e.g. encoders) by smaller and lightweight sensors (e.g. MEMS gyroscopes). It is worth highlighting that, in spite of absolute orientations of one part not being directly observable from gyroscopic readings (turning rates), the estimator *fuses* the sequence of sensory data with *a priori* knowledge (the MBS model), thus successfully recovering absolute orientations. Ideally, we should be able to run a state observer in real-time, simultaneously to the plant or vehicle operation, such that the estimator takes live sensory data and outputs the sought state which can then be used by control subsystems.

2 METHODOLOGY

In this paper we address a central question in MBS state observers, namely, deciding which estimation algorithm is the best choice among the large number of possibilities found in the probabilistic estimation



(a)

(b)

Figure 1. (a) Example results for state observer tracking the mechanism motion (θ). Here it is shown the probabilistic confidence interval (shaded area), the ground truth (thick black), the estimator most likely outcome (thick blue) and the dynamic simulation without sensor-based correction (thin dashed). Vertical axis stands for the d.o.f. of the mechanism (radians) while horizontal axis represents time steps. (b) Scaled image of (a).

literature [1, 9]. Therefore, we built a benchmark consisting of a simple mechanism, a planar four-bar linkage, shown in Figure 2, and performed a statistical analysis of the accuracy and relative efficiency of a number of estimators. All results rely on simulations in order to have a ground truth suitable for a fair comparison among the methods. We selected the orientation of the left-most link, θ in Figure 2, to be the degree of freedom (d.o.f.) to be estimated by the estimators from the readings of one (noisy) gyroscope which may be placed in any of the three links. Gravity is the only actuating force.

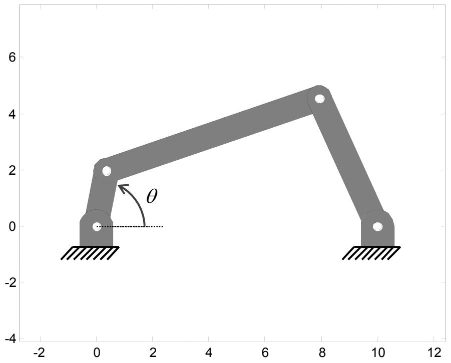


Figure 2. The four-bar linkage employed as a testbed in this work.

In order to test the different algorithms against difficulties usually found in practice, we intentionally introduced (i) different noise levels to the sensor outcome, and (ii) errors in the magnitude of actuating forces. Sensor noise or error is inherent to physical measurements and will always be present in real-world conditions. Thus, it becomes crucial to determine whether some methods tolerate more noise than others or not. Regarding the introduction of force errors, our intention is to characterize the usage of imperfect MBS models, something also always found in practice since no model will ever fit *exactly* real physical system.

In this work we focus exclusively on *filtering* probabilistic estimators, that is, methods devised to provide an immediate estimate of the mechanism state with the minimum delay since each sensor reading. Other possibilities, like fixed-lag smoothers or batch estimators [7] may be addressed in future research. We have evaluated the continuous and discrete-time versions of the Extended Kalman Filter, denoted CEKF and DEKF, respectively. The Unscented Kalman Filter (UKF), better suited to cope with strong nonlinearities, has been also included in the benchmark. In these three cases, which have been separately explored in previous related works [3, 8], the filter state only comprises independent coordinates. Additionally, we propose applying to MBS another existing algorithm (the Smoothly constrained Kalman Filter, or SCKF)

n	Number of dependent coordinates
m	Number of constraints
$g = n - m$	Number of degrees of freedom
$\mathbf{z}$	Vector of independent coordinates
$\mathbf{q} = \mathbf{q}(\mathbf{z})$	Vector of dependent coordinates
$\mathbf{\Phi}(\mathbf{q}) = 0$	Constraint equations
$\mathbf{\Phi_q}$, $\mathbf{\Phi_x}$	Jacobian of $\mathbf{\Phi}$ with respect to $\hat{\mathbf{q}}$, $\hat{\mathbf{x}}$
$\mathbf{M}$	Mass matrix
$\mathbf{Q}$	Vector of generalized forces
$\mathbf{x}, \hat{\mathbf{x}}$	Real value and estimation of the filter state vector
$\hat{\mathbf{x}}_k^-, \hat{\mathbf{x}}_k^+$	Estimation mean at time step k, before and after the update stage
$\mathbf{P}_k^-, \mathbf{P}_k^+$	Estimation covariance at time step k, before and after the update stage
$\mathbf{f}(\cdot), \mathbf{f_x}, \mathbf{f_q}$	Transition model and its Jacobians w.r.t. $\hat{\mathbf{x}}$ and $\hat{\mathbf{q}}$
$\mathbf{h}(\cdot), \mathbf{h_x}, \mathbf{h_q}$	Observation (sensor) model and its Jacobians w.r.t. $\hat{\mathbf{x}}$ and $\hat{\mathbf{q}}$
$\mathbf{o}_k$	Sensor measurements at time step k
$\mathbf{\Sigma}^P$	Covariance matrix of system transition ("plant") noise
$\mathbf{\Sigma}^S$	Covariance matrix of sensors noise
$\mathbf{K}$	Kalman gain matrix
$\mathbf{I}_N$	The $N \times N$ unit matrix

Table 1. Notation summary.

and introduce new ones, i.e. two variations of an iterated DEKF with dependent coordinates.

3 PROBABILISTIC STATE ESTIMATORS FOR MBS DYNAMICS

Next we introduce the different filtering algorithms that are benchmarked in this work. In spite of all of them being considered derived methods of the linear Kalman Filter (KF) [6], further method-specific details are in order due to the variety of subtleties in the different approaches with regard to how they handle nonlinearities, assumptions about the existence of constraints or not in the state space, etc. The reader can refer to Table 1 as a summary of the notation employed in the following.

A first fundamental classification of state estimators consists of distinguishing between those whose *state vector* comprises the independent coordinates only and those that include a complete set of redundant, dependent coordinates. The former family of estimators is the largest and comprises the oldest methods, since most probabilistic estimators proposed in the estimation and control theory literature assume a state vector in an n-dimensional Euclidean space ($\hat{\mathbf{q}} \in \mathbb{R}^n$), free of constraints. The underlying assumption is that each coordinate is, in principle, totally independent from the rest. It is only thanks to the cross-covariance terms of the estimate uncertainty and to the model and observation Jacobians that we can indirectly estimate all variables by means of an arbitrary observation function that depends on a subset of them. On the other hand, we find *dependent coordinate methods*, which have been proposed in the literature to handle the case of intra-state vector dependencies, exactly the situation found in MBS dynamics, i.e. via $\mathbf{\Phi}(\hat{\mathbf{q}}) = \mathbf{0}$. We firstly explore the solutions aimed at independent coordinate formulations, then address the dependent coordinate methods in section 3.2. In all cases we will directly present the equations of each filter as they should apply to the problem of MBS state estimation.

3.1 Independent coordinates filters

3.1.1 Continuous extended Kalman filter (CEKF)

This formulation was already described in some previous works [2] but will be reproduced here for the convenience of readers. The main idea under this formulation is to adapt the multibody equations in order to fit the Kalman filter structure. In its most basic form, the dynamics of a multibody system is described by the constrained Lagrangian equations:

$$\left\{ \begin{array}{rcl} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi_q}^{\top}\boldsymbol{\lambda} & = & \mathbf{Q} \\ \mathbf{\Phi} & = & \mathbf{0} \end{array} \right. \tag{1}$$

As the multibody equations are expressed in the form of continuous-time differential equations, they continuous Kalman filter equations must be selected as well. The multibody formalism employed is the R-matrix formulation [5]. The main idea behind this formulation is to obtain an ODE with dimension g equal to the number of degrees of freedom, starting with the identity $\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{z}}$, which relates dependent and independent velocities. Accelerations can be then expressed as follows:

$$\ddot{\mathbf{q}} = \mathbf{R}\ddot{\mathbf{z}} + \dot{\mathbf{R}}\dot{\mathbf{z}} \tag{2}$$

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

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

If now the filter state is defined as the vector $\mathbf{x}^{\top} = \left\{\mathbf{z}^{\top}, \dot{\mathbf{z}}^{\top}\right\}$, it turns out that:

$$\left\{ \begin{array}{c} \dot{\mathbf{z}} \\ \ddot{\mathbf{z}} \end{array} \right\} = \left\{ \begin{array}{c} \dot{\mathbf{z}} \\ \bar{\mathbf{M}}^{-1}\bar{\mathbf{Q}} \end{array} \right\} \quad \Rightarrow \quad \dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}\right) \tag{4}$$

These equations perfectly fit the continuous extended Kalman filter equation, so they can be straightforwardly applied. In particular, the state-space transition matrix is obtained as the linearization:

$$\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \left[\begin{array}{cc} \mathbf{0} & \mathbf{I} \\ \dfrac{\partial\left(\bar{\mathbf{M}}^{-1}\bar{\mathbf{Q}}\right)}{\partial \mathbf{z}} & \dfrac{\partial\left(\bar{\mathbf{M}}^{-1}\bar{\mathbf{Q}}\right)}{\partial \dot{\mathbf{z}}} \end{array} \right] \tag{5}$$

which can be approximated by:

$$\mathbf{A} \simeq \left[\begin{array}{cc} \mathbf{0} & \mathbf{I} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{array} \right] \tag{6a}$$

$$\mathbf{A}_{21} = -\bar{\mathbf{M}}^{-1}\mathbf{R}^{\top}\left(\bar{\mathbf{K}}\mathbf{R} + 2\mathbf{R}_{\hat{\mathbf{q}}}\mathbf{R}\ddot{\mathbf{z}}\right) \tag{6b}$$

$$\mathbf{A}_{22} = -\bar{\mathbf{M}}^{-1}\mathbf{R}^{\top}(\bar{\mathbf{C}}\mathbf{R} + \mathbf{M}\dot{\mathbf{R}}) \tag{6c}$$

where $\bar{\mathbf{K}}$ and $\bar{\mathbf{C}}$ are the stiffness and damping matrices, respectively. In this case the size of the problem is $2g$. Next, we introduce the CEKF *correction stage* [9], which fuses the sensor information into the filter, leading to:

$$\dot{\mathbf{z}} - \hat{\dot{\mathbf{z}}} + \bar{\mathbf{K}}^{\mathbf{z}}(\mathbf{y} - \mathbf{o}) = \mathbf{0} \tag{7a}$$

$$\bar{\mathbf{M}}\ddot{\mathbf{z}} - \bar{\mathbf{Q}} + \bar{\mathbf{M}}\bar{\mathbf{K}}^{\dot{\mathbf{z}}}(\mathbf{y} - \mathbf{o}) = \mathbf{0} \tag{7b}$$

In order to numerically integrate the result of the filter, the implicit single-step trapezoidal rule has been selected as integrator:

$$\hat{\dot{\mathbf{z}}}_{n+1} = \frac{2}{\Delta t}\hat{\mathbf{z}}_{n+1} - \left(\frac{2}{\Delta t}\hat{\mathbf{z}}_n + \hat{\dot{\mathbf{z}}}_n\right) \tag{8a}$$

$$\hat{\ddot{\mathbf{z}}}_{n+1} = \frac{2}{\Delta t}\hat{\dot{\mathbf{z}}}_{n+1} - \left(\frac{2}{\Delta t}\hat{\dot{\mathbf{z}}}_n + \hat{\ddot{\mathbf{z}}}_n\right) \tag{8b}$$

Combining Eq. (7) and Eq. (8) leads to the following nonlinear system,

$$\left\{ \begin{array}{l} \mathbf{g}_1(\hat{\mathbf{x}}_{n+1}) = \mathbf{0} \\ \mathbf{g}_2(\hat{\mathbf{x}}_{n+1}) = \mathbf{0} \end{array} \right. \Rightarrow \mathbf{g}(\hat{\mathbf{x}}_{n+1}) = \mathbf{0} \tag{9}$$

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

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}} = \left[\begin{array}{cc} \dfrac{2}{\Delta t}\mathbf{I} & -\mathbf{I} \\ \mathbf{R}^{\top}\bar{\mathbf{K}}\mathbf{R} & \mathbf{R}^{\top}(\bar{\mathbf{C}}\mathbf{R} + \mathbf{M}\dot{\mathbf{R}}) + \dfrac{2}{\Delta t}\bar{\mathbf{M}} \end{array} \right] + \left[\begin{array}{cc} \bar{\mathbf{K}}^{\mathbf{z}}\mathbf{h_z} & \bar{\mathbf{K}}^{\mathbf{z}}\mathbf{h}_{\dot{\mathbf{z}}} \\ \bar{\mathbf{M}}\bar{\mathbf{K}}^{\dot{\mathbf{z}}}\mathbf{h_z} & \bar{\mathbf{M}}\bar{\mathbf{K}}^{\dot{\mathbf{z}}}\mathbf{h}_{\dot{\mathbf{z}}} \end{array} \right] \tag{10}$$

where $\mathbf{h_z}$ and $\mathbf{h}_{\dot{\mathbf{z}}}$ are the position and velocity parts of the sensor Jacobian matrix.

3.1.2 Discrete extended Kalman filter (DEKF)

This is the discrete-time version of CEKF described above. A key difference between CEKF and the rest of estimators described from now on, which work in discrete time steps, is that the filter formulation clearly consists of two separated stages: state transition (also called *prediction*) and *state update*. The former relies on the transition model of the system (dynamical equations) while the latter includes the information from sensors, or *observations* – this is in contrast to CEKF where both stages are seamlessly fused together.

Each stage comprises differentiated equations for updating the state vector and the covariance matrix. Starting with the prediction stage, the EKF equations in their most generic form are:

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^+) \tag{11a}$$

$$\mathbf{P}_k^- = \mathbf{f_x}_{k-1}\mathbf{P}_{k-1}^+\mathbf{f_x}_{k-1}^\top + \mathbf{\Sigma}_{k-1}^P \tag{11b}$$

where $\mathbf{f}(\cdot)$ stands for the transition model of the system. By considering now the state vector of a MBS estimator in independent coordinates, $\hat{\mathbf{x}}^\top = \left\{\hat{\mathbf{z}}^\top, \hat{\dot{\mathbf{z}}}^\top\right\}$, and assuming the usage of the Euler method for numerical integration with time step Δt, we can put the integrator in a form that fits that required by the EKF transition function $\mathbf{f}(\cdot)$:

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^+) \quad \longrightarrow \begin{bmatrix} \hat{\mathbf{z}}_k \\ \hat{\dot{\mathbf{z}}}_k \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{z}}_{k-1} + \Delta t \hat{\dot{\mathbf{z}}}_{k-1} \\ \hat{\dot{\mathbf{z}}}_{k-1} + \Delta t \ddot{\mathbf{z}}_{k-1} \end{bmatrix} \tag{12}$$

Here, the only unknown term is the acceleration vector $\ddot{\mathbf{z}}_{k-1}$ for the previous time step, which must be computed by solving the multibody equations of motions as in Eq. (3). Thus, it follows that the transition model Jacobian $\mathbf{f_x}$ has a fairly simple structure:

$$\mathbf{f_x} \equiv \frac{\partial \mathbf{f}}{\partial \hat{\mathbf{x}}} = \frac{\partial}{\partial\{\hat{\mathbf{z}}, \hat{\dot{\mathbf{z}}}\}} \begin{bmatrix} \hat{\mathbf{z}} + \Delta t \hat{\dot{\mathbf{z}}} \\ \hat{\dot{\mathbf{z}}} + \Delta t \ddot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_g & \Delta t \mathbf{I}_g \\ \mathbf{0}_{g \times g} & \mathbf{I}_g \end{bmatrix} \tag{13}$$

Regarding the $\mathbf{\Sigma}_{k-1}^P$ covariance matrix appearing in Eq. (11), it stands for the additional uncertainty of the new state $\hat{\mathbf{x}}_k$, physically attributable to unmodeled forces and errors in the parameterization of the mechanism (e.g. lengths of bars, inertia values, etc.). Assuming independent and identically distributed (iid) Gaussian noise for each independent coordinate, its structure becomes:

$$\mathbf{\Sigma}_{k-1}^P = \left[\begin{array}{c|c} \sigma_{\hat{\mathbf{z}}}^2 \mathbf{I}_g & \mathbf{0}_{g \times g} \\ \hline \mathbf{0}_{g \times g} & \sigma_{\hat{\dot{\mathbf{z}}}}^2 \mathbf{I}_g \end{array}\right] \tag{14}$$

with the parameters $\sigma_{\hat{\mathbf{z}}}$ and $\sigma_{\hat{\dot{\mathbf{z}}}}$ specifying the standard deviations of the assumed noise in position and velocities, respectively.

The second stage of the DEKF method, the *update*, incorporates the sensor readings to improve the estimate:

$$\tilde{\mathbf{y}}_k = \mathbf{o}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-) \tag{15a}$$

$$\mathbf{S}_k = \mathbf{h_x}_k \mathbf{P}_k^- \mathbf{h_x}_k^\top + \mathbf{\Sigma}_k^S \tag{15b}$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{h_x}_k^\top \mathbf{S}_k^{-1} \tag{15c}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \tilde{\mathbf{y}}_k \tag{15d}$$

$$\mathbf{P}_k^+ = (\mathbf{I}_g - \mathbf{K}_k \mathbf{h_x}_k)\mathbf{P}_k^- \tag{15e}$$

where $\mathbf{h}(\cdot)$ stands for the observation model of the system, such that $\tilde{\mathbf{y}}_k$ in Eq. (15a) is clearly the error or mismatch (often called *innovation*) between the expected sensor readings and their actual values ($\mathbf{o}_k$). The covariance matrix $\mathbf{S}_k$ in Eq. (15b), or *innovation covariance*, represents the uncertainty in the system state projected via the sensor function ($\mathbf{h_x}_k \mathbf{P}_k^- \mathbf{h_x}_k^\top$) plus an additional additive Gaussian noise originated in the sensor itself ($\mathbf{\Sigma}_k^S$). Small values of $\mathbf{S}_k$ mean that the observation introduces useful information to constrain the estimation of the system state. By evaluating the temporary term known as *Kalman gain* ($\mathbf{K}_k$) we can update the estimate mean and covariance, in Eq. (15d) and Eq. (15e), respectively. These values are then used as the input to the next iteration of this iterative filter in the next time step.

3.1.3 Discrete iterated EKF with virtual acceleration sensor (DIEKF_acc)

The Iterated version of the EKF is more suitable to handle nonlinearities in the system models and also should provide a more accurate estimation thanks to the reduction in the errors introduced by the first-order linear approximation of the observation function in the underlying least-squares optimization problem [9].

In this work we propose to exploit this possibility by defining a DIEKF estimator over the entire dynamical state of the system, including its accelerations, that is $\hat{\mathbf{x}}^\top = \left\{\hat{\mathbf{z}}^\top, \hat{\dot{\mathbf{z}}}^\top, \hat{\ddot{\mathbf{z}}}^\top\right\}$. Then, we will augment the observation function to include, not only the real sensors, but also a virtual observation of the independent accelerations as they should be according to the equations of motion. For each time step, the estimator will then iterate until it finds the state that best matches both the sensor readings and the expected accelerations.

The prediction stage is similar to that of DEKF in Eq. (11), where the transition function and its Jacobian are slightly modified to account for the acceleration in the state vector. Assuming a constant acceleration motion model and an Euler integrator, we have:

$$\begin{bmatrix} \hat{\mathbf{z}}_k \\ \hat{\dot{\mathbf{z}}}_k \\ \hat{\ddot{\mathbf{z}}}_k \end{bmatrix} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}) = \begin{bmatrix} \hat{\mathbf{z}}_{k-1} + \Delta t \hat{\dot{\mathbf{z}}}_{k-1} \\ \hat{\dot{\mathbf{z}}}_{k-1} + \Delta t \hat{\ddot{\mathbf{z}}}_{k-1} \\ \hat{\ddot{\mathbf{z}}}_{k-1} \end{bmatrix} \tag{16a}$$

$$\mathbf{f}_\mathbf{x} \equiv \frac{\partial \mathbf{f}}{\partial\{\hat{\mathbf{z}}, \hat{\dot{\mathbf{z}}}, \hat{\ddot{\mathbf{z}}}\}} = \begin{bmatrix} \mathbf{I}_g & \Delta t \mathbf{I}_g & \mathbf{0}_{g\times g} \\ \mathbf{0}_{g\times g} & \mathbf{I}_g & \Delta t \mathbf{I}_g \\ \mathbf{0}_{g\times g} & \mathbf{0}_{g\times g} & \mathbf{I}_g \end{bmatrix} \tag{16b}$$

The difference between DEKF and DIEKF comes with the update stage, which now turns into an iterative evaluation of the terms $\hat{\mathbf{x}}_i^+$, $\mathbf{K}_{k,i}$ and $\mathbf{P}_{k,i}$ with $i = 0, 1, ...$ the iteration count. In this case, the update equations read:

$$\mathbf{K}_{k,i} = \mathbf{P}_k^- \, {\mathbf{h}_\mathbf{x}}_{k,i}{}^\top \underbrace{\left({\mathbf{h}_\mathbf{x}}_{k,i} \mathbf{P}_k^- {\mathbf{h}_\mathbf{x}}_{k,i}{}^\top + \mathbf{\Sigma}_k^S \right)}_{\text{Innovation covariance}}{}^{-1} \tag{17a}$$

$$\hat{\mathbf{x}}_{k,i}^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_{k_i} \underbrace{\left(\mathbf{o}_k - \mathbf{h}(\hat{\mathbf{x}}_{k,i}^+) \right)}_{\text{Innovation}} \tag{17b}$$

$$\mathbf{P}_k^+ = (\mathbf{I}_g - \mathbf{K}_{k,i} {\mathbf{h}_\mathbf{x}}_{k,i}) \mathbf{P}_k^- \tag{17c}$$

Iterations end when the increment in the state vector is below some predetermined threshold. Notice that Eq. (17c), which is the most costly operation, can be evaluated only once after the end of the iterative process for the sake of efficiency.

3.1.4 Unscented Kalman filter (UKF)

The Unscented Kalman Filter (UKF) [11] is an evolution of the family of Kalman filters that is better suited to cope with strong nonlinearities in the transition and observation models. Comprising the same prediction and update stages than DEKF, the differentiating feature of UKF is the avoidance of the first order Taylor approximation in the propagation of Gaussian random variables through the transition and observation functions. Instead, a set of samples are deterministically-chosen from the Gaussian distributions, transformed via the corresponding function, then those samples in the transformed space converted back into a parametric distribution, i.e. they are used to compute the mean and covariance of the corresponding Gaussian. As shown in [11], this approach captures the correct posterior mean and covariance up to the third order of a Taylor series expansion, in contrast to the first order of DEKF and most other methods. In turn, its computational cost is in general higher than simpler methods.

For the present benchmark, the state vector of UKF comprises the independent coordinates and their velocities, that is, $\hat{\mathbf{x}}^\top = \left\{\hat{\mathbf{z}}^\top, \hat{\dot{\mathbf{z}}}^\top\right\}$. As mentioned above, each filter iteration comprises the same two steps than DEKF, so only the differences will be highlighted here. Denoting the dimensionality of the state space $|\hat{\mathbf{x}}|$ as L, a total of $2L + 1$ deterministic samples (or *sigma points*) χ_i with $i = 0, ..., 2L$ are generated from the mean $\hat{\mathbf{x}}^+_{k-1}$ and covariance $\mathbf{P}^+_{k-1}$, each with a different weight W_i. Then, the samples are transformed with a forward Euler transition function identical to that of previous filters, and the predicted mean $\hat{\mathbf{x}}^-_k$ and covariance $\mathbf{P}^-_k$ estimated from them. A similar process apply to the propagation of the uncertainty in observations, taking into account both the uncertainty in the system state and the sensor noise (refer to the two terms in the innovation covariance of DEKF above). The reader is referred to the original work [11] for the filter equations, not reproduced here for the sake of conciseness.

3.2 Dependent coordinates filters

One problem of the previous filters is that we have a direct estimate of the value and uncertainty of only the independent coordinates of the multibody system. This implies the need to solve position and velocity problems at every time step, which is time consuming. For this reason it may be desirable to have the multibody constraints inside the observer equations, although they do not fit Kalman filter equations in an obvious way. Next we explore two different alternatives to achieve it.

3.2.1 Smoothly constrained Kalman filter (SCKF)

This filter is based in the algorithm described in [4]. In this filter, the state $\hat{\mathbf{x}}$ is build with the whole multibody coordinates and velocities vectors $\hat{\mathbf{q}}$ and $\hat{\dot{\mathbf{q}}}$. The SCKF transition function is built assuming the forward Euler integrator:

$$\hat{\mathbf{x}}^-_k = \mathbf{f}(\hat{\mathbf{x}}^+_{k-1}) \quad \longrightarrow \begin{bmatrix} \hat{\mathbf{q}}_k \\ \hat{\dot{\mathbf{q}}}_k \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{q}}_{k-1} + \Delta t \hat{\dot{\mathbf{q}}}_{k-1} \\ \hat{\dot{\mathbf{q}}}_{k-1} + \Delta t \hat{\ddot{\mathbf{q}}}_{k-1} \end{bmatrix} \tag{18}$$

Thus, the transition model Jacobian is as follows:

$$\mathbf{f}_\mathbf{x} \equiv \frac{\partial \mathbf{f}}{\partial \{\hat{\mathbf{q}}, \hat{\dot{\mathbf{q}}}\}} = \begin{bmatrix} \mathbf{I}_n & \Delta t \mathbf{I}_n \\ \mathbf{0}_{n\times n} & \mathbf{I}_n \end{bmatrix} \tag{19}$$

and the covariance matrix is updated as:

$$\mathbf{P}^-_k = \mathbf{f}_\mathbf{x} \mathbf{P}^+_{k-1} \mathbf{f}_\mathbf{x}{}^\top + \mathbf{\Sigma}^P \tag{20}$$

After the time update, the measurement update is undertaken, but taking into account only the measurements coming from the sensors:

$$\mathbf{K}_k = \mathbf{P}^- \mathbf{h}^\top (\mathbf{h}\mathbf{P}^- \mathbf{h}^\top + \mathbf{\Sigma}^S)^{-1} \tag{21}$$

$$\hat{\mathbf{x}}^+_{k,0} = \hat{\mathbf{x}}^-_k + \mathbf{K}_k(\mathbf{o}_k - \mathbf{h}(\hat{\mathbf{x}}^-_k)) \tag{22}$$

Up to now, this is the algorithm of a conventional EKF, but at this moment the states are not expected to fit the constraints, so an iterative process is started. First the weakening matrix is calculated as follows:

$$\boldsymbol{\xi}_0 = \alpha \mathbf{\Phi}_\mathbf{x} \mathbf{P}^+_{k,0} \mathbf{\Phi}^\top_\mathbf{x} \tag{23}$$

where α is a tuning parameter. This weakening matrix contains virtual noise to be added to the multibody constraints in order to ease the convergence of the problem. The iterative update is as follows:

$$\mathbf{K} = \mathbf{P}^+_{k,i} \mathbf{\Phi}_\mathbf{x} + (\mathbf{\Phi}_\mathbf{x} \mathbf{P}^+_{k,i} \mathbf{\Phi}^\top_\mathbf{x} + \boldsymbol{\xi}_i)^{-1} \tag{24}$$

$$\hat{\mathbf{x}}^+_{k,i+1} = \hat{\mathbf{x}}^+_{k,i} - \mathbf{K} \begin{bmatrix} \mathbf{\Phi}(\hat{\mathbf{x}}^+_{k,i}) \\ \dot{\mathbf{\Phi}}(\hat{\mathbf{x}}^+_{k,i}) \end{bmatrix} \tag{25}$$

$$\mathbf{P}^+_{k,i+1} = (\mathbf{I} - \mathbf{K}\mathbf{\Phi}_\mathbf{x}) \mathbf{P}^+_{k,i} (\mathbf{I} - \mathbf{K}\mathbf{\Phi}_\mathbf{x})^\top + \mathbf{K}_{k,i} \boldsymbol{\xi}_i \mathbf{K}^\top_{k,i} \tag{26}$$

$$\boldsymbol{\xi}_{i+1} = \boldsymbol{\xi}_i e^{-\beta} \tag{27}$$

being β another tuning parameter. This iterative process is performed until the position constraints $\mathbf{\Phi}$, and velocity constraints $\dot{\mathbf{\Phi}}$ fit the desired tolerance.

3.2.2 Discrete iterated extended Kalman filter with perfect measurements (DIEKF_pm)

Another approach to expand a standard DIEKF to cope with constraints in its state space is employing so-called *perfect measurements* [10]. The key idea consists of augmenting the vector of observations $\mathbf{h}(\mathbf{x})$ to include *virtual observations* that reflect the fulfillment of the kinematics constraints in both position and velocities. The augmented observation function $\mathbf{h}'(\mathbf{x})$ is strongly nonlinear, hence the application in this case of an iterated estimator, capable of reducing the linearization errors to acceptable levels.

For the benchmark at hand, the state vector of this estimator comprises the multibody model coordinates and their derivatives, that is, $\hat{\mathbf{x}}^\top = \left\{\hat{\mathbf{q}}^\top, \hat{\dot{\mathbf{q}}}^\top\right\}$. We define the augmented observation model $\mathbf{h}'(\hat{\mathbf{x}})$ as the concatenation of the real sensors $\mathbf{h}(\mathbf{x})$ and the kinematic constraints in position and velocity, such as $\mathbf{h}'(\mathbf{x})^\top = [\mathbf{h}(\mathbf{x})^\top \; \mathbf{\Phi}(\mathbf{x})^\top \; \dot{\mathbf{\Phi}}(\mathbf{x})^\top]$. This affects the calculation of the innovation (or "residual"), which must compare the actual sensor readings and current constraint errors with their predictions. For all time steps k and iteration index i, the predicted values of the constrains are always zero, i.e.

$$\mathbf{y}_{k,i} = \begin{bmatrix} \mathbf{o}_k \\ \mathbf{0}_{2n\times 1} \end{bmatrix} - \mathbf{h}'(\hat{\mathbf{x}}_{k,i}) = \begin{bmatrix} \mathbf{o}_k - \mathbf{h}(\hat{\mathbf{x}}_{k,i}) \\ -\mathbf{\Phi}(\hat{\mathbf{x}}_{k,i}) \\ -\dot{\mathbf{\Phi}}(\hat{\mathbf{x}}_{k,i}) \end{bmatrix} \tag{28}$$

The adjective "perfect" that names this method comes from the assumption that there is no error source in the virtual observations. In practice, this implies employing an extended sensor covariance matrix $\mathbf{\Sigma}_k^{S'}$ with the structure:

$$\mathbf{\Sigma}_k^{S'} = \begin{bmatrix} \mathbf{\Sigma}_k^S & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{29}$$

Apart from these extensions, this iterated DEKF filter follows the same prediction and update equations than the version introduced in section 3.1.3.

4 EXPERIMENTAL RESULTS

All six methods presented in the previous section have been implemented in MATLAB for the specific case of the four-bar linkage shown in Figure 2 equipped with a noisy gyroscope in the middle bar as the unique sensor. As mentioned above, we have evaluated increasing levels of two kinds of errors: (i) inaccuracies in the model (in particular, an incorrect actuating gravity force) and (ii) sensor noise. For each method and noise level we run the state observer 100 times and compute average statistics in order to obtain statistically-significant results. With a fixed time step of 5ms, each run simulates 15 seconds of the real system (ground truth) and the state observer.

Regarding the computational time employed by each method, the results are summarized in Figure 3 in the form of ratio between computational cost and real time. This parameter is crucial when the aim of the

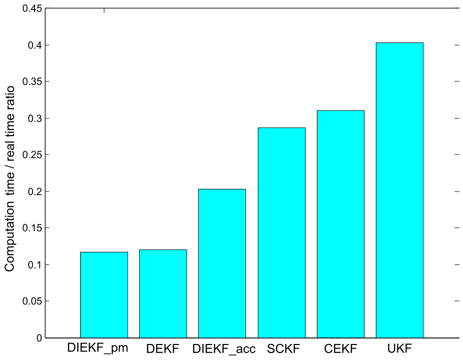
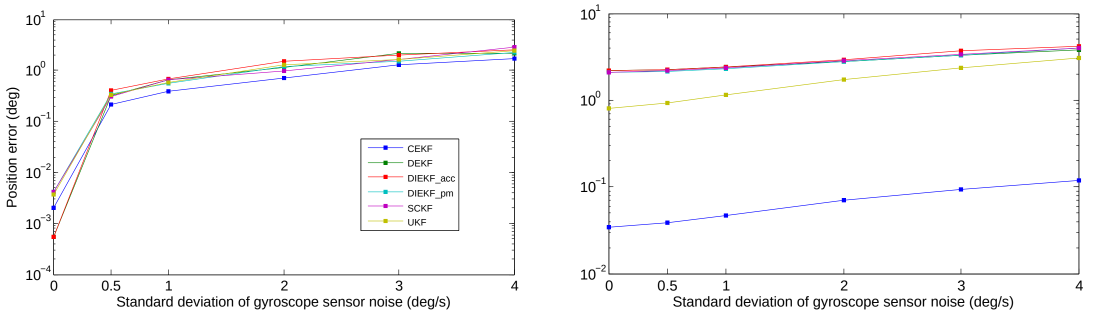


Figure 3. Execution time for each compared method, as the ratio between computation time and real time. Lower values mean faster, and those below 1.0 mean faster than real time.



(a)

(b)

Figure 4. (a) Root mean average error (RMSE). (b) Average Mahalanobis distance for each method.

observer is to provide feedback for a controller, where it is mandatory to achieve faster than real-time speed. All methods obtain values below the unity, reflecting that all of them are suitable for real-time operation in spite of the lack of optimizations and the usage of the MATLAB programming language. It can be seen how DIEKF with perfect measurements (DIEKF_pm) and DEKF are the most efficient methods. The efficiency of DIEKF_pm can be explained due to its quick convergence and the lack of separate position and velocity problems, present in the independent coordinate methods.

When exposed to errors in the MBS model, particularized as a mismatch in the assumed gravity force, no statistically significant differences were found among the behavior of all the tested estimators. Such a results may reflect the fact that all methods are robust enough to tolerate a wide range of errors in the system model as long as the sensor provides information enough to correct it.

The results for increasing levels of sensor (gyroscope) noise are summarized in Figure 4. Figure 4(a) illustrates the root mean average error (RMSE) d.o.f. of the mechanism, averaged over the 100 repetitions of each test case. Interestingly, most methods exhibit a very similar response. When dealing with probabilistic estimator, the RMSE is not the only accuracy metric that should be benchmarked: Figure 4(b) also shows the average Mahalanobis distance (MD) from the estimation to the ground truth for each method during the same experiments. Low MD values reflect either (i) a high accuracy in the estimated mean, or (ii) a filter that is too pessimistic in its estimation of the uncertainty. That is the reason MD must be contrasted with RMSE values. In this case, the low RMSE and MD of CEKF mean that it is really accurate in both, the estimated values and its uncertainty. For the rest of methods, a MD distance threshold of 3.0 is typically considered as the limit to consider a filter statistically consistent. In other words, a MD above that threshold indicates that the filter is overconfident in its estimation. CEKF and UKF are the unique methods that do not become overconfident over the entire range of tested sensor noise, with the rest of methods becoming inconsistent at some point.

5 CONCLUSIONS

The feasibility of different techniques for the estimation of states in MBS has been analyzed in this paper. A detailed explanation of each observer and its integration in the multibody model depending on the most suitable formulation in each case has been carried out. Then, the considered observers have been tested for predefined conditions of sensor noise, type of error and number of repetitions. This procedure allows setting the basis for benchmarking new combinations of observers and multibody formulations. This issue is presented as an interesting approach aimed to the selection of the most appropriate technique according to the requirements and the availability of sensors in real-world projects, as the case of automotive controllers. In particular, for the conditions under study in this paper it have been determined that the most computationally efficient methods are DEKF and DIEKF_pm, while the most accurate observer is CEKF. A good trade-off between efficiency and accuracy is provided by the DIEKF_pm method, one of the novel methods introduced in this work.

6 ACKNOWLEDGEMENTS

This work has been partially funded by the Spanish "Ministerio de Ciencia e Innovacion" under the contract "DAVARBOT" (DPI 2011-22513) and the grant program JDC-MICINN 2011, as well as by the Andalusian Regional Government grant programs FPDU 2008 and FPDU 2009, partially funded by the European Union through the European Regional Development Fund (ERDF). The research carried out by the third author is funded by doctoral fellowship BES-2013-063598 of Spanish Government.

REFERENCES

[1] Bishop, C.: Pattern Recognition and Machine Learning. New York: Springer, 2006.

[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] Cuadrado, J.; Dopico, D.; Perez, J. A.; Pastorino, R.: Automotive Observers Based on Multibody Models and the Extended Kalman Filter. Multibody System Dynamics, Vol. 27, No. 1, pp. 3–19, 2012.

[4] Geeter, Jan De and Van Brussel, Hendrik and Schutter, Joris De and Decréton, Marc: A Smoothly Constrained Kalman Filter. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 10, pp. 1171–1177, 1997.

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

[6] Kalman, R.: A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, Vol. 82, No. 1, pp. 35–45, 1960.

[7] Murphy, K. P.: Dynamic Bayesian Networks: Representation, Inference and Learning. Ph.D. thesis, University of California, 2002.

[8] Pastorino, R.; Richiedei, D.; Cuadrado, J.; Trevisani, A.: State Estimation Using Multibody Models and Non-Linear Kalman Filter. International Journal of Non-Linear Mechanics, Vol. 53, pp. 83–90, 2013.

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

[10] Simon, D.; Chia, T. L.: Kalman Filtering With State Equality Constraints. Aerospace and Electronic Systems, IEEE Transactions on, Vol. 38, No. 1, pp. 128–136, 2002.

[11] Wan, E.; Van Der Merwe, R.: The Unscented Kalman Filter for Nonlinear Estimation. In Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium, pp. 153–158, 2000.