

Deep learning of multibody minimal coordinates for state and input estimation with Kalman filtering

Andrea Angeli · Wim Desmet · Frank Naets

Received: date / Accepted: date

Abstract In general, multibody models are described with a set of redundant coordinates and additional constraints. Their dynamics is thus expressed through differential algebraic equations. As an alternative, the minimal coordinate formulation permits to describe a rigid system with the minimal number of variables leading to ordinary differential equations which can be employed in a coupled state/input estimation scheme. However, in some cases the explicit relation between the full system coordinates and the minimal coordinates may not be available or analytically obtainable, as for closed-loop mechanisms. In this work, a previously presented deep learning framework to find the non-linear mapping and reduce a generic multibody model from redundant to minimal coordinates is employed. The resulting equations are then exploited in an extended Kalman filter where the unknown inputs are considered as augmented states and jointly estimated. The necessary derivatives are given and it is shown that acceleration measurements are sufficient for the estimation. The method is experimentally validated on a slider-crank mechanism.

Keywords Kalman filter, input estimation, multibody dynamics, minimal coordinates, deep learning, physics-informed neural networks

1 Introduction

Nowadays, operational data are more and more important to guide the control algorithms in taking decisions: following the rise of ‘artificial intelligence’,

Andrea Angeli, Wim Desmet, Frank Naets
Department of mechanical engineering, KU Leuven
DMMS Lab, Flanders Make
Address: Celestijnenlaan 300, 3001 Leuven, Belgium
E-mail: andrea.angeli@kuleuven.be, wim.desmet@kuleuven.be, frank.naets@kuleuven.be

Nomenclature

\mathbb{Z}	integer numbers	\square^{-1}	matrix inverse
\mathbb{R}	real numbers	\otimes	Kronecker product
$\mathbf{a} \in \mathbb{R}^{n_1}$	column vector	$\underline{\mathbf{A}} \mathbf{A} = \underline{\mathbf{A}}_{\alpha,\beta,\gamma} \mathbf{A}_{\beta,\gamma}$	tensor-matrix product (Einstein sum convention)
$a \in \mathbb{R}$	scalar	$\dot{\square} = \frac{d\square}{dt}, \ddot{\square} = \frac{d^2\square}{dt^2}$	time derivatives
$\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$	matrix	$\frac{\partial \mathbf{a}_1}{\partial \mathbf{a}_2} \in \mathbb{R}^{n_{a_1} \times n_{a_2}}$	partial derivatives
$\underline{\mathbf{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$	3-d tensor	\square^-	a priori prediction
$\mathbf{I}_a \in \mathbb{Z}^{n_a \times n_a}$	identity matrix	\square^+	a posteriori prediction
$\mathbf{0}_{a_1} \in \mathbb{Z}^{n_{a_1}}$	zero vector	$\square_\tau = \square(t = \tau)$	τ -th time sample
$\mathbf{0}_{a_1, a_2} \in \mathbb{Z}^{n_{a_1} \times n_{a_2}}$	zero matrix		
t	time		
Δt	time step		
\square^T	transpose operator		

the smart products and production lines are expected to react to the environmental conditions. In order to do so, an evaluation of the relevant variables is required.

However, while some quantities such as temperature or acceleration are relatively easy to measure, others are more difficult to obtain in an inexpensive, non-invasive way. A typical example is the force/torque estimation in mechanical systems for which sensors exist, but they may be difficult or even impossible to mount in practice due to geometrical or economic reasons. These quantities might be of paramount importance in design, prediction or maintenance decisions. In this case, the engineering community frequently resorts to the state estimation discipline, combining models and available measurements.

If the physical system is a complex mechanism composed by several parts, its digital counterpart is typically a multibody model. Several coordinate selections have been proposed in literature and the most common ones, such as the ‘Cartesian Coordinates (CC)’ [1] or the ‘Natural Coordinates (NC)’ [2] for rigid systems, involve a redundant number of variables and additional constraints, leading to a formulation in Differential Algebraic Equations (DAEs) to express the dynamics. On the other hand, the ‘Minimal Coordinates (MC)’ [3] make use of a minimal amount of variables, allowing to formulate the dynamics through Ordinary Differential Equations (ODEs) and express the system in its state-space representation. However, for a generic model, the MC may not be available if the explicit non-linear mapping with the full system coordinates is unknown or not analytically obtainable as, for instance, for spatial closed-loop mechanisms.

In the case the model has an ODE structure, it can be effortlessly coupled with estimation algorithms since they are typically developed for such formulation. For example, a Kalman Filter (KF) [4,5] can be employed to combine model and measurements information to update the states which describe the system. Several KF variations have been proposed in literature for specific aims, for instance: the common discrete KF grants a straightforward imple-

mentation in digital computers (in contrast to the continuous time KF [6]), the extended KF permits to deal with non-linear equations thanks to a local linearisation and the augmented KF allows the combined estimation of states and unknown inputs added as augmented state.

Due to the typical redundant coordinate formulation and consequent DAEs structure of common multibody models, the application of Kalman filtering methods is usually not straightforward. Different approaches to convert such DAEs into ODEs to employ multibody models in a Kalman filtering scheme and perform state-input estimation have been proposed in literature [7]. In fact, over the past years, several authors have explored the use of dynamic multibody models for state-estimation purposes. On the one hand general multibody formulations have been explored in this context [8,9,10,11]. For example in [10], the constraints in the DAEs of the dynamics formulation are eliminated at each iteration allowing the reduction to ODEs and the coupling with an estimator. The price to pay is a non-negligible additional computational load during the simulation. In [11] on the other hand, the constraints are approximately enforced through penalty terms. While this solution permits to convert the DAEs to ODEs, care has to be taken in the choice of the penalty factors in order to approximate the constraints with sufficient accuracy. These methods have a broad applicability, but carry a relatively high computational cost and specific modifications to the multibody formulation are generally required to be able to handle the constraint equations. More dedicated approaches have also been explored where nonlinear reduced order modelling methods have been deployed [12,13] such that they can be more readily integrated in an estimation framework. In [12], the DAEs are reduced to ODEs in an offline phase for various configurations and during the on-line phase an interpolation is performed. This method has the benefit that it does not impact the simulation (and estimation) time, however, it requires an ad-hoc solver for the reduced order model setup and the interpolation may become non-trivial for spatial mechanisms with more rigid DOFs.

In this work, a model order reduction approach based on deep learning presented in [14,15,16] is exploited to obtain the minimal coordinate mapping and reduce a multibody model from redundant to minimal coordinates. The non-linear mapping from redundant to minimal coordinates is approximated with a neural network. This permits to reduce the general multibody model DAEs to ODEs that can be implemented in an estimation scheme. Here, in particular, the resulting minimal coordinate model will be employed in an Augmented Discrete Extended Kalman Filter (ADE-KF) for the combined estimation of system states and inputs. In this framework, the acceleration measurements are sufficient to guarantee the observability of the system as showed theoretically by the given derivatives and practically by the experimental results. This can be a strong benefit for industrial applications where accelerometers are typically available or easily applied, in contrast to more expensive or invasive sensors.

The paper is structured as follows: the remainder of this section lists the main contributions; section 2 recalls the Kalman filter formulation; section

3 refers to the multibody formulation and the model order reduction framework used in the procedure; section 4 reports the proposed methodology which combines the reduced-order model in a Kalman filter and provides the formulation for a generic case; section 5 shows a validation case on a slider-crank mechanism; section 6 contains some concluding remarks.

The main contributions are:

- deep learning is used to approximate the unknown explicit mapping between full and minimal multibody coordinates, obtaining an ODE model;
- the obtained model is embedded in an augmented discrete extended Kalman filter to jointly estimate states and unknown inputs and the required derivatives are given;
- the methodology is experimentally validated on a slider-crank mechanism, demonstrating that acceleration measures are sufficient to guarantee the observability of the system without additional position-level sensors.

2 Kalman filtering

In this section, the Augmented Discrete Extended Kalman Filter (ADE-KF) is briefly reviewed, as it will be exploited in the proposed estimation scheme.

2.1 (Discrete) Kalman filter

In general, the system to be predicted can be represented as a dynamic process, described by the states $\mathbf{x} \in \mathbb{R}^{n_x}$ and subjected to the inputs $\mathbf{u} \in \mathbb{R}^{n_u}$, which can then be related to the (measurable) outputs $\mathbf{y} \in \mathbb{R}^{n_y}$:

$$\dot{\mathbf{x}} = \mathbf{f}^*(\mathbf{x}, \mathbf{u}) + \mathbf{r}_x \quad (1)$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}) + \mathbf{r}_y \quad (2)$$

where $\mathbf{f}^* \in \mathbb{R}^{n_x}$, $\mathbf{h} \in \mathbb{R}^{n_y}$ respectively represent the system and measurement equations, each subjected to noise $\mathbf{r}_x \in \mathbb{R}^{n_x}$ and $\mathbf{r}_y \in \mathbb{R}^{n_y}$, in corresponding order.

The scope of the state-estimation discipline is to predict the process states \mathbf{x} . In practice, the (continuous) system equation \mathbf{f}^* is usually implemented in a digital computer, thus it is converted into its discrete form \mathbf{f} .

2.2 Augmented Kalman filter

In certain applications, some unknown excitations may act on the system. In this case \mathbf{x} , which represents the system states $\mathbf{x}^* \in \mathbb{R}^{n_{x^*}}$, can be augmented with the unknown inputs $\mathbf{a} \in \mathbb{R}^{n_a}$ to be estimated:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^* \\ \mathbf{a} \end{bmatrix} \quad (3)$$

In absence of more accurate information, the unknown input can be considered a random walk:

$$\dot{\mathbf{a}} = \mathbf{r}_a \quad (4)$$

where $\mathbf{r}_a \in \mathbb{R}^{n_a}$ is random noise.

Assuming that \mathbf{f} , \mathbf{h} are known together with the previous states $\mathbf{x}_{\tau-1}^+$ and inputs $\mathbf{u}_{\tau-1}$, they can be used to make the a priori prediction:

$$\mathbf{x}_{\tau}^- = \mathbf{f}(\mathbf{x}_{\tau-1}^+, \mathbf{u}_{\tau-1}) \quad (5)$$

When the measurement \mathbf{y}_{τ} then becomes available, it can be used to update the prediction a posteriori. The optimal least-square correction leads to the Kalman filter [5]:

$$\mathbf{y}_{\tau}^- = \mathbf{h}(\mathbf{x}_{\tau}^-, \mathbf{u}_{\tau}) \quad (6)$$

$$\mathbf{P}_{\tau}^- = \mathbf{F} \mathbf{P}_{\tau-1}^+ \mathbf{F}^T + \mathbf{R}_x \quad (7)$$

$$\mathbf{K}_{\tau} = \mathbf{P}_{\tau}^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_{\tau}^- \mathbf{H}^T + \mathbf{R}_y)^{-1} \quad (8)$$

$$\mathbf{x}_{\tau}^+ = \mathbf{x}_{\tau}^- + \mathbf{K}_{\tau} (\mathbf{y}_{\tau} - \mathbf{y}_{\tau}^-) \quad (9)$$

$$\mathbf{P}_{\tau}^+ = (\mathbf{I}_x - \mathbf{K}_{\tau} \mathbf{H}) \mathbf{P}_{\tau}^- \quad (10)$$

Where $\mathbf{P} \in \mathbb{R}^{n_x \times n_x}$ indicates the state covariance, $\mathbf{K} \in \mathbb{R}^{n_x \times n_y}$ is the Kalman gain and $\mathbf{R}_x \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{R}_y \in \mathbb{R}^{n_y \times n_y}$ are the supposedly known covariances of, respectively, \mathbf{r}_x , \mathbf{r}_y . $\mathbf{F} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{H} \in \mathbb{R}^{n_y \times n_x}$ are, in the case of linear systems, respectively the state-update and measurement matrices.

2.3 Extended Kalman filter

For non-linear systems, the formulation can be extended performing a local linearisation which leads to:

$$\mathbf{F} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \quad (11)$$

$$\mathbf{H} = \frac{\partial \mathbf{h}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \quad (12)$$

In the next section, a redundant multibody formulation is reduced to minimal coordinates, which permits to define the dynamic function \mathbf{f} in ODEs for multibody systems. It is then plugged into the KF scheme in section 4, where \mathbf{F} and \mathbf{H} are derived.

3 Multibody dynamics

This section reviews the different multibody formulations and the model order reduction procedure.

3.1 Generalized coordinates

Several formulations have been proposed in literature to describe the dynamics of a multibody model. In general, a system can be described by the Lagrange's equations:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} + \left(\frac{\partial \mathbf{c}_q(\mathbf{q})}{\partial \mathbf{q}} \right)^T \boldsymbol{\lambda}_c &= \mathbf{u}_q(\mathbf{q}) \\ \mathbf{c}_q(\mathbf{q}) &= \mathbf{0}_c \end{aligned} \quad (13)$$

where $\mathbf{q} \in \mathbb{R}^{n_q}$ is the vector of generalized coordinates of the system, $\mathbf{u}_q \in \mathbb{R}^{n_q}$ is the vector of (non-conservative) external forces, $\boldsymbol{\lambda}_c \in \mathbb{R}^{n_c}$ is the vector of Lagrange multipliers, $\mathbf{c}_q \in \mathbb{R}^{n_c}$ is the constraints function. \mathcal{L} is the Lagrangian function that, in the case of rigid mechanisms, corresponds to the kinetic energy, assuming that there are no additional energy-preserving forces:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}_q(\mathbf{q}) \dot{\mathbf{q}} \quad (14)$$

where $\mathbf{M}_q \in \mathbb{R}^{n_q \times n_q}$ is the mass matrix.

Generally, the coordinates \mathbf{q} can be a composition of position and angular DOFs.

3.2 Natural coordinates

A particular choice is the use of natural coordinates [2] $\mathbf{q}_n \in \mathbb{R}^{n_n}$, where (redundant) position-only DOFs are employed. As proposed in [17], the motion of each body b can be described by 12 variables, the elements of one point \mathbf{o}_b and of the rotation matrix \mathbf{R}_b , for the B bodies:

$$\mathbf{o}_b = [o_{b,x}, o_{b,y}, o_{b,z}]^T \quad (15)$$

$$\begin{aligned} \mathbf{R}_b &= [\mathbf{r}_{b,x}, \mathbf{r}_{b,y}, \mathbf{r}_{b,z}] \\ &= [[r_{b,1}, r_{b,2}, r_{b,3}]^T, [r_{b,4}, r_{b,5}, r_{b,6}]^T, [r_{b,7}, r_{b,8}, r_{b,9}]^T] \end{aligned} \quad (16)$$

$$\mathbf{q}_{n,b} = [\mathbf{o}_b^T, \mathbf{r}_{b,x}^T, \mathbf{r}_{b,y}^T, \mathbf{r}_{b,z}^T]^T \quad (17)$$

$$\mathbf{q}_n = [\mathbf{q}_{n,1}^T, \dots, \mathbf{q}_{n,b}^T, \dots, \mathbf{q}_{n,B}^T]^T \quad (18)$$

Choosing position-only DOFs allows to express a constant mass matrix $\mathbf{M}_q(\mathbf{q}_n) = \mathbf{M}_n \in \mathbb{R}^{n_n \times n_n}$, with the added benefit that the velocity inertia forces vanish. Thus, Eq. (13) becomes the DAEs:

$$\begin{aligned} \mathbf{M}_n \ddot{\mathbf{q}}_n + \left(\frac{\partial \mathbf{c}_n(\mathbf{q}_n)}{\partial \mathbf{q}_n} \right)^T \boldsymbol{\lambda}_c &= \mathbf{u}_n(\mathbf{q}_n) \\ \mathbf{c}_n(\mathbf{q}_n) &= \mathbf{0}_c \end{aligned} \quad (19)$$

Given its inertia and rotation parametrization, a generic (redundant DOFs) multibody model can be post-processed into NC [14]. Thus, NC are considered, without loss of generality, the starting formulation, in order to benefit from the constant mass matrix definition.

3.3 Minimal coordinates

As an alternative, it is possible to use a minimal amount of DOFs, as in the minimal coordinates [3] \mathbf{q}_m , which do not require additional constraints. Thus, it leads to the ODEs:

$$\mathbf{M}_m(\mathbf{q}_m) \ddot{\mathbf{q}}_m + \underline{\mathbf{G}}_m(\mathbf{q}_m) [\dot{\mathbf{q}}_m \dot{\mathbf{q}}_m^T] = \mathbf{u}_m(\mathbf{q}_m) \quad (20)$$

where $\mathbf{M}_m \in \mathbb{R}^{n_m \times n_m}$ and $\underline{\mathbf{G}}_m \in \mathbb{R}^{n_m \times n_m \times n_m}$ are respectively the MC mass matrix and the MC gyroscopic tensor.

The aim is to reduce the initial redundant DOFs to the minimal coordinates and eliminate the need of algebraic constraints.

3.4 Model order reduction

Knowing the non-linear mappings that relate the NC and the MC:

$$\mathbf{q}_m = \mathbf{g}_m(\mathbf{q}_n) \quad (21)$$

$$\mathbf{q}_n = \mathbf{g}_n(\mathbf{q}_m) \quad (22)$$

the model initially expressed (or post-processed into) NC can be projected into the corresponding MC model through:

$$\mathbf{M}_m = \left(\frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \right)^T \mathbf{M}_n \frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \quad (23)$$

$$\underline{\mathbf{G}}_m = \left(\frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \right)^T \mathbf{M}_n \frac{\partial^2 \mathbf{g}_n}{\partial \mathbf{q}_m \partial \mathbf{q}_m} \quad (24)$$

$$\mathbf{u}_m = \left(\frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \right)^T \mathbf{u}_n \quad (25)$$

as showed in [14] where it is proposed to use deep learning to approximate the mappings \mathbf{g}_m , \mathbf{g}_n as in general they may not be available or analytically obtainable.

Such procedure thus permits to reduce the original DAEs in NC \mathbf{q}_n to ODEs in MC \mathbf{q}_m which can be solved with a more versatile choice of integration schemes and straightforwardly included in an estimation framework. The reference to the system full coordinates can be re-obtained with a back-projection as in Eq. (22):

$$\mathbf{q}_n = \mathbf{g}_n(\mathbf{q}_m)$$

whose time derivatives are:

$$\dot{\mathbf{q}}_n = \frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \dot{\mathbf{q}}_m \quad (26)$$

$$\ddot{\mathbf{q}}_n = \frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \ddot{\mathbf{q}}_m + \frac{\partial^2 \mathbf{g}_n}{\partial \mathbf{q}_m \partial \mathbf{q}_m} [\dot{\mathbf{q}}_m \dot{\mathbf{q}}_m^T] \quad (27)$$

Thus, the model order reduction procedure presented in [14] which allows to obtain the minimal coordinate model for a generic multibody system is here exploited in order to obtain the ODE formulation which can then be combined in the estimation scheme. The procedure with the required system and measurement equations (and related derivatives necessary for the Kalman update) is presented in the following section.

4 State and input estimation for multibody models

In this section, the MC multibody model is embedded in a KF scheme for coupled state and input estimation. The given equations are general for any number of MC n_m and augmented states n_a .

4.1 System equation

Given a mechanical application where the mechanism is represented with a multibody model, some external inputs acting on the system may be unknown because they are too difficult or impractical to measure. The system dynamics of Eq. (20) can thus be rewritten as:

$$\mathbf{M}_m(\mathbf{q}_m) \ddot{\mathbf{q}}_m + \underline{\mathbf{G}}_m(\mathbf{q}_m) [\dot{\mathbf{q}}_m \dot{\mathbf{q}}_m^T] = \mathbf{u}_m(\mathbf{q}_m) + \mathbf{a}_m(\mathbf{a}) \quad (28)$$

where $\mathbf{a}_m \in \mathbb{R}^{n_m}$, whose expression is defined in subsection 4.3, is an additional force term dependent on the unknown parameters $\mathbf{a} \in \mathbb{R}^{n_a}$ which can be considered augmented states to be estimated.

As shown in section 2, state-estimators are usually derived for first-order systems, with few exception (for example [18]). Thus, the system equation can be rewritten as a first-order relation in the states \mathbf{x} :

$$\mathbf{x} = \begin{bmatrix} \mathbf{q}_m \\ \dot{\mathbf{q}}_m \\ \mathbf{a} \end{bmatrix} \quad (29)$$

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{q}}_m \\ \ddot{\mathbf{q}}_m \\ \dot{\mathbf{a}} \end{bmatrix} = \mathbf{f}^*(\mathbf{x}, \mathbf{u}_m) \quad (30)$$

where as in Eq. (4), the augmented state \mathbf{a} is assumed a random walk:

$$\dot{\mathbf{a}} = \mathbf{r}_a$$

and from Eq. (28):

$$\ddot{\mathbf{q}}_m = \mathbf{M}_m^{-1} (\mathbf{u}_m + \mathbf{a}_m - \underline{\mathbf{G}}_m [\dot{\mathbf{q}}_m \dot{\mathbf{q}}_m^T]) \quad (31)$$

Here, in order to discretize \mathbf{f}^* into \mathbf{f} , a forward Euler solver is used leading to:

$$\mathbf{x}_\tau = \mathbf{f}(\mathbf{x}_{\tau-1}, \mathbf{u}_{m,\tau-1}) \quad (32)$$

$$\begin{bmatrix} \mathbf{q}_{m,\tau} \\ \dot{\mathbf{q}}_{m,\tau} \\ \mathbf{a}_\tau \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{m,\tau-1} \\ \dot{\mathbf{q}}_{m,\tau-1} \\ \mathbf{a}_{\tau-1} \end{bmatrix} + \Delta t \begin{bmatrix} \dot{\mathbf{q}}_{m,\tau-1} \\ \ddot{\mathbf{q}}_{m,\tau-1} \\ \mathbf{0}_a \end{bmatrix} + \begin{bmatrix} \mathbf{0}_m \\ \mathbf{0}_m \\ \mathbf{r}_a \end{bmatrix} \quad (33)$$

It is thus possible to derive \mathbf{F} :

$$\mathbf{F} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}_m)}{\partial \mathbf{x}} = \mathbf{I}_x + \Delta t \begin{bmatrix} \mathbf{0}_{m,m} & \mathbf{I}_m & \mathbf{0}_{m,a} \\ \frac{\partial \ddot{\mathbf{q}}_m}{\partial \mathbf{q}_m} & \frac{\partial \ddot{\mathbf{q}}_m}{\partial \dot{\mathbf{q}}_m} & \frac{\partial \ddot{\mathbf{q}}_m}{\partial \mathbf{a}} \\ \mathbf{0}_{a,m} & \mathbf{0}_{a,m} & \mathbf{0}_{a,a} \end{bmatrix} \quad (34)$$

where:

$$\begin{aligned} \frac{\partial \ddot{\mathbf{q}}_m}{\partial \mathbf{q}_m} &= \frac{\partial \mathbf{M}_m^{-1}}{\partial \mathbf{q}_m} (\mathbf{u}_m + \mathbf{a}_m - \underline{\mathbf{G}}_m [\dot{\mathbf{q}}_m \dot{\mathbf{q}}_m^\top]) + \\ &+ \mathbf{M}_m^{-1} \left(\frac{\partial \mathbf{u}_m}{\partial \mathbf{q}_m} + \frac{\partial \mathbf{a}_m}{\partial \mathbf{q}_m} - \frac{\partial \underline{\mathbf{G}}_m}{\partial \mathbf{q}_m} [\dot{\mathbf{q}}_m \dot{\mathbf{q}}_m^\top] \right) \end{aligned} \quad (35)$$

$$\frac{\partial \ddot{\mathbf{q}}_m}{\partial \dot{\mathbf{q}}_m} = \mathbf{M}_m^{-1} \left(\frac{\partial \mathbf{a}_m}{\partial \dot{\mathbf{q}}_m} - 2 \underline{\mathbf{G}}_m \dot{\mathbf{q}}_m \right) \quad (36)$$

$$\frac{\partial \ddot{\mathbf{q}}_m}{\partial \mathbf{a}} = \mathbf{M}_m^{-1} \left(\frac{\partial \mathbf{a}_m}{\partial \mathbf{a}} \right) \quad (37)$$

with $\frac{\partial \mathbf{a}_m}{\partial \mathbf{a}}$ defined in Section 4.3 and:

$$\begin{aligned} \frac{\partial \mathbf{M}_m}{\partial \mathbf{q}_m} &= \left(\frac{\partial^2 \mathbf{g}_n}{\partial \mathbf{q}_m \partial \mathbf{q}_m} \right)^\top \mathbf{M}_n \frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} + \left(\frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \right)^\top \mathbf{M}_n \frac{\partial^2 \mathbf{g}_n}{\partial \mathbf{q}_m \partial \mathbf{q}_m} \\ &= \underline{\mathbf{G}}_m^\top + \underline{\mathbf{G}}_m \end{aligned} \quad (38)$$

$$\frac{\partial \mathbf{M}_m^{-1}}{\partial \mathbf{q}_m} = -\mathbf{M}_m^{-1} \frac{\partial \mathbf{M}_m}{\partial \mathbf{q}_m} \mathbf{M}_m^{-1} \quad (39)$$

$$\frac{\partial \mathbf{u}_m}{\partial \mathbf{q}_m} = \left(\frac{\partial^2 \mathbf{g}_n}{\partial \mathbf{q}_m \partial \mathbf{q}_m} \right)^\top \mathbf{u}_n \quad (40)$$

$$\begin{aligned} \frac{\partial \underline{\mathbf{G}}_m}{\partial \mathbf{q}_m} &= \left(\frac{\partial^2 \mathbf{g}_n}{\partial \mathbf{q}_m \partial \mathbf{q}_m} \right)^\top \mathbf{M}_n \frac{\partial^2 \mathbf{g}_n}{\partial \mathbf{q}_m \partial \mathbf{q}_m} + \\ &+ \left(\frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \right)^\top \mathbf{M}_n \frac{\partial^3 \mathbf{g}_n}{\partial \mathbf{q}_m \partial \mathbf{q}_m \partial \mathbf{q}_m} \end{aligned} \quad (41)$$

4.2 Measurement equation

As measurements, it is assumed to employ only accelerometers. Recalling the definition of NC displacements for each body, Eq. (17), and for the full system,

Eq. (18), they can be extended to NC accelerations:

$$\ddot{\mathbf{q}}_{n,b} = [\ddot{\mathbf{o}}_b^T, \ddot{\mathbf{r}}_{b,x}^T, \ddot{\mathbf{r}}_{b,y}^T, \ddot{\mathbf{r}}_{b,z}^T]^T \quad (42)$$

$$\ddot{\mathbf{q}}_n = [\ddot{\mathbf{q}}_{n,1}^T, \dots, \ddot{\mathbf{q}}_{n,b}^T, \dots, \ddot{\mathbf{q}}_{n,B}^T]^T \quad (43)$$

where $\ddot{\mathbf{q}}_{n,b} \in \mathbb{R}^{12}$ is the body NC acceleration for the $b = 1, \dots, B$ bodies and $\ddot{\mathbf{q}}_n \in \mathbb{R}^{n_n}$ is the full-system NC acceleration.

Thus, assuming that the NC points \mathbf{o}_b are chosen in correspondence of the accelerometer positions, the measurements \mathbf{y} can be written as:

$$\mathbf{y} = \mathbf{S}_{y \in n} \ddot{\mathbf{q}}_n \quad (44)$$

where $\mathbf{S}_{y \in n} \in \mathbb{R}^{n_y \times n_n}$ is the (generalized Kronecker delta) sparse matrix to select the n_y accelerometers channels from the n_n natural coordinate accelerations.

Thus, recalling Eq. (27) for the acceleration, the measurement equation becomes:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}_m) = \mathbf{S}_{y \in n} \ddot{\mathbf{q}}_n = \mathbf{S}_{y \in n} \left(\frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \ddot{\mathbf{q}}_m + \frac{\partial^2 \mathbf{g}_n}{\partial \mathbf{q}_m \partial \mathbf{q}_m} [\dot{\mathbf{q}}_m \dot{\mathbf{q}}_m^T] \right) \quad (45)$$

And it is possible to derive \mathbf{H} :

$$\mathbf{H} = \frac{\partial \mathbf{h}(\mathbf{x}, \mathbf{u}_m)}{\partial \mathbf{x}} = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{q}_m} \quad \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}_m} \quad \frac{\partial \mathbf{h}}{\partial \mathbf{a}} \right] \quad (46)$$

where:

$$\frac{\partial \mathbf{h}}{\partial \mathbf{q}_m} = \mathbf{S}_{y \in n} \left[\frac{\partial^2 \mathbf{g}_n}{\partial \mathbf{q}_m \partial \mathbf{q}_m} \ddot{\mathbf{q}}_m + \frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \frac{\partial \ddot{\mathbf{q}}_m}{\partial \mathbf{q}_m} + \frac{\partial^3 \mathbf{g}_n}{\partial \mathbf{q}_m \partial \mathbf{q}_m \partial \mathbf{q}_m} [\dot{\mathbf{q}}_m \dot{\mathbf{q}}_m^T] \right] \quad (47)$$

$$\frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}_m} = \mathbf{S}_{y \in n} \left[\frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \frac{\partial \ddot{\mathbf{q}}_m}{\partial \dot{\mathbf{q}}_m} + 2 \frac{\partial^2 \mathbf{g}_n}{\partial \mathbf{q}_m \partial \mathbf{q}_m} \dot{\mathbf{q}}_m \right] \quad (48)$$

$$\frac{\partial \mathbf{h}}{\partial \mathbf{a}} = \mathbf{S}_{y \in n} \left[\frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \frac{\partial \ddot{\mathbf{q}}_m}{\partial \mathbf{a}} \right] \quad (49)$$

In case the accelerometer rotates in space, the local body-frame acceleration $\ddot{\mathbf{o}}_b^*$, which supposedly has the same orientation of the accelerometer, can be retrieved with the body rotation matrix \mathbf{R}_b as:

$$\ddot{\mathbf{o}}_b^* = \mathbf{R}_b \ddot{\mathbf{o}}_b \quad (50)$$

and used instead of the global-frame acceleration $\ddot{\mathbf{o}}_b$. In this situation, the derivative of \mathbf{R}_b with respect to the MC \mathbf{q}_m has to be included in Eq. (47).

It is also highlighted that the assumption for which the NC points are chosen in correspondence of the accelerometer positions does not impact generality since, if it is not the case, an affine transformation can be performed:

$$\begin{aligned} \mathbf{q}_{n,b}^* &= \mathbf{V}_i(\mathbf{o}_i, \mathbf{R}_i) \mathbf{q}_{n,b} \\ &= \left[\begin{bmatrix} 1, & \mathbf{o}_i^T \\ \mathbf{0}_3, & \mathbf{R}_i \end{bmatrix} \otimes \mathbf{I}_3 \right] \mathbf{q}_{n,b} \end{aligned} \quad (51)$$

where, for body b , the initial $\mathbf{q}_{n,b}$ can be projected into the i -accelerometer position $\mathbf{q}_{n,b}^*$ through the constant projection matrix $\mathbf{V}_i \in R^{12 \times 12}$ given the accelerometer initial position and orientation, respectively $\mathbf{o}_i \in \mathbb{R}^3$, $\mathbf{R}_i \in \mathbb{R}^{3 \times 3}$, with respect to the local body frame.

4.3 Augmented state

In certain cases, an external force \mathbf{a}_m dependent on some unknown input may be acting on the system and its estimation can be performed including an augmented state \mathbf{a} to track (and estimate) the related unknown input.

Assuming that the projection matrix of external inputs into NC is known, it leads to:

$$\mathbf{a}_n = \mathbf{S}_n(\mathbf{q}_m) \mathbf{a} \quad (52)$$

$$\mathbf{a}_m = \left(\frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \right)^T \mathbf{S}_n \mathbf{a} \quad (53)$$

where $\mathbf{a}_n \in \mathbb{R}^{n_n}$, $\mathbf{a}_m \in \mathbb{R}^{n_m}$ are respectively the NC and MC projection of the unknown external force and $\mathbf{S}_n \in \mathbb{R}^{n_n \times n_a}$ is the input projection matrix whose expression is given in App. A.

Then, it is possible to determine the derivatives required for the calculation of \mathbf{F} , \mathbf{H} :

$$\frac{\partial \mathbf{a}_m}{\partial \mathbf{q}_m} = \left(\frac{\partial^2 \mathbf{q}_n}{\partial \mathbf{q}_m \partial \mathbf{q}_m} \right)^T \mathbf{S}_n \mathbf{a} + \left(\frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \right)^T \frac{\partial (\mathbf{S}_n \mathbf{a})}{\partial \mathbf{q}_m} \quad (54)$$

$$\frac{\partial \mathbf{a}_m}{\partial \dot{\mathbf{q}}_m} = \mathbf{0}_{m,m} \quad (55)$$

$$\frac{\partial \mathbf{a}_m}{\partial \mathbf{a}} = \left(\frac{\partial \mathbf{g}_n}{\partial \mathbf{q}_m} \right)^T \mathbf{S}_n \quad (56)$$

With these quantities, all contributions are known for evaluating the augmented state-estimator.

4.4 Observability analysis

An essential part in the deployment of estimation schemes is to ensure the long term stability, or boundedness, of the estimator covariance. If the system is *observable*, the long term stability of the estimator is guaranteed [5]. Ensuring observability in multibody systems and input estimators is often particularly challenging as steady-state drift can occur when no position level measurements are available [11, 19]. As a pragmatic approach, *dummy measurements* can be added to stabilize these measurements, but this scheme requires additional tuning and can lead to a reduced accuracy [20].

The (local) observability of a system can be assessed through the Popov-Belevitch-Hautus (PBH) observability matrix [21]. For the proposed scheme, the following PBH observability matrix \mathbf{O}_{PBH} can be obtained:

$$\begin{aligned} \mathbf{O}_{PBH} &= \begin{bmatrix} \mathbf{F}^* - s \mathbf{I}_x \\ \mathbf{H} \end{bmatrix} \\ &= \begin{bmatrix} -s \mathbf{I}_m & \mathbf{I}_m & \mathbf{0}_{m,a} \\ \frac{\partial \dot{\mathbf{q}}_m}{\partial \mathbf{q}_m} & \left(\frac{\partial \ddot{\mathbf{q}}_m}{\partial \mathbf{q}_m} - s \mathbf{I}_m \right) & \frac{\partial \ddot{\mathbf{q}}_m}{\partial \mathbf{a}} \\ \mathbf{0}_{a,m} & \mathbf{0}_{a,m} & -s \mathbf{I}_a \\ \frac{\partial \mathbf{h}}{\partial \mathbf{q}_m} & \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}_m} & \frac{\partial \mathbf{h}}{\partial \mathbf{a}} \end{bmatrix} \end{aligned} \quad (57)$$

where $\mathbf{F}^* \in \mathbb{R}^{n_x \times n_x}$ is the continuous time counterpart of \mathbf{F} . This system is (locally) observable if:

$$\text{rank}(\mathbf{O}_{PBH}) = n_x \quad \forall s \in \mathbb{C} \quad (58)$$

where \mathbb{C} is the complex domain and s represents the eigenvalues of \mathbf{F}^* to assess the observability at the corresponding eigenfrequencies. A full rank should be obtained from the combination of the system and measurement matrices in order to establish the observability. For $s = 0$, the upper part of the observability matrix becomes singular and this is typically where issues occur for classical multibody model-based estimators, requiring additional position-level measures as, for instance, in [11]. However when considering the measurement equations from Eqs. (45)-(49), it can be seen that the structure of the measurements enables a full column rank observability matrix \mathbf{O}_{PBH} . This is obtained thanks to the independent coordinate formulation and it implies that, for the proposed approach, a fully observable system is obtained in the case of $n_a \leq n_x \leq n_y$, provided that the n_y measurements are linearly independent. The estimation scheme can therefore be reliably deployed over long time frames, which is key when considering multibody applications. In particular, in the validation case of the next section, it is demonstrated that a single unknown input force and a single accelerometer measurement lead to a stable estimation; in general, the observability analysis should be repeated for different combinations of sensors.

5 Validation

In this section, the experimental validation of the methodology is presented.

5.1 Slider-crank mechanism

The proposed approach is validated on a slider-crank mechanism. The original multibody model is based on the CAD representation of the test rig and composed by 10 rigid bodies: the base block, the motor support, the crank support,

the motor, the motor shaft, the crank shaft, the crank, the connecting rod, the slider, the track.

The NC description which consists of 12 variables per body (centre of gravity position plus parameters of the rotation matrix) is employed, according to [17]. The bodies are linked by joints which constraint the relative motion, leading to one rigid degree of freedom. In particular: the motor shaft is connected to the motor with a revolute joint and fixed to the crank; the crank is connected to the connecting rod with a revolute joint; the connecting rod is connected to the ground with a sliding joint; the remaining bodies are fixed to the ground (they are reported for completeness even if not impacting the dynamic analysis). The constraint equations are supposedly unknown and not used in the proposed methodology as presented in the model order reduction procedure of Sec. 3.4.

The experimental setup and the corresponding multibody model are shown in Fig. 1, while the list of bodies and related inertia properties are reported in Tab. 1.

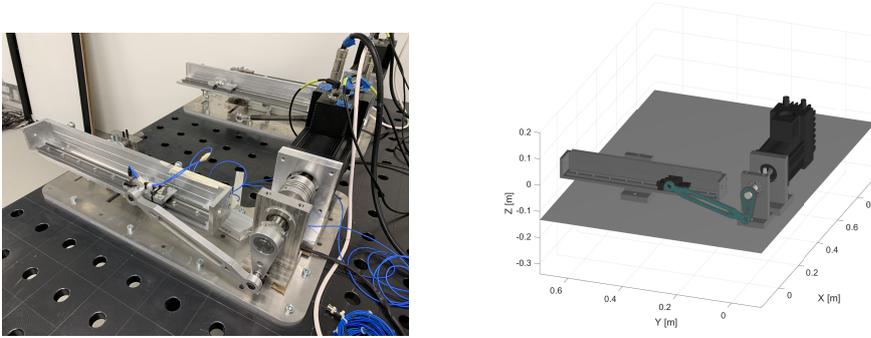


Fig. 1 the slider-crank mechanism. On the left, the experimental setup; on the right, the corresponding multibody model.

body	$m[kg]$	$J_{xx}[kg \cdot m^2]$	$J_{yy}[kg \cdot m^2]$	$J_{zz}[kg \cdot m^2]$
Base Block	3217.5	292.256	627.413	871.406
Motor Support	1.349	$6.052 \cdot 10^{-3}$	$3.89 \cdot 10^{-3}$	$2.346 \cdot 10^{-3}$
Crank Support	0.841	$2.678 \cdot 10^{-3}$	$1.954 \cdot 10^{-3}$	$8.04e \cdot 10^{-4}$
Motor	14.2	0.0286	0.0937	0.0937
Motor Shaft	1	$1.214 \cdot 10^{-3}$	$2.997 \cdot 10^{-3}$	$2.997 \cdot 10^{-3}$
Crank Shaft	0.496	$3.873 \cdot 10^{-5}$	$7.069 \cdot 10^{-4}$	$7.069 \cdot 10^{-4}$
Crank	0.183	$1.652 \cdot 10^{-37}$	$4.015 \cdot 10^{-39}$	$1.506 \cdot 10^{-35}$
Connecting Rod	0.254	$7.535 \cdot 10^{-34}$	$5.477 \cdot 10^{-39}$	$2.897 \cdot 10^{-35}$
Slider	0.219	$2.271 \cdot 10^{-4}$	$9.641 \cdot 10^{-5}$	$2.808 \cdot 10^{-4}$
Track	13.774	0.331	0.0523	0.334

Table 1 the list of bodies for the slider-crank model with the respective mass m and inertia moments J .

The brushless servomotor is a MAC3000 with integrated controller MAC00-B4 from JVL (www.jvl.dk). The motor encoder sensor returns the measured rotation angle and velocity of the shaft, respectively indicated as θ and $\dot{\theta}$.

In addition, a monoaxial MEMS accelerometer (3711D1FB200G) from PCB (www.pcb.com) is placed on the slider to measure its translational acceleration along the y-axis, indicated as \ddot{Y} .

A target on the (rotational) velocity is given to the controller which subsequently determines the motor torque, indicated as T .

5.2 Input estimation

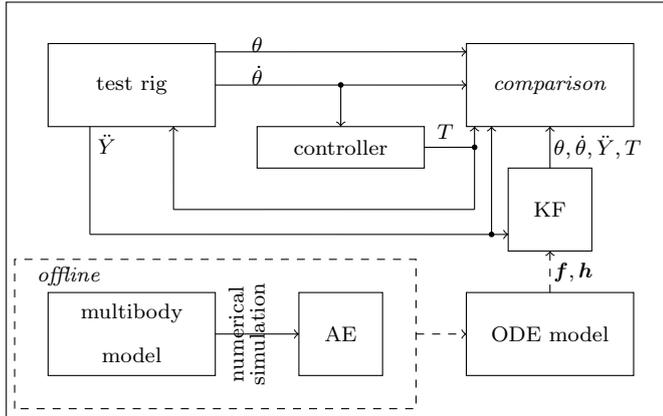


Fig. 2 diagram of the coupled state-input estimation scheme and signal comparisons. θ and $\dot{\theta}$ are respectively the crank angle and rotational velocity; \ddot{Y} is the slider translational acceleration; T is the motor torque. \mathbf{f} and \mathbf{h} indicate respectively the system and measurement equations.

The original NC multibody model is reduced to MC according to the procedure presented in section 3. The crank angle is chosen as minimal DOF and the MC-NC mapping is obtained with a (periodic) neural network as introduced in [14, 16].

In this application, the neural network approximating the \mathbf{g}_n mapping consists of two non-linear layers with 8 units (each) and the (hyperbolic tangent) sigmoid activation function [22] to ensure the required twice differentiability and a final (linear) regression layer to reconstruct the original NC. The neural network model is built in Tensorflow [23], a common machine learning library. A numerical simulation of the NC model is used in the (offline) neural network training to obtain the reduced-order MC model, without any experimental data which are employed only in the Kalman filtering validation. The neural network training is performed with the Adam optimization scheme [24] fed with batches [22] of the training set; 10% of the data are employed as

test set for the early stopping regularization [25]; 10 000 epochs (i.e. training iterations) are employed, taking about 30 minutes on a consumer laptop.

The obtained MC model is then embedded in the ADE-KF as showed in section 4. The unknown input torque is estimated through an augmented state and the Kalman filter update is based solely on the slider y -acceleration. The crank rotational angle and velocity are also retrieved with the reduced order model as additional virtual measures, but they are used merely for comparison purposes and not employed in the estimator.

A diagram of the coupled state-input estimation scheme with the time signals retrieval and comparison is reported in Fig. 2.

A (rotational) velocity profile of 10 seconds and composed by three levels (respectively 40, 50, 60 rad/s) is given to the slider-crank controller. The noise on the multibody states – i.e. MC position and velocity – is supposed zero and all the noise is assumed on the augmented state as $r_a = 1.5 \cdot 10^{-4}$.

A comparison between the measured and the estimated quantities – i.e. the crank angle and rotational velocity, the slider translational acceleration – is reported in Fig. 3, with a zoom-in in Fig. 4 to show the start-up and the two velocity transitions.

The supposedly unknown input torque is shown in Fig. 5 for the full time-series and the velocity transitions, where the difference in amplitude between the real and the estimated quantity can be ascribed to the un-modelled friction acting on the slider.

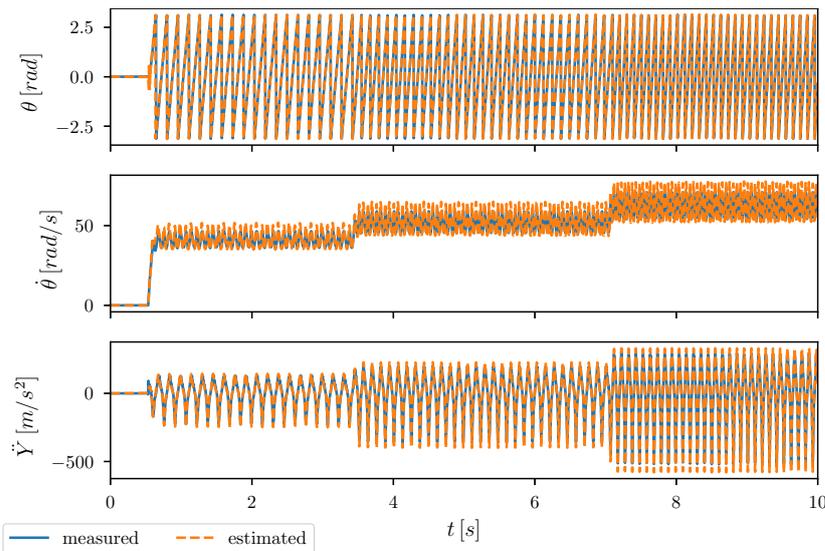


Fig. 3 comparison between the measured and estimated quantities for the crank angle θ and rotational velocity $\dot{\theta}$; slider translational acceleration \ddot{Y} . Only the latter sensor is used for the Kalman filter update.

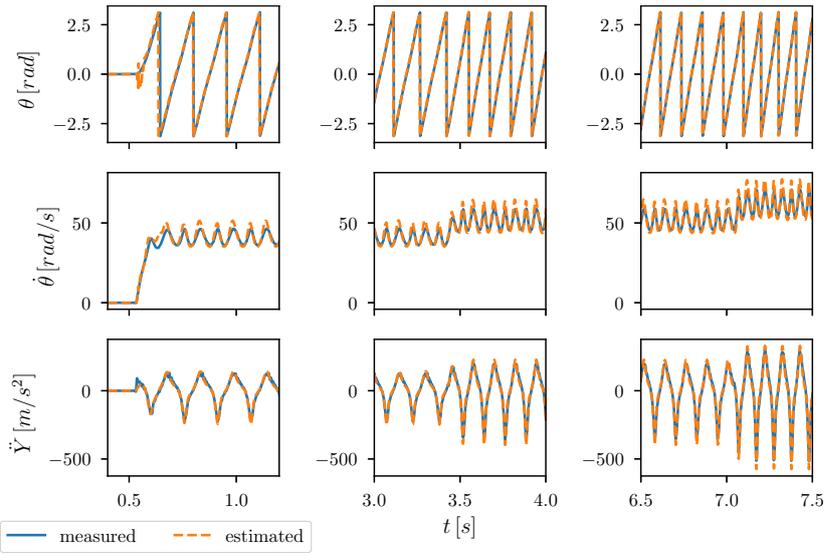


Fig. 4 comparison between the measured and estimated quantities for the crank angle θ and rotational velocity $\dot{\theta}$; slider translational acceleration \dot{Y} . Zoom-in (per column) on the velocity transitions.

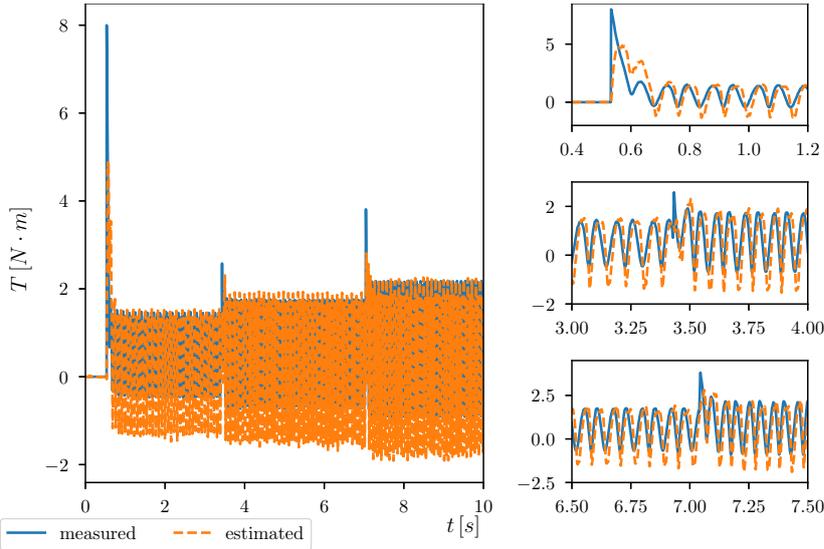


Fig. 5 comparison between the real and estimated torque T . On the left, full time series; on the right, zoom-in on the velocity transitions.

In order to better match the peaks in the velocity transitions, the assumed noise on the augmented state can be increased to, for instance, $\mathbf{r}_a = 5 \cdot 10^{-3}$ leading to the results shown in Fig. 6 and Fig. 7 for the sensor quantities and in Fig. 8 for the input torque estimation. While the peaks, in particular the

initial one at start-up, can be better matched, as expected the noise content in the estimation is higher. It can be noticed that the stability of the filter is not impacted, as shown by the comparison of the augmented state evolution in the two cases, reported in Fig. 9.

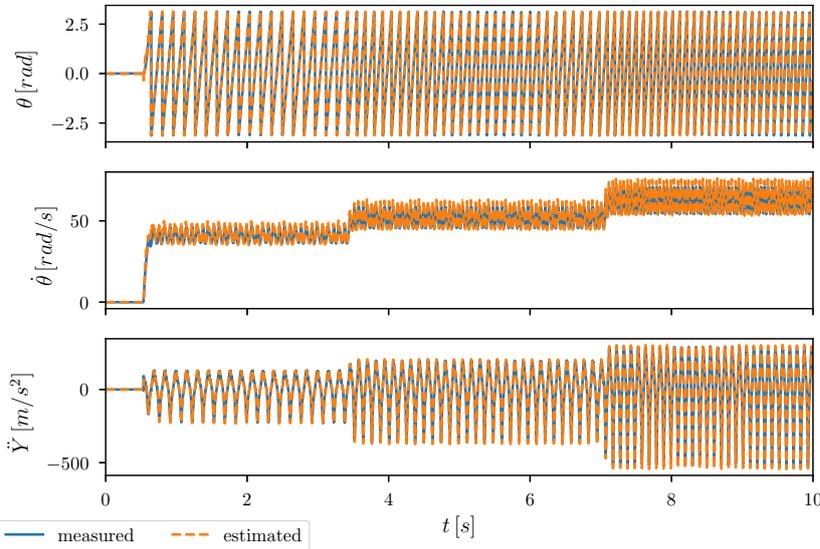


Fig. 6 comparison between the measured and estimated quantities for the crank angle θ and rotational velocity $\dot{\theta}$; slider translational acceleration \dot{Y} . Case with assumed augmented state noise $\mathbf{r}_a = 5 \cdot 10^{-3}$.

In the case information on the input torque is available (based on the application, e.g. periodic input), they could be included in a more detailed augmented-state model instead of the generic random walk in order to improve the estimation results. Similarly, the addition of an accurate friction model is expected to reduce the discrepancy between the motor torque and the estimation.

Finally, it is noted that, while in this specific slider-crank application it would have been possible to employ an analytical mapping between the minimal and the full coordinates, the model order reduction procedure is generally applicable as it does not require access to the constraint equations (which in the case of common multibody software packages may not be accessible) and can be employed also on mechanisms with multiple rigid degrees of freedom or with closed loops (for which analytical mappings cannot be obtained), as shown in [14].

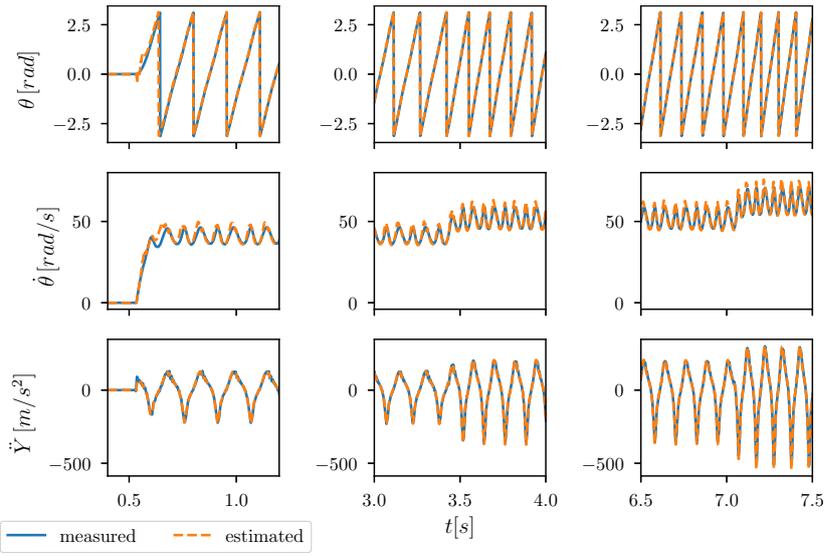


Fig. 7 comparison between the measured and estimated quantities for the crank angle θ and rotational velocity $\dot{\theta}$; slider translational acceleration \dot{Y} . Zoom-in (per column) on the velocity transitions. Case with assumed augmented state noise $r_a = 5 \cdot 10^{-3}$.

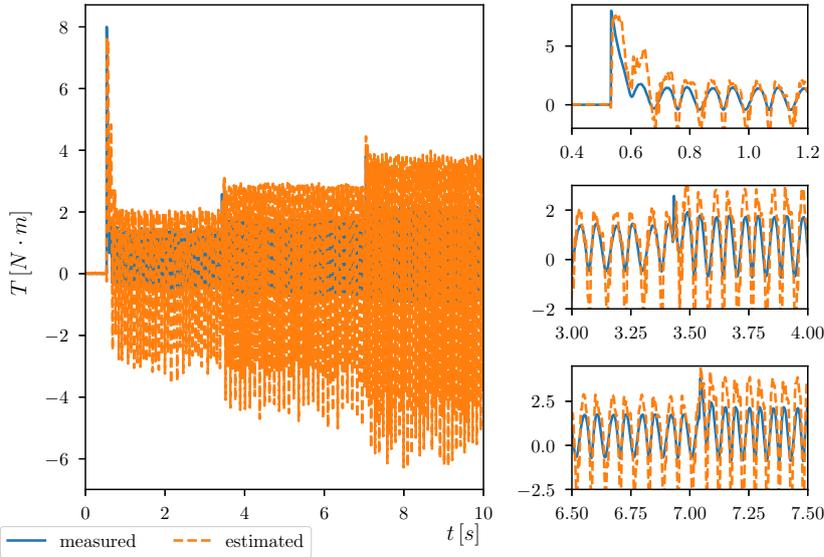


Fig. 8 comparison between the real and estimated torque T . On the left, full timeseries; on the right, zoom-in on the velocity transitions. Case with assumed augmented state noise $r_a = 5 \cdot 10^{-3}$.

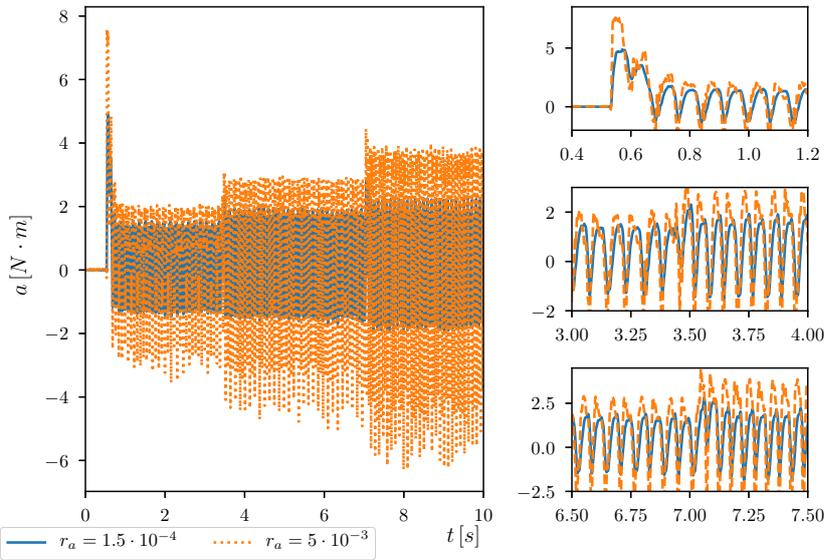


Fig. 9 comparison between the evolution of the augmented state \mathbf{a} with the two different covariance values. While the noise level adapts, the stability of the prediction is not impacted.

6 Conclusion

A model order reduction framework based on deep learning is exploited in order to find a minimal amount of variables to describe a redundant coordinates multibody model. It allows to describe the dynamics with ordinary differential equations which can be easily employed in an estimation scheme. Thus, the application with the augmented discrete extended Kalman filter to concurrently estimate states and unknown inputs (as augmented states) is introduced and the formulation for a generic case is shown. Finally, the methodology is demonstrated on an slider-crank mechanism. The benefit of the novel technique based on a non-linear model order reduction is the possibility to use solely acceleration measurements without compromising the observability of the system.

Acknowledgements

The Research Fund KU Leuven, the Flanders Innovation & Entrepreneurship Agency within the IMPROVED project and the AI impulse program of the Flemish Government are gratefully acknowledged for their support.

A Input projection matrix

Here, the generic input projection into natural coordinates is obtained. Assuming that the external input (to be estimated) \mathbf{a} is a force or a torque in one of the 6 directions:

$$\mathbf{a} = \begin{bmatrix} \mathbf{a}_t \\ \mathbf{a}_r \end{bmatrix} = \begin{bmatrix} a_{t,x} \\ a_{t,y} \\ a_{t,z} \\ a_{r,x} \\ a_{r,y} \\ a_{r,z} \end{bmatrix} \quad (59)$$

where $\mathbf{a}_t, \mathbf{a}_r \in \mathbb{R}^3$ are respectively the translational and rotational components.

If the force/torque is acting on the body b , its projection into natural coordinates $\mathbf{a}_{n,b} \in \mathbb{R}^{12}$ can be obtained as:

$$\mathbf{a}_{n,b} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{a}_t \\ \boldsymbol{\Theta}_b^T & \mathbf{a}_r \end{bmatrix} = \mathbf{S}_{n,b} \mathbf{a} \quad (60)$$

where $\boldsymbol{\Theta}_b \in \mathbb{R}^{3 \times 9}$ is the NC angular transformation matrix:

$$\boldsymbol{\Theta}_b(\mathbf{r}_b) = \left(\frac{\partial \boldsymbol{\theta}_b}{\partial \mathbf{r}_b} \right)^T \quad (61)$$

with $\mathbf{r}_b \in \mathbb{R}^9$ representing the body rotation matrix components as in Eq. (16) and $\angle \boldsymbol{\theta}_f \in \mathbb{R}^3$ as the angular displacement. Knowing [26] the NC force due to a torque along a certain unit vector, the NC angular transformation matrix can be expressed as:

$$\boldsymbol{\Theta}_b = -\frac{1}{2} \begin{bmatrix} [\mathbf{r}_{b,x}]_{\times} \\ [\mathbf{r}_{b,y}]_{\times} \\ [\mathbf{r}_{b,z}]_{\times} \end{bmatrix}^T \quad (62)$$

Therefore, the obtained NC force can be projected into minimal coordinates as:

$$\mathbf{a}_m = \left(\frac{\partial \psi}{\partial \mathbf{q}_m} \right)^T \mathbf{S}_{b \in n}^T \mathbf{a}_{n,b} \quad (63)$$

where $\mathbf{S}_{b \in n} \in \mathbb{Z}^{12 \times n_n}$ is the (generalized Kronecker delta) sparse matrix to select the 12 body coordinates from the n_n natural coordinates and $\mathbf{a}_m \in \mathbb{R}^{n_m}$ is the MC projection of the unknown external forces. If, instead of the reference body frame, the input is applied on the generic body-attached frame f , a projection as in Eq. (51) has to be performed.

It is finally noted that while for the generic case the derivative of the input projection matrix \mathbf{S}_n has to be taken into account in Eq. (54), in the case of purely translational forces it is constant, simplifying the expression.

References

1. E. J. Haug, *Computer aided kinematics and dynamics of mechanical systems*, vol. 1. Allyn and Bacon Boston, 1989.
2. J. García de Jalón and E. Bayo, *Kinematic and dynamic simulation of multibody systems: the real-time challenge*. Springer Science & Business Media, 2012.
3. M. Hiller and A. Kecskemethy, "Dynamics of multibody systems with minimal coordinates," in *Computer-Aided Analysis of Rigid and Flexible Mechanical Systems*, pp. 61–100, Springer, 1994.
4. R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
5. D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

6. R. Kalman and R. Bucy, “New results in linear filtering and prediction theory,” *Journal of Basic Engineering*, vol. 83, no. 3, pp. 95–108, 1961.
7. E. Sanjurjo, M. Á. Naya, J. L. Blanco-Claraco, J. L. Torres-Moreno, and A. Giménez-Fernández, “Accuracy and efficiency comparison of various nonlinear Kalman filters applied to multibody models,” *Nonlinear Dynamics*, vol. 88, no. 3, pp. 1935–1951, 2017.
8. J. Cuadrado, D. Dopico, A. Barreiro, and E. Delgado, “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.
9. J. Cuadrado, D. Dopico, J. Perez, and R. Pastorino, “Automotive observers based on multibody models and the extended Kalman filter,” *Multibody System Dynamics*, vol. 27, no. 1, pp. 3–19, 2012.
10. R. Pastorino, D. Richiedei, J. Cuadrado, and A. Trevisani, “State estimation using multibody models and non-linear Kalman filters,” *International Journal of Non-Linear Mechanics*, vol. 53, pp. 83–90, 2013.
11. E. Risaliti, T. Tamarozzi, M. Vermaut, B. Cornelis, and W. Desmet, “Multibody model based estimation of multiple loads and strain field on a vehicle suspension system,” *Mechanical Systems and Signal Processing*, vol. 123, pp. 1–25, 2019.
12. F. Naets, R. Pastorino, J. Cuadrado, and W. Desmet, “Online state and input force estimation for multibody models employing extended Kalman filtering,” *Multibody System Dynamics*, vol. 32, no. 3, pp. 317–336, 2014.
13. I. Palomba, D. Richiedei, and A. Trevisani, “Reduced-order observers for nonlinear state estimation in flexible multibody systems,” *Shock and Vibration*, vol. 2018, 2018.
14. A. Angeli, W. Desmet, and F. Naets, “Deep learning for model order reduction of multibody systems to minimal coordinates,” *Computer Methods in Applied Mechanics and Engineering*, vol. 373, p. 113517, 2021.
15. A. Angeli, F. Naets, and W. Desmet, “A machine learning approach for minimal coordinate multibody simulation,” in *Multibody Dynamics 2019*, pp. 417–424, Springer, 2020.
16. A. Angeli, F. Naets, and W. Desmet, “Deep learning of (periodic) minimal coordinates for multibody simulations,” in *16th International Conference on Multibody Systems, Nonlinear Dynamics, and Control (MSNDC), presented at-2020 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE2020*, 2020.
17. C. Kraus, M. Winckler, and H. Bock, “Modeling mechanical DAE using natural coordinates,” *Mathematical and Computer Modelling of Dynamical Systems*, vol. 7, no. 2, pp. 145–158, 2001.
18. E. M. Hernandez, “Optimal model-based state estimation in mechanical and structural systems,” *Structural control and health monitoring*, vol. 20, no. 4, pp. 532–543, 2013.
19. E. N. Chatzi and C. Fuggini, “Online correction of drift in structural identification using artificial white noise observations and an unscented Kalman filter,” *Smart Structures and Systems*, vol. 16, no. 2, pp. 295–328, 2015.
20. F. Naets, J. Cuadrado, and W. Desmet, “Stable force identification in structural dynamics using Kalman filtering and dummy-measurements,” *Mechanical Systems and Signal Processing*, vol. 50, pp. 235–248, 2015.
21. B. K. Ghosh and J. Rosenthal, “A generalized Popov-Belevitch-Hautus test of observability,” *IEEE transactions on automatic control*, vol. 40, no. 1, pp. 176–180, 1995.
22. Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, “Efficient backprop,” in *Neural networks: Tricks of the trade*, pp. 9–48, Springer, 2012.
23. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.
24. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
25. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
26. P. Betsch and N. Sanger, “On the consistent formulation of torques in a rotationless framework for multibody dynamics,” *Computers & Structures*, vol. 127, pp. 29–38, 2013.