Journal: Multibody System Dynamics. manuscript No.
(will be inserted by the editor)

Multibody dynamic systems as Bayesian Networks: applications to robust state estimation of mechanisms

J.L. Blanco-Claraco · J.L. Torres-Moreno ·
A. Giménez-Fernández

Authors' accepted manuscript. The final publication is available at Springer via:
http://dx.doi.org/10.1007/s11044-014-9440-9

Abstract This article addresses the problem of robustly estimating the dynamic state of a mechanism from a set of noisy sensor measurements. We start with a rigorous treatment of the problem from the perspective of *graphical models*, a popular formalism in the fields of statistical inference and machine learning. The modeling power of such formalism is demonstrated by showing how the sequential estimation of a mechanism state with an Extended Kalman Filter (EKF), often used in previous works, becomes just one of the possible solutions. As an interesting alternative, we derive the formulation of a Sequential Monte Carlo (SMC) filter, also known as particle filter (PF), suitable for online tracking the state of a rigid mechanism. We validate our ideas with both simulated and real datasets. Moreover, we prove the usefulness of the particle filtering solution for real-work applications due to its unmatched capability of automatically inferring the initial states of the mechanism along with its "assembly configuration" or "branch" if several ones are possible, a feature not matched by any previously-proposed state observer in the multibody literature.

Keywords State observers · EKF · Particle filter · Uncertainty · Rigid Multibody Systems · Virtual sensors

1 Introduction

With the introduction of inexpensive digital computers, the field of computational mechanics witnessed an intense development of numerical methods for analyzing multibody systems (MBS). Much effort has been devoted to devising dynamic formulations and choices of coordinate systems that lead to precise solutions and efficient computer implementations [1–8], since achieving real-time simulation of multibody dynamics (MBD) becomes a must for applications such as Hardware-in-

J.L. Blanco-Claraco, J.L. Torres-Moreno, A. Giménez-Fernández
University of Almería
E-mail: {jlblanco,jltmoreno,agimfer}@ual.es

the-loop (HIL) testing or debugging controllers for mechanical designs that have not yet been built.

Given that efficient simulation of moderate-size rigid MBS could be now considered a partly-solved issue [9–13], the multibody community has opened new research lines that build upon the existence of simulators capable of real-time operation. One of such lines is the topic of the present work, namely, MBS *state observers* or *estimators*, where readings from a set of sensors are employed together with a multibody model of the actual machine to infer its real dynamical state at each instant of time. An important niche of applicability for state observers is the implementation of controllers, where the state subject of control is not directly measured but estimated instead.

Addressing MBS estimators under a probabilistic inference perspective, as done in this work, is motivated by a series of reasons: (i) the need to cope with uncertainty, (ii) the usage of virtual sensors and (iii) robustness. These goals are justified in the following paragraphs.

First of all, any engineering application must deal with *uncertainty*. Inaccuracies during the fabrication and assembling of mechanical parts, imperfect modeling of actuators, long-term weathering and unexpected external events should be all taken into account, at least up to some extent. As an example, we could cite NASA employing Monte Carlo methods in their Mars Science Laboratory landing estimation [14]. Regarding the problem of predicting the effects of unknown parameters in multibody systems, the straightforward approach consists of applying the Fokker-Planck diffusion equation, also known as the Kolmogorov forward equation [15]. More convenient techniques based on generalized polynomial chaos theory has been proposed in the series of works [16–18], which allow obtaining accurate prediction of uncertainties in the dynamic response in both, time and frequency domains.

While those methods are useful in predicting the expected behavior of a system, reconstructing *actual* trajectories in either post-processing stages or online for real-time control, requires *estimators*. Estimation of systems with uncertainty differs from prediction in the inclusion of partial and noisy observations carrying information about the system evolution over time that must be fused with the predictions. To the best of authors knowledge, the application of estimators to multibody dynamics only started in recent years [19–21]. We believe that the rigorous probabilistic treatment proposed in this work based on graphical models may be useful as a solid basis to further developing estimation techniques that deal with uncertainty in mechanical systems.

Secondly, in vehicles, robotic manipulators and other mechanisms it often becomes expensive or cumbersome to obtain measurements of all the physical magnitudes required to perform proper control or to monitor its operation. State observers provide the designer the possibility of using *virtual sensors*, that is, to infer some measurements from readings of different sensors which may be more economic than the replaced one or be easier to install and maintain. Examples of magnitudes that are difficult to observe directly are the torque and forces at some joints, and the tire-soil interaction forces in the analysis of vehicle dynamics.

Thirdly, probabilistic estimators as proposed in this work are able of robustly fusing information from redundant or complementary sensors, leading to more accurate and reliable systems. Additionally, to the best of the authors' knowledge, this work proposes for the first time a statistical framework for multibody estimators capable of distinguishing between a discrete set of "branches" (refer to Figure 3) or operation modes. Apart from the usage demonstrated in the presented experiments, where an estimator is capable of automatically determining the branch of a four-bar linkage, the same framework could be employed to quickly detect different prototypical failures or anomalies that are expected during a machine lifetime.

The rest of this article is structured as follows. Section 2 firstly introduces the concept of graphical model and then proposes such a model for MBD. Next, the recursive filtering equations of a Bayesian filter for a MBS are derived in Section 3, while all its required probabilistic terms are explained in detail in Section 4. The particular case of implementing the observer as a particle filter is addressed in Section 5. Finally, some experiments are presented in Section 6 after which we end with some conclusions.

2 A graphical model for multibody dynamics

In this section we firstly propose a graphical model for the dynamic simulation of rigid MBS. Then, it will be employed in section 3 to rigorously derive the equations of filtering state observers.

2.1 Uncertainty and graphs

As mentioned above, there exist several good reasons to address MBD under a probabilistic point of view. However, working with probabilities in our case means simultaneously handling *probability density functions* (PDFs) of a large number of variables (degrees of freedom, sensor readings, forces, etc.), a task that only becomes manageable if we establish a clear and *structured* model. Here is where graphical models reveal indispensable.

The formalism of *graphical models* is a powerful tool which fuses concepts from statistics and graph theory [22]. Although its origins can be traced back to works by J.W.Gibbs on statistical mechanics (c.1902), the field witnessed a huge development during the last decades. At present, graphical models have become an essential interdisciplinary concept in a multitude of engineering areas [22–24].

A graphical model consists of a graph with nodes and edges, where nodes represent random variables and edges encode relationships (i.e. conditional probabilities) between them. There exist three main kinds of graphical models[1], but in this work we focus on directed acyclic graphical models, also known as *Bayesian networks* (BNs). In such models, edges are directed and can be interpreted as *causal* relationships, i.e. an edge $a \to b$ could be read "a causes b". When the involved variables evolve with time, as is our case in MBD, a different instance of each variable exists for every time step t and we have a *Dynamic Bayesian Network* (DBN).

Interestingly, the *absence* of edges between some of the nodes is the most valuable information in a graph, since it allows us to simplify many probabilistic

[1] The three types are: (i) directed acyclic graphs (or Bayesian Networks) [24], (ii) undirected graphs (Markov random fields) [25] and (iii) factor graphs [26].

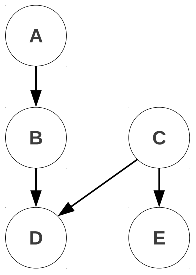


Fig. 1: Example Bayesian model.

expressions. In particular, a graph can encode the *conditional independence* between two sets of variables a and b provided knowledge about another set c, which is denoted as $a \perp\!\!\!\perp b|c$ [27]. The exact rules to determine such independences can be found elsewhere [22], but informally we can say that c "d-separates" a and b if all the possible paths from a to b following the edges in the graph must go through c. Note that the actual criterion takes into account the edge directions, but we refer the details to the excellent existing literature [22].

It becomes instructive to explain the previous definition with the sample BN shown in Figure 1, which encodes the relationships between a set of variables $\{A, B, C, D, E\}$. Here, it follows from the graph that D and E are not independent, in the statistical sense, since both share a common ancestor (C). However, assuming a perfect knowledge about C, the two variables D and E become conditional independent since C blocks all possible paths between them; hence, we have $D \perp\!\!\!\perp E|C$. The utility of this knowledge is allowing us to simplify (*factorize*, in statistical terminology) joint distributions such as $p(D, E|C) = p(D|C)p(E|C)$. The absence of a direct edge between D and E is what allows such a factorization. By applying similar factorizations we can reduce the joint distribution of all five variables in Figure 1 to a product of five simpler distributions:

$$p(A, B, C, D, E) = p(A)p(C)p(B|A)p(D|B, C)p(E|C) \tag{1}$$

which is preferable for the dimensionality reduction of all the involved functions.

Three categories of problems can be addressed with graphical models [23]: (i) learning the structure of the graph itself, (ii) learning the conditional distributions encoded by the edges, and (iii) performing *inference*. The latter, trying to estimate the state of a mechanism from noisy and partial observations of the system, is what motivates our interest on graphical models applied to computational mechanics. The first two problems are not applicable in our context since we can devise a physically-plausible model for both, the graph itself and the conditional distributions, as shown immediately below.

2.2 A graph for MBD

Let the problem of MBD simulation be split into three sets of variables: (i) those describing the dynamic state of the mechanism, (ii) observations or readings from sensors and (iii) the set of all external forces. Following the common notation in

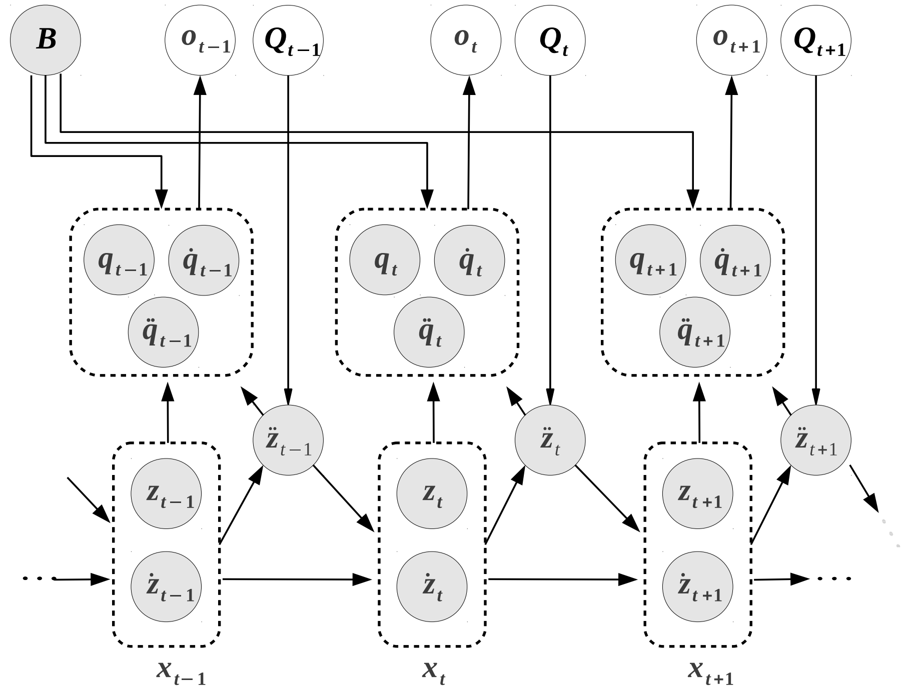
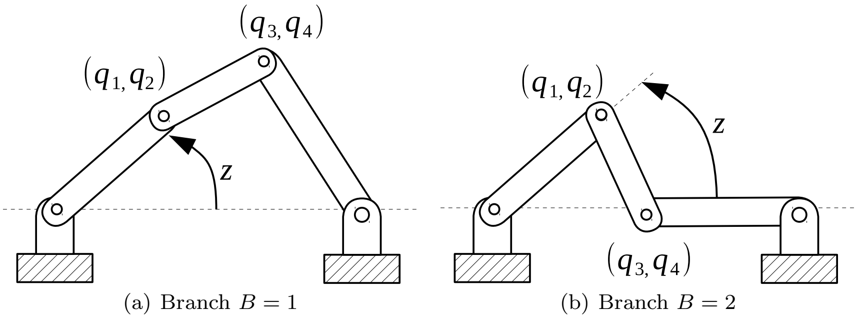


Fig. 2: Dynamic Bayesian Model (DBN) for the discrete-time estimator of a MB system in dependent coordinates. A DBN is a BN where the same pattern of variables repeat for a sequence of time steps –here we depict $t-1$, t and $t+1$ only. The mechanism branch B is the only variable that we assumed to remain static over time, hence its lack of a time step subscript. Shaded and unshaded nodes stand for variables considered in this work as hidden and observed, respectively, while directed arcs represent a causality relationship. Variables that share exactly the same connections have been grouped (dashed rectangles). Refer to the text for details.

the literature, assume that the physical system is described by means of a set of *dependent coordinates* $\mathbf{q}$, which in turn can be determined from a minimum number of *independent coordinates* $\mathbf{z}$ [2]. Thus, the first set of variables comprises $\mathbf{q}_t$, $\dot{\mathbf{q}}_t$ and $\ddot{\mathbf{q}}_t$, corresponding to the position, velocity and acceleration vectors of the mechanism for a discrete time step $t = \{1, 2, ...\}$, plus their counterparts in independent coordinates $\mathbf{z}_t$, $\dot{\mathbf{z}}_t$ and $\ddot{\mathbf{z}}_t$. All sensor readings for one time step t will be stacked together into a single vector $\mathbf{o}_t$. Our statistical model makes no assumptions on the number or kind of sensors employed. Finally, forces will be represented as a vector of generalized forces $\mathbf{Q}_t$ of identical length than the configuration $\mathbf{q}_t$.

One of the main contributions of this work is the proposal of the DBN model shown in Figure 2 as a plausible statistical model for all those variables. As in any dynamic model, variables are connected in a specific pattern that repeats over and over again for each time step. Notice that we decided to cluster the independent configuration and velocity vectors together into an auxiliary variable



(a) Branch $B = 1$

(b) Branch $B = 2$

Fig. 3: Given a four-bar linkage with its degree of freedom denoted as $\mathbf{z}$ (specified via the angle highlighted in the figure), there exist two possible sets of dependent (natural) coordinates $\mathbf{q} = [q_1\ q_2\ q_3\ q_4]^{\mathrm{T}}$ for each value of $\mathbf{z} = [z]$. Figures (a)–(b) illustrate the two possibilities, which we model as different values of a discrete branch variable $B = \{1, 2\}$.

$\mathbf{x}_t^{\mathrm{T}} = \{\mathbf{z}_t^{\mathrm{T}}, \dot{\mathbf{z}}_t^{\mathrm{T}}\}$, leaving the acceleration $\ddot{\mathbf{z}}_t$ as an independent node. Our motivation is twofold: firstly, $\mathbf{z}_t$ and $\dot{\mathbf{z}}_t$ always appear strongly coupled in the equations of motion thus joining them into one node significantly reduces the number of edges in the graph without loosing any information, i.e. no conditional independence will be missed. Secondly, the equations of motion of MBD themselves reveal that accelerations $\ddot{\mathbf{z}}_t$ are functions of the external forces $\mathbf{Q}_t$ and the current state $\mathbf{x}_t^{\mathrm{T}} = \{\mathbf{z}_t^{\mathrm{T}}, \dot{\mathbf{z}}_t^{\mathrm{T}}\}$, but are independent of all previous states and forces. Or, put in probabilistic terms, $\ddot{\mathbf{z}}_t$ is conditionally independent of all present and future variables excepting the current forces $\mathbf{Q}_t$ and state $\mathbf{x}_t$. For this important information to be encoded in the DBN we need to define an independent node for accelerations.

A word is in order about how our model deals with mechanisms with more than one possible branch. Notice how, in Figure 2, the set of dependent coordinates $\mathbf{q}$ is a function of independent coordinates $\mathbf{z}$ but also of an additional *discrete* variable B (named after "branch"). This latter variable models the additional information that is required to unequivocally map independent into dependent coordinates, as illustrated with a four-bar mechanism in the example of Figure 3. Although the mechanism configuration is known in advance for many practical cases, we will assume in the following, for the sake of generality, that it is also an unknown. Therefore, it becomes the responsibility of the probabilistic estimator to distinguish between the different possibilities. To the best of our knowledge, no other previous work has attempted the formulation of such a seamless estimation process for MBS. Naturally, problems where the assembly configuration is perfectly known in advance can leave the variable B out of the estimator's vector of unknowns.

Since changing the active branch B during operation is an unrealistic event in practice, we will assume in the following that B is constant during the entire operation, although the DBN can be straightforwardly modified to allow a dynamic B if needed.

Other remarkable points in the model are observations and transitions between time steps. Sensor observations $\mathbf{o}_t$ have been modeled as depending only on the current mechanism position, velocity and acceleration; note how this simple model

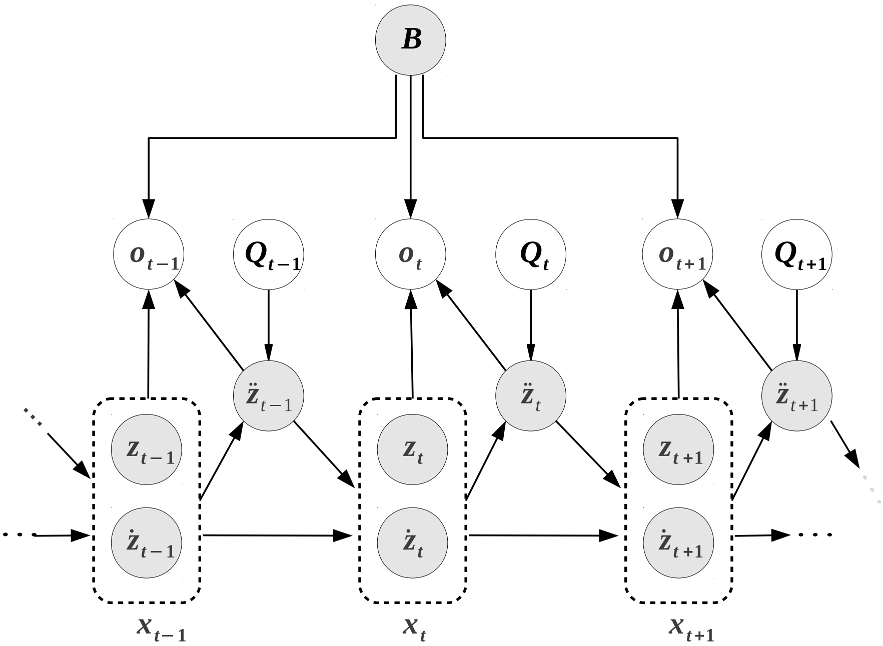


Fig. 4: The DBN in Figure 2 after marginalizing out all dependent coordinates. This is the final model employed in the rest of the work.

includes most common sensors such as encoders, resolvers, accelerometers, gyroscopes, etc. Regarding the transitions between consecutive time steps ($t \to t+1$), they simply correspond to the application of a numerical integrator.

Thinking of the process of statistical inference at which the DBN model is aimed we have already colored the nodes in Figure 2 accordingly: forces ($\mathbf{Q}_t$) and sensor readings ($\mathbf{o}_t$) are *observed* (known) variables, while the mechanism dynamical state and its branch (B) are represented as shaded nodes because they are *hidden* nodes to be estimated, as addressed in section 3. Note however that the same model may be employed to identify part of the acting forces or some poorly modeled parameters of the mechanism, by just applying a different partitioning of nodes into known and hidden.

Before deriving the filtering equations it becomes convenient to apply one simplification to the graphical model in Figure 2. Recall that edges in a DBN encode *probabilistic* relationships between sets of variables. Still, in the particular case of MBD we find that dependent coordinates $\mathbf{q}$ are *deterministic* functions of the independent ones $\mathbf{z}$ and the branch B, thus in an estimation problem it makes sense to focus on recovering only the independent coordinates and the configuration. When dependent coordinates $\mathbf{q}$ are required (for instance, while predicting the outcome of an inertial sensor attached to the mechanism) they could be trivially computed from $\mathbf{z}$ and B. Removing some of the random variables from a probabilistic model is called *marginalization* and depending on the specific application the process may have pros and cons. In our case, marginalizing out the dependent variables is desirable because of the reduction in the dimensionality of the estimation problem.

This idea is closely related to the *Rao-Blackwellization* technique [28], well-known in statistical inference, where part of a target probability distribution is computed analytically (in closed-form) as a function of the other part which is estimated with a filter.

The reduced DBN after marginalizing dependent coordinates is shown in Figure 4. Our motivation for introducing the branch variable B now becomes clear under the statistical perspective: if dependent coordinates $\mathbf{q}_t$ were formulated as depending on the previous set $\mathbf{q}_{t-1}$ as is common in the MBD literature, more edges would appear after marginalizing them out[2], leading to an impractical, barely sparse graph. Instead, the proposed graphical model remains sparse and presents a repetitive structure over time steps, which eases the derivation of filters or any other kind of estimators.

Another argument favoring the introduction of the branch variable B arises if one reflects about the fact that the proposed graphical model aims at representing the underlying structure of an estimation problem in multibody dynamics. One should prefer the *smallest* theoretical model among those that fit a large number of problems in order to reduce the overall number of unknowns. *Smallest* above means having the minimum number of variables and relationships between them. Therefore, it makes sense to get rid of the coupling, for each time step t, between all dependent coordinates $\mathbf{q}_t$ and their predecessors $\mathbf{q}_{t-1}$, a coupling that only seems natural due to the customary approach of deriving $\mathbf{q}_t$ from independent coordinates $\mathbf{z}_t$ and the *hint* of the nearby configuration $\mathbf{q}_{t-1}$. However, from a conceptual point of view the introduction of the branch variable B involves fewer constraints, fewer variables, and also opens the possibility of employing techniques capable of automatically determining the discrete branches of the constraint manifold [29].

Finally, another incentive to marginalize out dependent coordinates is that most estimators assume a state space of independent variables.

3 Recursive Bayesian filtering

Now that the variables involved in the dynamic simulation of a multibody system have been modeled as a DBN we can derive a recursive equation for Bayesian filtering of such a system.

Our goal is to perform statistical inference to estimate the *posterior* density distribution of the hidden variables (the branch B and the dynamic state $\mathbf{x}_t = \{\mathbf{z}_t, \dot{\mathbf{z}}_t\}$ and $\ddot{\mathbf{z}}_t$) at some discrete time step t given the sequence of all sensory data $\mathbf{o}_{1:t} = \{\mathbf{o}_1, ..., \mathbf{o}_t\}$ and known forces $\mathbf{Q}_{1:t} = \{\mathbf{Q}_1, ..., \mathbf{Q}_t\}$, that is, the PDF:

$$\underbrace{p\left(\mathbf{x}_t, \ddot{\mathbf{z}}_t, B | \mathbf{Q}_{1:t}, \mathbf{o}_{1:t}\right)}_{\text{Posterior for } t} \tag{2}$$

Applying the definition of conditional probability[3] we have:

[2] The rule for removing variables from a DBN determines that new edges must be created between the neighbors of removed nodes.

[3] From $p(a|b) = p(a,b)/p(b)$ it follows that $p(a,b) = p(a|b)p(b)$, which can be generalized to $p(a,b|c) = p(a|b,c)p(b|c)$ via, for example, the multiplication rule for $p(a,b,c)$.

$$p\left(\mathbf{x}_t, \ddot{\mathbf{z}}_t, B | \mathbf{Q}_{1:t}, \mathbf{o}_{1:t}\right) = \underbrace{p\left(\ddot{\mathbf{z}}_t | \mathbf{x}_t, B, \mathbf{Q}_{1:t}, \mathbf{o}_{1:t}\right)}_{T_1} \underbrace{p\left(\mathbf{x}_t, B | \mathbf{Q}_{1:t}, \mathbf{o}_{1:t}\right)}_{T_2} \tag{3}$$

Applying the Bayes rule[4] to the first term (T_1) over the latest observation $\mathbf{o}_t$ one obtains:

$$T_1 \propto p\left(\mathbf{o}_t | \ddot{\mathbf{z}}_t, \mathbf{x}_t, B, \mathbf{Q}_{1:t}, \mathbf{o}_{1:t-1}\right) p\left(\ddot{\mathbf{z}}_t | \mathbf{x}_t, B, \mathbf{Q}_{1:t}, \mathbf{o}_{1:t-1}\right) \tag{4}$$

where the proportionality constant does not affect the estimation and can be ignored. Now, the information about conditional independence can be exploited to simplify the expression above:

$$T_1 \propto \underbrace{p\left(\mathbf{o}_t | \ddot{\mathbf{z}}_t, \mathbf{x}_t, B\right)}_{\text{Observation likelihood}} \quad \underbrace{p\left(\ddot{\mathbf{z}}_t | \mathbf{x}_t, \mathbf{Q}_t\right)}_{\text{Prob. equation of motion}} \tag{5}$$

Indeed, a careful observation of the DBN in Figure 4 reveals the following conditional independence:

$$\begin{aligned} p\left(\mathbf{o}_t | \ddot{\mathbf{z}}_t, \mathbf{x}_t, B, \mathbf{Q}_{1:t}, \mathbf{o}_{1:t-1}\right) &\overbrace{=}^{\mathbf{o}_t \perp\!\!\!\perp \mathbf{Q}_{1:t}, \mathbf{o}_{1:t-1} | \mathbf{x}_t, \ddot{\mathbf{z}}_t, B} p\left(\mathbf{o}_t | \ddot{\mathbf{z}}_t, \mathbf{x}_t, B\right) \\ p\left(\ddot{\mathbf{z}}_t | \mathbf{x}_t, B, \mathbf{Q}_{1:t}, \mathbf{o}_{1:t-1}\right) &\underbrace{=}_{\ddot{\mathbf{z}}_t \perp\!\!\!\perp \mathbf{o}_{1:t-1}, \mathbf{Q}_{1:t-1}, B | \mathbf{x}_t, \mathbf{Q}_t} p\left(\ddot{\mathbf{z}}_t | \mathbf{x}_t, \mathbf{Q}_t\right) \end{aligned}$$

We must remark the importance of the term denoted as *observation likelihood* in Eq. (5): this is the unique point at which data from sensors are contrasted to predictions from the estimator, making possible filter corrections as the actual movement of the mechanism is tracked. In the case of multihypothesis estimators (as the proposal in section 5) the term also allows the discrimination among the different branches B, assigning a rigorously well-grounded probability to each possibility. Regarding the other term of T_1 in Eq. (5), $p\left(\ddot{\mathbf{z}}_t | \mathbf{x}_t, \mathbf{Q}_t\right)$, it can be interpreted as a *probabilistic equation of motion*. Section 4 provides further details on how to implement these two probabilistic models.

Next, we address the term $T_2 = p\left(\mathbf{x}_t, B | \mathbf{Q}_{1:t}, \mathbf{o}_{1:t}\right)$, which represents a *marginal distribution* where all past states $\mathbf{x}_{1:t-1}$ and all accelerations $\ddot{\mathbf{z}}_{1:t}$ have been filtered out. However, it is clear from the DBN in Figure 4 that any present state $\mathbf{x}_t$ depends on the previous dynamical state $\mathbf{x}_{t-1}$ and $\ddot{\mathbf{z}}_{t-1}$, thus they should appear somehow in the expression for $\mathbf{x}_t$. Here we invoke the law of total probability[5] to

[4] The Bayes rule establishes how the observation of b affects the prior knowledge regarding another variable $p(a)$. It says that $p(a|b) \propto p(b|a)p(a)$, with $p(b|a)$ customarily called the *likelihood* of b. In the text, we apply the rule extension for conditional probabilities, where all terms can be conditioned to another variable or variable set c, i.e. $p(a|b,c) \propto p(b|a,c)p(a|c)$.

[5] This law states that any marginal distribution $p(a)$ can be expressed as the summation or integration (for discrete or continuous domains, respectively) of the conditional $p(a|b)$ for all possible values of b, i.e. $p(a) = \int_{-\infty}^{\infty} p(a|b)p(b)db$. It can be trivially extended for conditioned probabilities, as required in our derivation, i.e. $p(a|c) = \int_{-\infty}^{\infty} p(a|b,c)p(b|c)db$. For discrete variables, as the case of the branch B, integrals are replaced by summations over all the potential values.

introduce those two terms. For convenience we also introduce B as a conditioning variable, leading to:

$$\begin{aligned} T_2 &= p\left(\mathbf{x}_t, B|\mathbf{Q}_{1:t}, \mathbf{o}_{1:t}\right) \\ &= \sum_B \int\int_{-\infty}^{\infty} \underbrace{p\left(\mathbf{x}_t, \cancel{B}|\mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1}, \cancel{B}, \mathbf{Q}_{1:t}, \mathbf{o}_{1:t}\right)}_{\text{Since: } p(a,b|b)=p(a,\cancel{b}|\cancel{b})=p(a)} p\left(\mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1}, B|\mathbf{Q}_{1:t}, \mathbf{o}_{1:t}\right) d\mathbf{x}_{t-1} d\ddot{\mathbf{z}}_{t-1} \\ &= \sum_B \int\int_{-\infty}^{\infty} p\left(\mathbf{x}_t|\mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1}\right) p\left(\mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1}, B|\mathbf{Q}_{1:t-1}, \mathbf{o}_{1:t-1}\right) d\mathbf{x}_{t-1} d\ddot{\mathbf{z}}_{t-1} \end{aligned} \tag{6}$$

where the following conditional independences from the DBN model have been employed during the last step to drop non-informative terms:

$$\begin{array}{ccc} p\left(\mathbf{x}_t|\mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1}, \mathbf{Q}_{1:t}, \mathbf{o}_{1:t}\right) & \overbrace{=}^{\mathbf{x}_t \perp\!\!\!\perp \mathbf{Q}_{1:t}, \mathbf{o}_{1:t} | \mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1}} & p\left(\mathbf{x}_t|\mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1}\right) \\ p\left(\mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1}, B|\mathbf{Q}_{1:t}, \mathbf{o}_{1:t}\right) & \underbrace{=}_{\mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1}, B \perp\!\!\!\perp \mathbf{Q}_{1:t-1}, \mathbf{o}_{1:t-1} | \mathbf{Q}_t, \mathbf{o}_t} & p\left(\mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1}, B|\mathbf{Q}_{1:t-1}, \mathbf{o}_{1:t-1}\right) \end{array}$$

Putting together Eqs. (3)–(6) we obtain the sought recursive expression:

$$\begin{aligned} &\overbrace{p\left(\mathbf{x}_t, \ddot{\mathbf{z}}_t, B|\mathbf{Q}_{1:t}, \mathbf{o}_{1:t}\right)}^{\text{Posterior for } t} \propto \overbrace{p\left(\mathbf{o}_t|\ddot{\mathbf{z}}_t, \mathbf{x}_t, B\right)}^{\text{Observation likelihood}} \times \overbrace{p\left(\ddot{\mathbf{z}}_t|\mathbf{x}_t, \mathbf{Q}_t\right)}^{\text{Prob. equation of motion}} \times \\ &\times \sum_B \int\int_{-\infty}^{\infty} \underbrace{p\left(\mathbf{x}_t|\mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1}\right)}_{\text{Numerical integration}} \underbrace{p\left(\mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1}, B|\mathbf{Q}_{1:t-1}, \mathbf{o}_{1:t-1}\right)}_{\text{Posterior for } t-1} d\mathbf{x}_{t-1} d\ddot{\mathbf{z}}_{t-1} \end{aligned} \tag{7}$$

which takes as input the posterior PDF at $t-1$ and allows the estimation of the updated one for t. Hence the *recursive* nature of this kind of Bayesian estimator. It is worth pointing out that only the latest dynamical state is estimated at each instant of time with the present formulation, as corresponds to our decision of approaching the estimation problem from a *filtering* perspective. The DBN model devised in this work can be also employed to derive other inferring schemes such as batch estimators and smoothers [30]. However, they have been left out of the present work for the sake of conciseness.

At this point we already have the recursive probabilistic equation Eq. (7) but still have to decide how to implement it in practice. Two fundamentally different kinds of approaches exist in the literature: those based on parametric [31, 32] and non-parametric distributions [28,33]. The formers are basically dominated by methods that assume multivariate Gaussian distributions for all the involved variables, a convenient assumption that allows modeling all filtered distributions as analytics functions with a few parameters (i.e. a mean vector and a covariance matrix). This is the approach already proposed in the literature for state observers based on multibody models [34]. In order to illustrate that other alternatives do exist, this paper will implement the filtering equation by making use of non-parametric models, leading to the *particle filter* algorithm that will be introduced in section 5.

4 Probabilistic models for MBD

Dealing with uncertainty and probability distributions is not common in the multibody literature, thus we deserve this section to giving a detailed description of how to merge MBD concepts with probability such that we are able to define all the terms involved in Eq. (7).

4.1 Observation likelihood

The first probabilistic term that appears in the recursive filtering equation is $p\left(\mathbf{o}_t|\ddot{\mathbf{z}}_t, \mathbf{x}_t, B\right)$. Let us denote as L to the number of sensors installed on the mechanism, which may be of any kind as long as their readings $\mathbf{o}_t$ can be modeled according to the very general form:

$$\mathbf{o}_t = \begin{bmatrix} o_t^1 \\ \vdots \\ o_t^L \end{bmatrix} = \begin{bmatrix} h^1(\ddot{\mathbf{z}}_t, \mathbf{x}_t, B) + n_t^1 \\ \vdots \\ h^L(\ddot{\mathbf{z}}_t, \mathbf{x}_t, B) + n_t^L \end{bmatrix} = \mathbf{h}(\ddot{\mathbf{z}}_t, \mathbf{x}_t, B) + \mathbf{n}_t \tag{8}$$

where $\mathbf{h}(\cdot)$ is the arbitrary model that predicts the sensor readings from a known dynamic state $(\ddot{\mathbf{z}}_t, \mathbf{x}_t)$ and branch (B), and $\mathbf{n}_t$ is additive zero-mean Gaussian noise that models measuring errors, a wrong calibration, electrical noise, etc. In general, the noise from each sensor will be independent (i.e. uncorrelated to the other errors), but for the sake of generality we assume that this noise follows a general multivariate Gaussian distribution with covariance matrix $\boldsymbol{\Sigma}_s$, that is:

$$\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_s) \tag{9}$$

where $x \sim p(x)$ means "x follows the $p(x)$ probability distribution" and $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

Using this information to derive the expression for the observation likelihood requires some knowledge that will not be evident until section 5, namely whether the values $\ddot{\mathbf{z}}_t$, $\mathbf{x}_t$ and B also have associated uncertainty or not. As will be shown, in the particular estimator introduced in this paper all three variables can be assumed to be perfectly known, since they will be hypotheses within a particle filter. Therefore, all the arguments of the sensor model $\mathbf{h}(\cdot)$ are perfectly known and do not add additional uncertainty to the sensor reading predictions, leaving the likelihood as follows:

$$\begin{aligned} &p\left(\mathbf{o}_t|\ddot{\mathbf{z}}_t, \mathbf{x}_t, B\right) = \mathcal{N}(\hat{\mathbf{o}}_t, \boldsymbol{\Sigma}_s) \\ &\text{with:} \qquad \hat{\mathbf{o}}_t = \mathbf{h}(\ddot{\mathbf{z}}_t, \mathbf{x}_t, B) \end{aligned} \tag{10}$$

4.2 Probabilistic equations of motion

From the variety of different formulations that exist for solving the dynamic equations of motion of a rigid MBS [6], this work assumes the usage of any such formulation that works with *independent coordinates* [2]. Moreover, our implementation makes use of natural coordinates but nothing in the probabilistic estimators would change if another decision was made.

For a sequential estimator, the equations of motion behave like a black box that gives the (independent) accelerations for each kinematic and dynamic situation:

$$\ddot{\mathbf{z}}_t = \mathbf{f}(\underbrace{\mathbf{z}_t, \dot{\mathbf{z}}_t}_{\mathbf{x}_t}, \mathbf{Q}_t) + \mathbf{v}_t \tag{11}$$

where $\mathbf{v}_t$ is a zero-mean additive Gaussian noise with covariance matrix $\boldsymbol{\Sigma}_a$ that models errors in the physical parameters of the mechanism, the effects of non-modeled friction, unexpected internal or external forces, etc. Please note that the Gaussianity assumption of acceleration errors applies pointwise at any possible state $\mathbf{x}_t$. The uncertainty of such state space is by no means constrained to be modeled in any particular form. In particular, section 5 proposes a non-parametric representation.

As an example of what to expect as a function $\mathbf{f}(\cdot)$, in the case of employing the Matrix $\mathbf{R}$ method [2, §5.2.3] it looks like:

$$\mathbf{f}(\mathbf{z}_t, \dot{\mathbf{z}}_t, \mathbf{Q}_t) = \left(\mathbf{R}^{\mathrm{T}}\mathbf{M}\mathbf{R}\right)^{-1} \mathbf{R}^{\mathrm{T}} \left(\mathbf{Q}_t - \mathbf{M}\mathbf{S}\mathbf{c}\right) \tag{12}$$

where $\mathbf{M}$ is the system mass matrix, $\mathbf{R}$ is a projection matrix such that $\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{z}}$ and the product $\mathbf{Sc}$ equals $\ddot{\mathbf{q}}$ when $\ddot{\mathbf{z}} = \mathbf{0}$ (refer to [2] for further details).

Since neither the state $\mathbf{x}_t$ nor the generalized forces $\mathbf{Q}_t$ have uncertainty in the formulation of estimators as particle filters because each particle carries its own hypothesis about the state, as discussed in section 5, the only source of uncertainty in accelerations is the above mentioned additive noise, which gives raise to the following simple expression for the sought distribution:

$$\begin{aligned} p\left(\ddot{\mathbf{z}}_t|\mathbf{x}_t, \mathbf{Q}_t\right) &= \mathcal{N}(\hat{\ddot{\mathbf{z}}}_t, \boldsymbol{\Sigma}_a) \\ \text{with:} \qquad \hat{\ddot{\mathbf{z}}}_t &= \mathbf{f}(\mathbf{x}_t, \mathbf{Q}_t) \end{aligned} \tag{13}$$

4.3 Numerical integration

In the statistics literature, terms like $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1}\right)$ are interpreted as *transition models* because they define how a variable ($\mathbf{x}$) evolves over time, possibly affected by some external influence ($\ddot{\mathbf{z}}$). Particularizing for MBD, it becomes clear that the term describes the process of numerical integration (n.i.) of independent coordinates that takes place at each time step. Again, the particular form of this probabilistic term depends on whether or not the involved variables have associated uncertainty. In the particular case of using a particle filter (described in next section), all variables are perfectly known, thus we arrive at a simple deterministic model where the transition becomes a Dirac delta:

$$\begin{aligned} p\left(\mathbf{x}_t|\mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1}\right) &= \delta(\mathbf{x}_t - \hat{\mathbf{x}}_t) \\ \text{with:} \qquad & \hat{\mathbf{x}}_t \xleftarrow{\text{n.i.}} \mathbf{x}_{t-1}, \ddot{\mathbf{z}}_{t-1} \end{aligned} \tag{14}$$

which imposes the usage of an *explicit* integrator –we used an explicit fourth-order Runge-Kutta method in our particle filter implementation. It must be noted that *implicit* integrators can also be used with our DBN model, as long as iterative estimators (e.g. iterated EKF) are employed.

5 Online estimation with sequential MonteCarlo

Starting from the graphical model, section 3 showed how to obtain the recursive equations of a probabilistic MBD estimator in a generic mathematical form. Next we address the implementation of such equation into a feasible computer program suitable for practical problems.

5.1 A particle filter for MBD

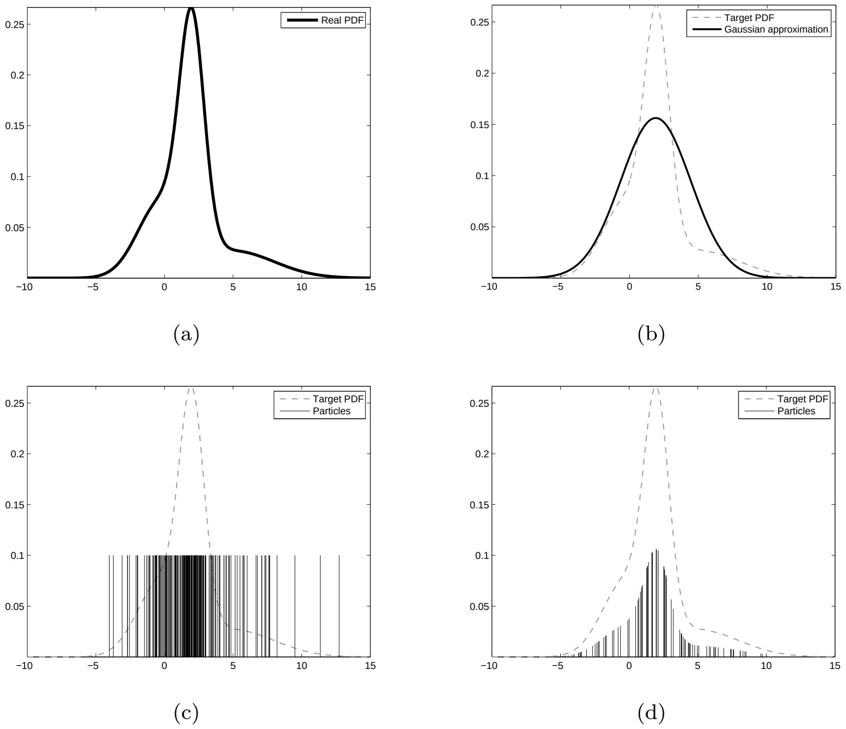
Implementing the recursive formula in Eq. (7) requires making a decision about how to model each probability distribution in the equation. One possibility is assuming *parametric* probability distributions, such as multivariate Gaussians or sums of Gaussians. Under this convenient choice, the integrals of products of distributions that appear in the equation also turn out to be Gaussians, thus it is only needed to store a few parameters (i.e. the mean and covariance) for the distributions at each time step. Such an approach leads to the family of Kalman-like filters, which have the advantage of being computationally efficient. However, they are unable to automatically distinguish between the different ”branches” of a mechanism, neither are applicable when the initial state is unknown, due to their limited support to handle nonlinearities.

To solve these drawbacks, we decided to employ *non-parametric* distributions in this work. In particular, a technique called *importance sampling* [35]. If the distribution that is modeled with samples is employed to perform sequential Bayesian estimation, the resulting method receives different names depending on the community, being the most common ones *sequential MonteCarlo* (SMC) and *particle filter* (PF). Next we introduce the basis of this approach, but the interested reader can refer to the introduction in [36] or to more exhaustive reviews of theoretical advances in the field [28,33,37].

Importance sampling consists of approximating probability distributions with finite sets of weighted hypotheses, also called *particles.* Informally speaking, the goal is approximating the target distribution with the weighted *density* of particles at each point of the state space. Under ideal conditions all weights tend to be identical, such that all the samples are equally significant, meaning that the particles can be considered to be samples drawn from some target distribution $p(x)$:

$$x^{[i]} \sim p(x) \qquad , \text{for } i = \{1, ..., M\} \tag{15}$$

where $x^{[i]}$ are the M particles. In general, each particle is also associated a relative weight $w^{[i]}$ and then it follows that the PDF can be approximated as a weighted combination of Dirac deltas $\delta(\cdot)$ at each particle position:



(a) (b) (c) (d)

Fig. 5: An illustration of the principles underlying importance sampling for a one-dimensional state space, here represented as the horizontal axis. (a) The target probability density function (PDF). (b) A Gaussian approximation of the PDF. (c)–(d) Two different approximations with particles, where all have exactly the same weight (c) or different weights (d). Note that the particles themselves are the position at the state space axis (represented as solid segments), while their weights are represented as the height of each segment.

$$p(x) \approx \sum_{i=1}^{M} w^{[i]} \delta(x - x^{[i]}) \tag{16}$$

It is worth making a brief stop at this point in order to gain an insight about how importance sampling really works in comparison to parametric methods. Consider some target PDF, which may have an arbitrary shape, such as the one shown in Figure 5(a). In general, the shape of a PDF $p(x)$ resulting from probabilistic filtering of non-linear systems is too complex to be described in closed form, hence our need to rely on approximations. One alternative is approximating it as a Gaussian distribution, which only requires providing two parameters: a mean and a variance, a situation illustrated in Figure 5(b). Alternatively, the same PDF can be approximated with importance sampling. One such approximations is that shown in Figure 5(c), where we find the ideal situations of all particles having exactly the

Algorithm 1 MBD_SIR $\{\mathbf{s}_t^{[i]}, w_t^{[i]}\}_{i=1}^{M} \rightarrow \{\mathbf{s}_{t+1}^{[i]}, w_{t+1}^{[i]}\}_{i=1}^{M}$

```
for all particles $\mathbf{s}_t^{[i]} = \{\mathbf{x}_t^{[i]}, \ddot{\mathbf{z}}_t^{[i]}, B^{[i]}\}$ do // For each $i = 1...M$
    $\mathbf{x}_{t+1}^{[i]} \xleftarrow{\text{n.i.}} \mathbf{x}_t^{[i]}, \ddot{\mathbf{z}}_t^{[i]}$ // Numerical integration, see Eq. (14)
    $\ddot{\mathbf{z}}_{t+1}^{[i]} \sim p\left(\ddot{\mathbf{z}}_{t+1} | \mathbf{x}_{t+1}^{[i]}, \mathbf{Q}_{t+1}\right)$ // Draw acceleration sample, see Eq. (13)
    $\mathbf{s}_{t+1}^{[i]} \leftarrow \{\mathbf{x}_{t+1}^{[i]}, \ddot{\mathbf{z}}_{t+1}^{[i]}, B^{[i]}\}$ // Assemble new particle data
    $\hat{w}_{t+1}^{[i]} \leftarrow w_t^{[i]} p\left(\mathbf{o}_t | \ddot{\mathbf{z}}_t^{[i]}, \mathbf{x}_t^{[i]}, B^{[i]}\right)$ // Update weight, see Eq. (10)
end for
// Normalize weights
for all $i = 1...M$ do
    $w_{t+1}^{[i]} = \frac{\hat{w}_{t+1}^{[i]}}{\sum_{k=1}^{M} \hat{w}_{t+1}^{[k]}}$
end for
$\hat{N}_{eff} \leftarrow \left(\sum_{i=1}^{M} \left(w_{t+1}^{[i]}\right)^2\right)^{-1}$ // Evaluate ESS
if $\hat{N}_{eff} < \frac{M}{2}$ then // Selective resampling
    $\{\mathbf{s}_{t+1}^{[i]}, w_{t+1}^{[i]}\}_{i=1}^{M} \leftarrow \text{resample}(\{\mathbf{s}_{t+1}^{[i]}, w_{t+1}^{[i]}\}_{i=1}^{M})$
end if
```

same weight. This means that the "density" of samples exactly tends towards the target PDF as the number of samples tends to infinity. In other words: all particles can be considered to be statistical samples drawn from the target PDF. Note how difficult is to achieve this ideal since, in practical applications, we do not know how to draw samples from an unknown PDF. That is where weights play an important correcting role: as illustrated in Figure 5(d), if we now draw particles following a uniform distribution over the state space and then set their weights according to *pointwise evaluations* of the target PDF, the resulting "weighted density" of samples also converges towards the desired PDF as the number of particles grows. The key insight here is that we are able to model a PDF by only evaluating its value pointwise, even if we lack a closed-form expression for it.

Once the basics of importance sampling have been introduced, we can recover the goal of deriving a filtering algorithm for MBD. Most previous works on particle filters that can be found in the literature assume the following simple filtering equation:

$$p(\mathbf{x}_{t+1}|\mathbf{o}_{1:t+1}) \propto p(\mathbf{o}_t|\mathbf{x}_{t+1}) \int_{-\infty}^{\infty} p(\mathbf{x}_{t+1}|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{o}_{1:t}) d\mathbf{x}_t \tag{17}$$

which does not exactly matches the one derived in the previous section. Therefore, we must directly apply importance sampling principles to Eq. (7) in order to derive the correct algorithm to be applied in the estimation of MBD.

We start by defining M particles $\mathbf{s}_t^{[i]} = \{\mathbf{x}_t^{[i]}, \ddot{\mathbf{z}}_t^{[i]}, B^{[i]}\}$ for $i = \{1, ..., M\}$, with weights $w_t^{[i]}$, as the approximation of the posterior distribution at time step t. Defining these particles for $t = 0$ is known as the filter initialization problem, and is addressed in section 5.2. A particle filtering algorithm aims at computing an updated set of particles $\mathbf{s}_{t+1}^{[i]}$ and weights $w_{t+1}^{[i]}$ for $t + 1$ that fulfills the recursive equation Eq. (7). Implementing that Bayesian equation is enough for obtaining a working estimator, known as the *Sequential Importance Sampling* (SIS) algorithm [35]. However, it has been demonstrated that the variance of the weights increases over time [38], eventually leading to the degeneracy of the filter. That is the reason why we adopt the *SIS with resampling* (SIR) algorithm [39], where an additional *resampling* step is introduced in order to replace particles with low weights with copies of more likely particles. Different resampling schemes are possible [40], but for MBD they all have similar performance.

To help with the introduction of the resulting method, a pseudocode description is provided in Algorithm 1, whose line numbers are referenced in the following. The algorithm takes as input a set of particles and weights for one time step t and returns the updated sets for the $t+1$. The three parts of the SIR algorithm are clearly identified: (i) Implementation of Bayesian filtering (lines 1–6), (ii) normalization of weights such as they sum up the unity (lines 7–10) and (iii) selective resampling if the effective sample size (ESS), denoted as $\hat{N}_{eff}$, drops below a given threshold [41] (lines 11–14). The first part is the most complex one, thus we next provide a more detailed description. From the three terms in the product at the right hand side of Eq. (7), the implementation firstly evaluates the third term in the product (the sum of integrals). Since the posterior distribution for t is represented as a set of particles, hence as a sum of Dirac deltas as in Eq. (16), the sums and integrals in Eq. (7) simplify into the application of the numerical integrator to each particle (see line 2). Let us write down this step for the sake of clarity:

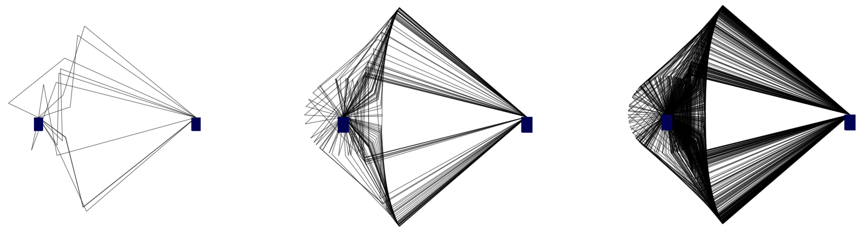
$$\sum_B \int\int_{-\infty}^{\infty} \underbrace{p\left(\mathbf{x}_{t+1}|\mathbf{x}_t,\ddot{\mathbf{z}}_t\right)}_{\text{Numerical integration}} \underbrace{p\left(\mathbf{x}_t,\ddot{\mathbf{z}}_t,B|\mathbf{Q}_{1:t},\mathbf{o}_{1:t}\right)}_{\text{Posterior for } t} d\mathbf{x}_t d\ddot{\mathbf{z}}_t =$$

$$\sum_B \int\int_{-\infty}^{\infty} \underbrace{\delta(\mathbf{x}_{t+1} - \overbrace{\hat{\mathbf{x}}_{t+1}(\mathbf{x}_t,\ddot{\mathbf{z}}_t)}^{\mathbf{x}_{t+1}^{[i]}})}_{\text{n.i. as of Eq. (14)}} \underbrace{\sum_{i=1}^{M} \omega_t^{[i]} \delta\left(\{\mathbf{x}_t,\ddot{\mathbf{z}}_t,B\} - \left\{\mathbf{x}_t^{[i]},\ddot{\mathbf{z}}_t^{[i]},B^{[i]}\right\}\right)}_{\text{Importance sampling model}} d\mathbf{x}_t d\ddot{\mathbf{z}}_t =$$

$$\sum_{i=1}^{M} \omega_t^{[i]} \delta\left(\{\mathbf{x}_{t+1},\ddot{\mathbf{z}}_t,B\} - \left\{\mathbf{x}_{t+1}^{[i]},\ddot{\mathbf{z}}_t^{[i]},B^{[i]}\right\}\right) \tag{18}$$

Next, the probabilistic equations of motion, as discussed in section 4.2, simply become the solution of the standard MBD problem with the addition of random noise to the obtained accelerations (line 3). Mathematically, this imply approximating the product of Eq. (18) and the equations of motion $p\left(\ddot{\mathbf{z}}_{t+1}|\mathbf{x}_{t+1},\mathbf{Q}_{t+1}\right)$. Using Eq. (13) it is clear that the resulting distribution becomes a sum of Gaussians which, following the SIR algorithm [39], is approximated as a set of samples by drawing one random sample from each Gaussian. Given that the branch B does not change over time, it remains unmodified, thus all the particle information for the new time step is ready at that point (line 4). Finally, the remaining term in the recursive Bayesian product in Eq. (7), that is, the observation likelihood, turns out to be a well-determined scalar for each particle i, thus it can be absorbed into the posterior as a modification of the particle weight (see line 5).

It must be noted that the described method is just one of the possible ways of implementing the Bayesian equation as a particle filter, but it is the simplest one and therefore the most suitable for real-time estimation of mechanisms. Fu-



(a) 10 particles (b) 100 particles (c) 500 particles

Fig. 6: Simultaneous representation of all initial hypotheses for an increasing number of particles aimed at estimating the dynamic state of a four-bar linkage similar to that shown in Figure 3. The initial distribution follows a uniform distribution over the joint space of independent coordinates and branches.

ture works may explore other alternatives such as the usage of optimal proposal distributions [38] which would allow accounting for implicit integrators.

Finally, we must address the issue of recovering the filter *most-likely* estimation at any instant of time, which may be required in real-time to, for example, feed a controller whose actuation depends on the mechanism state. In those cases, we can easily evaluate the *expected value* of the posterior density function as:

$$\left[\hat{\mathbf{z}}_t, \hat{\dot{\mathbf{z}}}_t, \hat{\ddot{\mathbf{z}}}_t\right] = \sum_{i=1}^{M} w_t^{[i]} \left[\mathbf{z}_t^{[i]}, \dot{\mathbf{z}}_t^{[i]}, \ddot{\mathbf{z}}_t^{[i]}\right] \tag{19}$$

Caution must be paid to the formula for $\hat{\mathbf{z}}_t$ above when there exist independent coordinates that model angles instead of Euclidean coordinates, since averaging over the manifold of planar orientations requires a modified averaging equation[6]. In turn, the angular velocities and accelerations corresponding to those coordinates do not present such a complication.

Regarding the most-likely branch B of the mechanism at any time step t, it does not make sense to evaluate an "average value" due to the discrete nature of that variable. Therefore, all we can do is computing the *probability* of the mechanism to be at each branch b_k out of the $|B|$ possible ones, which amounts to summing the weights of all the particles that carry such a hypothesis:

$$P(B = b_k) = \sum_{\forall i / B_t^{[i]} = c_k} w_t^{[i]} \qquad \text{, for } k = 1..., |B| \tag{20}$$

A practical application of this formula will be demonstrated in Figure 11(c)–(d) when determining the assembly of a real mechanism from sensor readings.

[6] For planar rotations in $\mathbf{SO}(2)$, weighted averaging can be performed by weighting the two-dimensional coordinates of points at each given angle over the unit circle, then projecting the average point back to the unit circle. For $\mathbf{SO}(3)$ rotations, several averaging methods exist depending on the desired matrix metric over the manifold [42].

5.2 Filter initialization

All filtering estimators consist of a recursive equation that propagates the state belief from one time step to the next one. The important issue of how to set the initial belief of the filter remains as an open problem that must be solved by the analyst depending on the particularities of the filter, the mechanism, the sensors and the prior knowledge available at start up. Parametric filters such as those based on EKF [43] or the Unscented Kalman Filter (UKF) [32] impose a serious limitation in this sense, since, in general, the initial state of the mechanism must be roughly known in advance in order to be well approximated as a Gaussian, restricting their practical utility to well-controlled scenarios.

In turn, one of the clearest advantages of particle filters is their suitability to deal with ambiguous situations, such that the lack of information about the mechanism state when the state estimator starts up. As an example, Figure 6 illustrates the initial state of our proposed particle filter for a different number of particles when used to track a four-bar linkage whose initial position is totally unknown. Mathematically, each component of the initial set of particles for $t = 0$, that is,

$$\mathbf{s}_0^{[i]} = \{\underbrace{\mathbf{z}_0^{[i]}, \dot{\mathbf{z}}_0^{[i]}}_{\mathbf{x}_0^{[i]}}, \ddot{\mathbf{z}}_0^{[i]}, B\} \tag{21}$$

must be generated as a random sample from the corresponding *uniform distribution* according to the physical limits of the variables used to describe the mechanism. Particularizing for a four bar mechanism, which has two possible branches ($B = 1$ or $B = 2$) and only one degree of freedom, and assuming that the independent variable is the angular position of the crank (θ_t), we have:

$$\begin{cases} \theta_0^{[i]} \sim \mathcal{U}(-\pi, \pi) \\ \dot{\theta}_0^{[i]} \sim \mathcal{U}(-w_{max}, w_{max}) \\ \ddot{\theta}_0^{[i]} \sim \mathcal{U}(-\dot{w}_{max}, \dot{w}_{max}) \\ B_0^{[i]} \sim \mathcal{U}(\{1, 2\}) \end{cases} \tag{22}$$

where $\mathcal{U}(\cdot)$ stands for the uniform distribution[7], and w_{max} and $\dot{w}_{max}$ are upper limits for feasible initial angular velocities and accelerations, respectively. Clearly, using initial distributions for mechanisms with several degrees of freedom would require an excessive number of particles to sufficiently cover the entire state space, rendering impossible their application to real-time operation. However, we demonstrate with the experiments below that particle filters are a suitable, robust solution for simple-enough mechanisms.

[7] The continuous uniform distribution $p(x) = \mathcal{U}(a, b)$ assigns the same probability density to every point $x \in (a, b)$, whereas the discrete version $P(x) = \mathcal{U}(S)$ models the case where each event in the set S has exactly the same probability of occurrence.

6 Experiments

In order to evaluate the suitability, robustness and accuracy of the proposed PF algorithm for multibody tracking we performed two series of experiments: a first set of tests on a simulated mechanism and real tests on a physical prototype. In both cases the employed model is a four-bar linkage. Next we address the two sets of experiments. We also recommend viewing the online video demonstrating the results regarding particle filter convergence under large initial uncertainty[8]. The C++ source code for our experiments is also publicly available online[9].

6.1 Simulations

The aim of these first experiments is to demonstrate how a PF estimator is able to automatically determine the starting configuration of a mechanism from the sequence of readings from one noise sensor. To this end, a computer program was developed which simultaneously runs two processes: (i) a multibody dynamic simulator of a four-bar linkage, similar to that shown in Figure 3(a), and (ii) the PF estimator described in previous sections. The idea is to employ the first simulation as a "ground truth" or "real" mechanism, from which sensor readings are simulated, corrupted with zero-mean additive Gaussian noise (with standard deviation σ_r) and fed to the estimator. By employing a simulated reference mechanism we are able to precisely analyze how diverse kinds of errors affect the performance and accuracy of the estimator.

The mechanism, shown in Figure 9(a), is modeled with natural coordinates [44] by means of eight redundant coordinates $\mathbf{q} = [x_A \; y_A \; x_1 \; y_1 \; x_2 \; y_2 \; x_B \; y_B]^{\mathrm{T}}$, with A and B being the left and right fixed points, respectively. As dynamic formulation we employed the Matrix $\mathbf{R}$ projection method [2] with full-pivoting LU decomposition of the constraint Jacobian, implemented with the *Eigen* C++ library [45]. This allows the code to automatically determine, at each time step, which redundant coordinate in $\mathbf{q}_t$ is better suited for being selected as the independent coordinate. Notice that this is the reason why the acceleration noise in Table 1 has units of linear acceleration (m/s^2), as opposed to the frequent case where a four-bar linkage is modeled with mixed coordinates that include the crank angle. No external forces exist apart from gravity, and only one virtual sensor was simulated: a gyroscope attached to the longest bar –refer to the sketch in Figure 9(a). Notice that this choice leads to angular velocity observations that only indirectly reveal information about the crank angle at the opposite side of the device, which will be computed for each time step to reflect the mechanism position. Two experiments have been performed on this simulation benchmark, as detailed next.

6.1.1 Tracking experiment

Here, the PF estimator was initialized using a uniform distribution to model the situation where the filter has no prior information at all about the mechanism

[8] See `http://www.youtube.com/watch?v=7Zru0oiz36g`

[9] Hosted at `https://github.com/jlblancoc/mbde`

Parameter	**Value**
Number of particles	100
Additive gyroscope noise (σ_r)	0.3 deg/s
Sensor noise assumed by the state observer (σ_s, with $\boldsymbol{\Sigma}_s = \sigma_s^2$ in Eq. (10))	1 deg/s
Equations of motion acceleration noise (σ_a, with $\boldsymbol{\Sigma}_a = \sigma_a^2$ in Eq. (13))	1 m/s^2

Table 1: Parameters of the simulation experiment.

state. Therefore, the initial hypotheses of the PF are distributed as in the examples of Figure 6. For this experiment, the multibody model employed by the PF exactly matches the real mechanism (i.e. lengths, masses, etc.), hence in this sense we have an ideal situation. Most relevant parameters of this setup are summarized in Table 1. It can be observed there how the sensor noise assumed by the PF is larger than the actual simulated noise, that is, $\sigma_s > \sigma_r$. This may seem suboptiomal, since the estimator will not "trust" the readings as much as it should, discarding valuable information. However, artificially increasing the uncertainty is a widely-accepted technique employed to avoid the sample impoverishment problem [46]. Informally, this effect can be described as hypotheses losing the minimal "randomness" required to fit both the multibody model and the sensor noise, which ends up in the assignment of high weights to random samples that are not really the ones that best represent the posterior density distribution.

The results are summarized in Figure 7(a), which shows the ground truth crank angle (i.e. the "real" value being estimated) in solid black, along with the PF estimation over time. We have also computed the average or *expected* crank angle from the filter (in solid red) as well as the corresponding 95% confidence interval, which represents how much uncertainty exist about the mechanism dynamic state. Notice how the filter starts with a large uncertainty, corresponding the the absolute lack of information about the orientation of the crank. We must insist in the impossibility of modeling such a large uncertainty with parametric filters (as the EKF or UKF), since the involved nonlinearities are excessively large for that kind of filters to be useful. In less that 0.5 seconds, however, the filter converges to the actual position of the mechanism, as can be seen in the Figure 7(a). The error in the crank angle estimation is also shown in Figure 7(b), along with its uncertainty.

6.1.2 Robustness benchmark

The second experiment consists of a benchmark against errors in the modeling of the mechanical device. Here, the focus is on determining whether the estimator is able to track the real movement of the mechanism in the case of mismatches between the real model and the parameters of the multibody model employed by the PF. Such benchmark is relevant since, in practice, it becomes difficult to accurately determine the weight, dimensions, inertia matrices and friction parameters of all the elements of a machine.

We performed a characterization of the estimation accuracy for different errors in the length and mass of the crank element. In particular, we evaluated errors of 0% (perfect model), 1%, 2% and 3% with respect to the ground truth. Simultaneously, for each error level the benchmark was also run for a wide range of values of

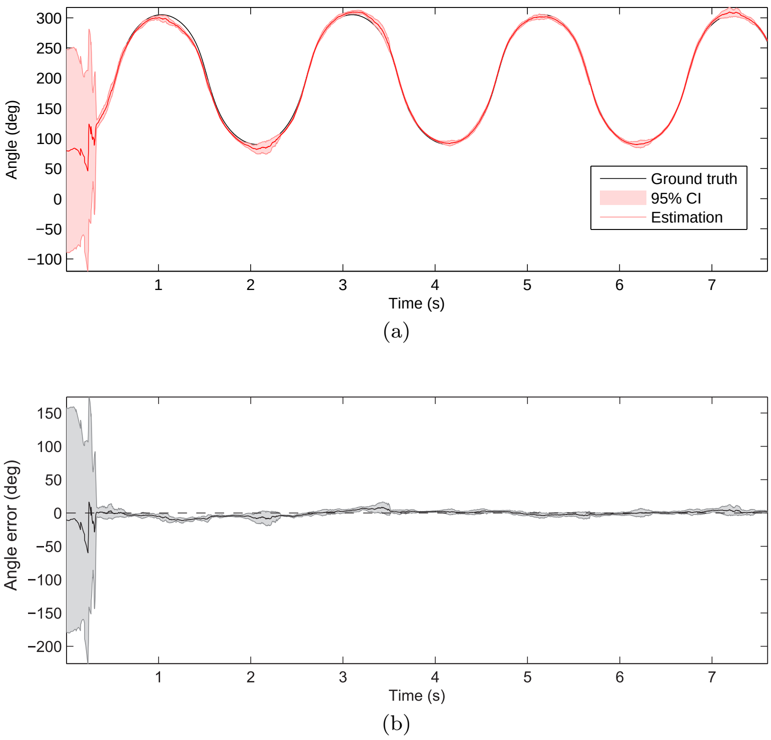


Fig. 7: Results from a simulated mechanism being tracked with a PF state observer. (a) The evolution over time of the degree of freedom of the real mechanism (ground truth) and the probabilistic estimation, including a 95% confidence interval. (b) Error in the tracking and its 95% confidence interval.

σ_a (with $\boldsymbol{\Sigma}_a = \sigma_a^2$), the standard deviation of the acceleration noise in our statistical equations of motion in Eq. (13). The intention is to experimentally verify the theoretical role of $\boldsymbol{\Sigma}_a$, whose optimal value arises as the trade-off between very low values, which reduce the uncertainty by forcing the estimator to follow the kinematics and dynamics of the multibody model, and high values which allows the estimator to fit unexpected friction forces, length errors, etc. In short, that parameter should require some tuning for each specific application.

The results for the particular mechanism at hand, which should not be extrapolated to a more general case, are summarized in Figure 8 as the root mean squared error (RMSE) of the estimated crank angle. In fact, the displayed values are the mean of 100 repetitions of each combination of error level and $\boldsymbol{\Sigma}_a$. Three remarkable insights can be obtained from these curves. Firstly, it is clear that the achievable accuracy strongly depends on how precisely the multibody model represents the real mechanism: the best accuracy (MRSE of $\sim 0.3°$) can be only obtained with a perfect model (0% error in the graph). Note that an imporant practical implication of this result is that inexpensive sensors (e.g. a gyroscope) could replace a more expensive or intrusive one (e.g. an encoder), provided that

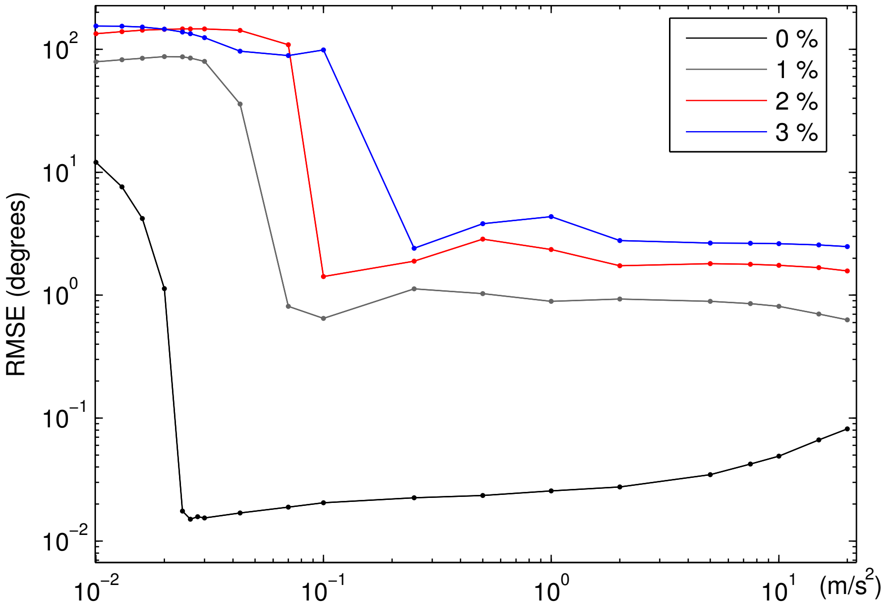


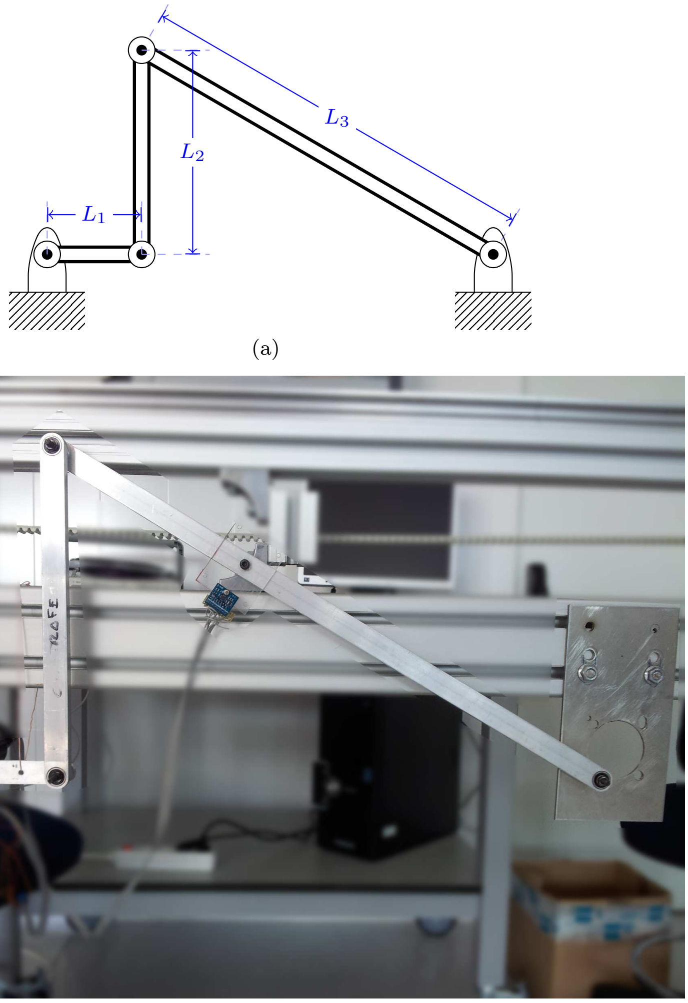
Fig. 8: Results of the benchmark for different levels of mismatch (0%, 1%, 2% and 3%) between the real mechanism and the model assumed by the PF estimator. The horizontal axis stands for σ_a, the standard deviation of the equations of motion noise model, with $\boldsymbol{\Sigma}_a = \sigma_a^2$. The vertical axis stands for the root mean squared error (RMSE) of the expected crank angle as estimated by the filter.

a quality multibody model is available. Secondly, there is not a universal optimal value for $\boldsymbol{\Sigma}_a$, since it depends on the actual level of inaccuracy in the model. As expected from its theoretical meaning, higher values of this parameter are required to achieve the minimum error as the inaccuracy grows. And thirdly, the sample impoverishment problem [46] clearly reveals itself as $\boldsymbol{\Sigma}_a$ becomes too small, becoming specially relevant in the curve for a perfect model (0% error) .

6.2 Real-time tracking of a real mechanism

We also built a physical prototype on which to test the PF estimator, as can be seen in Figure 9(b). The main parameters of the design and the estimator are summarized in Table 2.

Two tests were performed for the two possible configurations of the mechanism: one with the middle bar upwards (which will be called configuration $B = 1$) and another one with it downwards ($B = 2$). The challenging goal of this experiment is to obtain a reasonable estimation of the mechanism state, including its configuration among the two possibilities, from a short sequence of noisy gyroscopic data. Note that the friction in the joints, which is high enough to stopping the mechanism after a few seconds, is totally neglected in the multibody model, making the estimation problem harder.



(a)

(b)

Fig. 9: The four-bar linkage employed as a testbed for our nonlinear, particle filter estimator. (a) Schematic view. Refer to Table 2 for the nominal lengths and masses of each link. (b) The real prototype. The solid-state gyroscope can be seen attached to the longest bar.

As in the simulation experiments, the only acting force is gravity and the unique sensor available to the estimator is an inexpensive triaxial MEMS gyroscope (L3G4200D) attached to one of the links. The real position of the mechanism is obtained by means of a FAULHABER optical encoder with 500 pulses per revolution, attached to the crank. Readings from this encoder are not fed to the state observer, but rather used as a reference to check the quality of the estimation.

For each configuration, the crank is manually raised, then released, and data from the gyroscope and the encoder are recorded until the mechanism stops moving

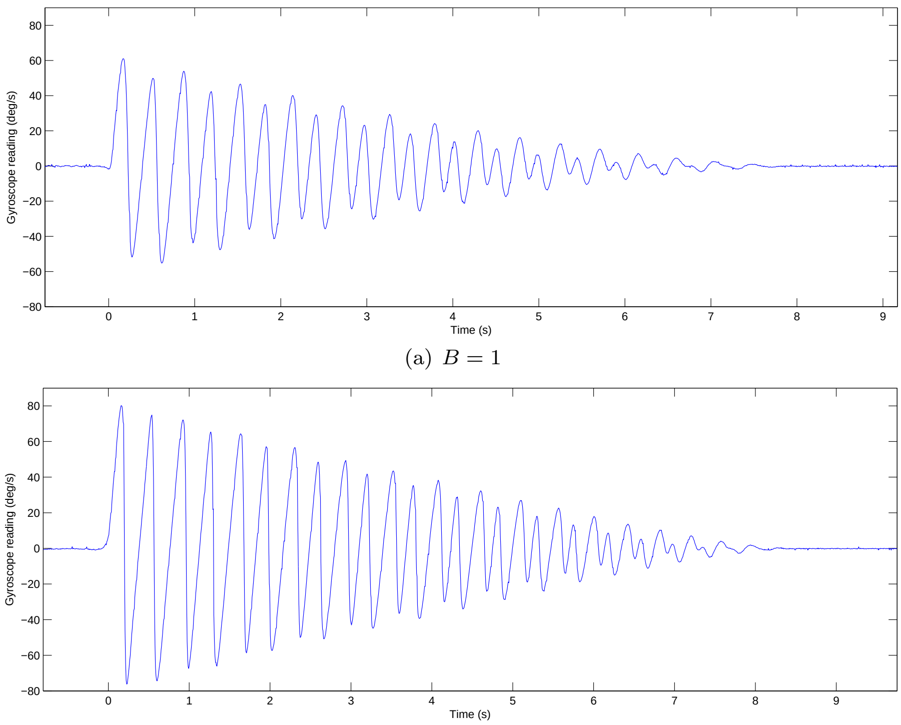
Parameter	**Value**
Number of particles	50
Equations of motion acceleration noise (σ_a, with $\boldsymbol{\Sigma}_a = \sigma_a^2$ in Eq. (13))	$4 \ m/s^2$
Sensor noise assumed by the state observer (σ_s, with $\boldsymbol{\Sigma}_s = \sigma_s^2$ in Eq. (10))	$1 \ deg/s$
Gyroscope measuring period (T_s)	6 ms
PF working period (T)	3 ms
L_1	125 mm
L_2	270 mm
L_3	540 mm
m_1	0.039 kg
m_2	0.161 kg
m_3	0.303 kg

Table 2: Parameters of the real experiment. The L_i and m_i correspond to the lengths and masses of the links, respectively. All of them have been determined experimentally.

due to friction. The starting angle at which the crank is manually taken is arbitrary, since our goal is to demonstrate how the PF is able to recover that information. The raw gyroscopic data for the two different configurations (i.e. see Figure 3) are shown in Figure 10, where it can be appreciated how similar they look like.

One of the most surprising results of this work is the short time required by the filter to correctly determine, from gyroscopic data, the actual assembly of the system, i.e. the unknown B, simultaneously to the determination of the initial position of the crank, which is also unknown. As can be seen in Figure 11(c)–(d), it takes less than 6 sensor readings (about 36 ms) for the probability of the correct configuration to grows up to the unity. Recall that these probability values are extracted from the state observer using Eq. (20). Regarding the reconstruction of the crank angle over time for each experiment, illustrated in Figure 11(a)–(b), there exist larger errors than in the simulation experiment presented in the previous section, specially in the configuration $B = 1$. The better performance for $B = 2$ is due to the largest velocities and accelerations attained in this situation, which provide a better discrimination among particle hypotheses when the unique sensors are gyroscopes. Nevertheless, it must be remarked that, even provided a poor multibody model that disregards all friction forces, the state observer is still able to track the link oscillations at the correct frequency and with approximate amplitudes without any precise information about the initial angle. Although this presents a qualitative improvement with respect to previous state observers based on parametric filters, it becomes patent that considering better multibody models (e.g. with approximate friction forces) is crucial to obtain high quality estimations.

The computer code for this experiment was implemented in C++, compiled in GNU/Linux and executed on an Intel Core i7-3770 CPU at 3.40 GHz. The PF runs with a time step of $T = 3$ ms, incorporating the readings from the gyroscope only when they are ready. The average period of sensory data was $T_s = 6$ ms, which means that in roughly half of the iterations no new sensory data were available, in which case the proposed SIR algorithm remains unmodified but for the weight update stage which is skipped.



(a) $B = 1$

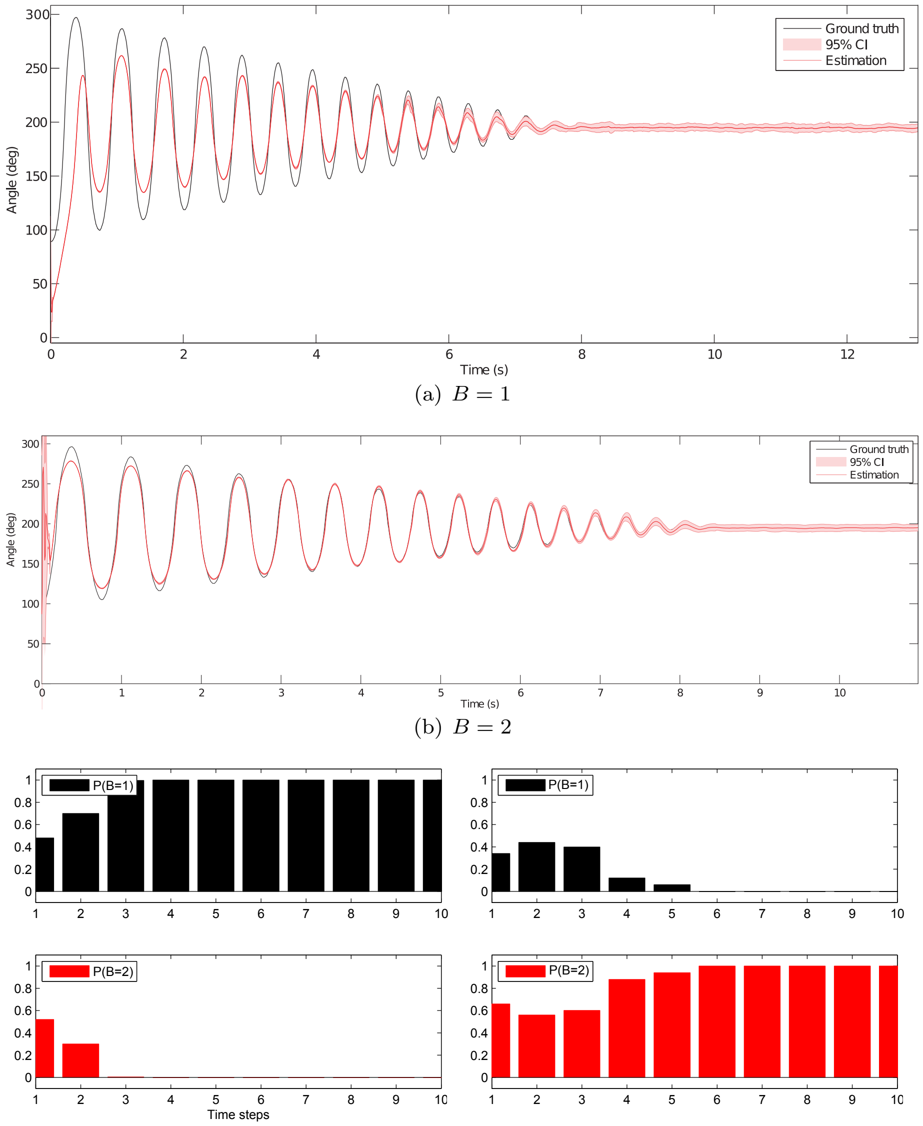
(b) $B = 2$

Fig. 10: Gyroscopic readings for two experiments, one for each possible configuration of the mechanism.

Regarding computational efficiency, our implementation achieves faster-than-real-time execution for a four bar linkage model and 50 particles. For an experiment with real duration of 10.98 seconds, the overall computational cost of running the PF is 9.2 seconds (83.75% of real-time). Indeed, the most expensive stage of our algorithm is the implementation of the Bayesian filtering (lines 1-6 of Algorithm 1) with a 97.47% of the total execution time. Solving the equations of motion for each particle is the most time-consuming task in that stage. The weight renormalization and particle resampling stages represent the remaining 0.03% and 2.50% of the time, respectively.

7 Conclusions

The present work has contributed two significant achievements in the field of probabilistic estimation of MBD. Firstly, a graphical model has been proposed for the first time in the literature. It serves as a theoretical foundation to previously-used estimators such as EKF or UKF, but can be also used as a guide to implement new estimators in future works. Secondly, it was shown how to rigorously derive, from such graph, the equations of a sequential filter for online estimation of a



(a) $B = 1$

(b) $B = 2$

(c) $B = 1$

(d) $B = 2$

Fig. 11: Experiment results from the real prototype. (a)–(b) Real crank angle (solid black line), estimation from the state observer (red) and is 95% confidence interval (gray area) for the two assembly configurations, or branches, of the four-bar linkage. (c)–(d) Time evolution of the probability of the mechanism to be in each branch, as estimated by the proposed method. It is remarkable how quickly the correct solution, i.e. $B = 1$ in (c) and $B = 2$ in (d), becomes dominant.

mechanism. The particular implementation as a particle filter has been shown to be suitable to real-time tracking of small mechanisms with inexpensive sensors, reporting a decent accuracy and robustness to non-modeled effects. For the first time in the literature, an estimator has been presented that can distinguish between the different configurations of the system by accumulating statistical evidence over a few milliseconds of sensory data.

Despite the successful results, two research lines remain open after this work: (i) performing a systematic and consistent benchmarking among different state observers in order to quantitatively establish their pros and cons in different application areas, and (ii) developing new estimators from the presented DBN, including potential ways of fusing the advantages of both parametric and non-parametric methods or the determination of system parameters or external forces as problem unknowns. More future work is clearly required in order to address these topics.

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-Verlag, 1994.
3. W. Schiehlen, "Multibody system dynamics: roots and perspectives," *Multibody system dynamics*, vol. 1, no. 2, pp. 149–188, 1997.
4. A. A. Shabana, "Flexible multibody dynamics: review of past and recent developments," *Multibody system dynamics*, vol. 1, no. 2, pp. 189–222, 1997.
5. M. Géradin and A. Cardona, *Flexible multibody dynamics: a finite element approach.* Wiley, 2001.
6. A. A. Shabana, *Dynamics of multibody systems.* Cambridge university press, 2005.
7. J. Wittenburg, *Dynamics of multibody systems.* Springer, 2008.
8. O. A. Bauchau and A. Laulusa, "Review of contemporary approaches for constraint enforcement in multibody systems," *Journal of Computational and Nonlinear Dynamics*, vol. 3, no. 1, p. 011005, 2008.
9. A. Jain, "Unified formulation of dynamics for serial rigid multibody systems," *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 3, pp. 531–542, 1991.
10. R. Featherstone, "A divide-and-conquer articulated-body algorithm for parallel O (log (n)) calculation of rigid-body dynamics. Part 1: Basic algorithm," *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 867–875, 1999.
11. R. Featherstone, "A divide-and-conquer articulated-body algorithm for parallel O (log (n)) calculation of rigid-body dynamics. Part 2: Trees, loops, and accuracy," *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 876–892, 1999.
12. K. Kreutz-Delgado, A. Jain, and G. Rodriguez, "Recursive formulation of operational space control," *The International journal of robotics research*, vol. 11, no. 4, pp. 320–328, 1992.
13. M. Poursina and K. S. Anderson, "An extended divide-and-conquer algorithm for a generalized class of multibody constraints," *Multibody System Dynamics*, vol. 29, no. 3, pp. 235–254, 2013.
14. R. Prakash, P. D. Burkhart, A. Chen, K. A. Comeaux, C. S. Guernsey, D. M. Kipp, L. V. Lorenzoni, G. F. Mendeck, R. W. Powell, T. P. Rivellini, *et al.*, "Mars science laboratory entry, descent, and landing system overview," in *Aerospace Conference, 2008 IEEE*, pp. 1–18, IEEE, 2008.
15. A. Kolmogorov, "On analytical methods in the theory of probability," *Math. Ann*, vol. 104, pp. 415–458, 1931.
16. A. Sandu, C. Sandu, and M. Ahmadian, "Modeling multibody systems with uncertainties. Part I: Theoretical and computational aspects," *Multibody System Dynamics*, vol. 15, no. 4, pp. 369–391, 2006.
17. C. Sandu, A. Sandu, and M. Ahmadian, "Modeling multibody systems with uncertainties. Part II: Numerical applications," *Multibody System Dynamics*, vol. 15, no. 3, pp. 241–262, 2006.

18. M. Poursina, "An efficient application of polynomical chaos expansion for the dynamic analysis of multibody systems with uncertainty," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE)*, 2014.
19. J. Cuadrado, D. Dopico, J. A. 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. Cited By (since 1996):2.
20. 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.
21. J. Cuadrado, D. Dopico, J. A. Perez, and R. Pastorino, "Influence of the sensored magnitude in the performance of observers based on multibody models and the extended kalman filter," in *ECCOMAS Thematic Conference on Multibody Dynamics*, pp. 126–127, 2009.
22. C. Bishop, *Pattern recognition and machine learning.* Springer New York, 2006.
23. F. V. Jensen, *An introduction to Bayesian networks*, vol. 74. UCL press London, 1996.
24. D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques.* MIT press, 2009.
25. R. Kindermann, J. L. Snell, *et al.*, *Markov random fields and their applications*, vol. 1. American Mathematical Society Providence, RI, 1980.
26. H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, 2004.
27. A. Dawid, "Conditional independence in statistical theory," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–31, 1979.
28. A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice.* Springer, 2001.
29. J. M. Porta, L. Ros, O. Bohigas, M. Manubens, C. Rosales, and L. Jaillet, "An open-source toolbox for motion analysis of closed-chain mechanisms," in *Computational Kinematics*, pp. 147–154, Springer, 2014.
30. G. Kitagawa, "Monte carlo filter and smoother for non-gaussian nonlinear state space models," *Journal of computational and graphical statistics*, vol. 5, no. 1, pp. 1–25, 1996.
31. R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
32. D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches.* Wiley.com, 2006.
33. B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications.* Artech House, 2004.
34. R. Pastorino, D. Richiedei, J. Cuadrado, and A. Trevisani, "State estimation using multibody models and nonlinear kalman filters," *International Journal of Non-Linear Mechanics*, vol. 53, pp. 83–90, 2013.
35. D. Rubin, "A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm," *Journal of the American Statistical Association*, vol. 82, no. 398, pp. 543–546, 1987.
36. M. Arulampalam, S. Maskell, N. Gordon, T. Clapp, D. Sci, T. Organ, and S. Adelaide, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
37. J.-L. Blanco, J. González, and J.-A. Fernández-Madrigal, "Optimal filtering for non-parametric observation models: applications to localization and slam," *The International Journal of Robotics Research*, vol. 29, no. 14, pp. 1726–1742, 2010.
38. A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 176–183, 2000.
39. N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEEE Proceedings on Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.
40. R. Douc and O. Cappé, "Comparison of resampling schemes for particle filtering," in *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pp. 64–69, IEEE, 2005.
41. J. S. Liu and R. Chen, "Blind deconvolution via sequential imputations," *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 567–576, 1995.

42. M. Moakher, "Means and averaging in the group of rotations," *SIAM journal on matrix analysis and applications*, vol. 24, no. 1, pp. 1–16, 2002.
43. S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of the American Control Conference*, vol. 3, pp. 1628–1632, 1995.
44. M. A. Serna, R. Avilés, and J. García de Jalón, "Dynamic analysis of plane mechanisms with lower pairs in basic coordinates," *Mechanism and Machine Theory*, vol. 17, no. 6, pp. 397–403, 1982.
45. G. Guennebaud, B. Jacob, *et al.*, "Eigen v3." http://eigen.tuxfamily.org, 2010.
46. J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 146, no. 1, pp. 2–7, 1999.