Contact State Segmentation using Particle Filters for Programming by Human Demonstration in Compliant Motion Tasks

Wim Meeussen, Johan Rutgeerts, Klaas Gadeyne, Herman Bruyninckx, Joris De Schutter
Department of Mechanical Engineering, Katholieke Universiteit Leuven,
Celestijnenlaan 300B, B-3001 Leuven, Belgium.
Email: wim.meeussen@mech.kuleuven.be

***Abstract*— This paper presents a contribution to *programming by human demonstration*, in the context of *compliant motion* task specification for sensor-controlled robot systems that physically interact with the environment. One wants to learn about the geometric parameters of the task and segment the total motion executed by the human into subtasks for the robot that can each be executed with simple compliant motion task specifications. The motion of the human demonstration tool is sensed with a 3D camera, and the interaction with the environment is sensed with a force sensor in the human demonstration tool. Both measurements are uncertain, and do not give direct information about the *geometric* parameters of the contacting surfaces, or about the *contact formations* encountered during the human demonstration. The paper uses a Bayesian Sequential Monte Carlo method (also known as a *particle filter*) to do the simultaneous estimation of the contact formation (discrete information) and the geometric parameters (continuous information). The simultaneous contact formation segmentation and the geometric parameter estimation are helped by the availability of a *contact state graph* of all possible contact formations. The presented approach applies to all compliant motion tasks involving polyhedral objects with a known geometry, where the uncertain geometric parameters are the poses of the objects. This work improves the state of the art by *scaling* the contact estimation to all possible contacts, by presenting a prediction step based on the *topological information* of a contact state graph, and by presenting efficient algorithms that allow the estimation to operate in *realtime*. In real world experiments it is shown that the approach is able to discriminate in realtime between some 250 different contact formations in the graph.**

***Index Terms*— compliant motion, Bayesian estimation, particle filter, human demonstration, task segmentation**

I. Introduction

COMPLIANT motion [1] refers to tasks in which an object held by a manipulator moves while maintaining contact with the environment. The force interaction at the contact is used to guide the manipulated object along the surface of the environmental object, to help overcome geometric uncertainties associated with the task. Mason introduced the *Task Frame Formalism* (TFF) [2] as an intuitive interface to compliance and force control for the specification of force controlled robot tasks in the *Hybrid Control Paradigm* (HCP). Bruyninckx and De Schutter made an extensive catalog of TFF models and specifications in [3], which, in the context of *programming by human demonstration*, [4], [5], can be used as primitive building blocks to create a compliant motion task from processed sensor data collected during a human demonstration.

Although the TFF has applications in many contact tasks, [6], [7], it cannot model “complex” contact situations, i.e., with multiple contacts, even with polyhedral objects. Recently, De Schutter presented a constraint-based task specification framework that overcomes the limitations of the TFF, and provides a powerful interface to specify complex compliant motion tasks involving multiple (contact) constraints [8]. The work presented in this paper is compatible with this task specification framework.

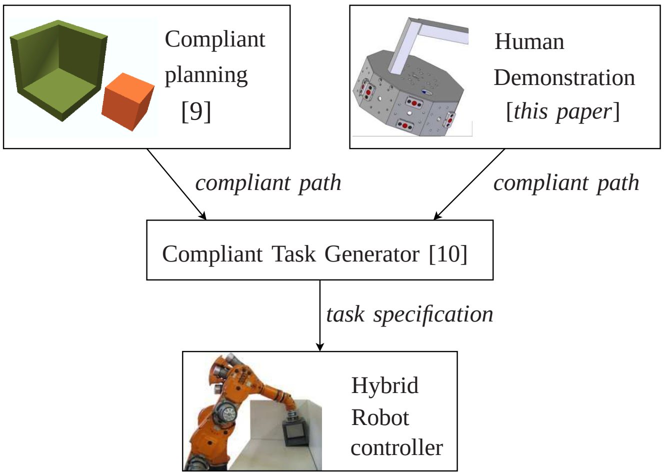

Fig. 1. Previous research covered the integration of compliant planning and force controlled compliant motion [10]. In this research the offline planner is replaced by *programming by human demonstration*.

Even between two simple polyhedral objects, *hundreds* of contact formations are possible, and, hence, hundreds of transitions between neighboring contact formations. In [9] Ji and Xiao developed a compliant motion planner which automatically calculates a compliant path between two compliant configurations, given the geometric model of the objects. The resulting compliant path is given by a sequence of relative poses between the objects, and their contact formations. In [10] Meeussen et al. automatically converted this path into a task specification for a force controlled manipulator, using the *Compliant Task Generator* (see Fig. 1). In this paper, a compliant path is obtained using *programming by human* *demonstration*, where a human demonstrates the desired compliant motion task. The demonstration can be performed in a virtual environment using a haptic device, in the real world by directly interacting with a robot through a master slave system, or, like the example presented in this paper, by observing human motion when the demonstrator directly manipulates the objects in the environment without the use of a robot. After the demonstration, in an interpretation step, the sensor data is translated into a compliant path, given the geometric model of the objects. This path is defined by the same parameters as a path generated by the compliant motion planner, and can be directly converted into a task specification for a force controlled manipulator by the compliant task generator. The same approach as used in the interpretation step, can be used during the execution of a compliant motion task, to monitor if the executed compliant path is the same as the specified compliant path.

Major challenges in the automatic translation from human compliant motion demonstration into a path plan of a compliant motion are: (i) to recognize the *contact formation* to which the human demonstration is currently subjected, (ii) to estimate the geometric parameters of that contact formation (i.e., position of contact point(s), direction of contact normal(s), etc.), and (iii) to detect when exactly the human demonstration execution changes between two contact formations.

Initial research on the identification of contact formations mainly focused on two different approaches: (i) ad hoc identification strategies that exploit geometric knowledge of the contacting objects, but that have a limited stochastic foundation (e.g. [11], [12]) and (ii) Hidden Markov Model based (hence stochastic) solutions to assembly problems that can recognize contact formation transitions very fast but only with limited allowed uncertainty (e.g. [13], [14]).

Previous work by the research group of the authors has pioneered (in the domain of force-controlled compliant motion) the integration of both approaches, applying state-of-the-art Bayesian probability techniques that have already become much more popular and successful in other robotics domains with similar estimation challenges, namely mobile robot navigation (e.g. [15], [16]) and 3D computer vision with natural landmarks (e.g. [17]). All these domains must match the sensor data to available “maps” of primitive geometric building blocks, such as, for example, polyhedral structures in the world. Not surprisingly, all domains are confronted with similar *hybrid state space* estimation problems (also called “data association”): the estimation problem involves some *continuous* state parameters, that get their precise meaning only in the context of a (“hidden”) *discrete* state parameter. This latter, symbolic parameter indicates the *model* that links the sensor data to the continuous state parameters. For example, the discrete state in this paper is a so-called *(polyhedral) contact formation*, such as an *edge-face* contact; the continuous state parameters are then the position of the edge and the face.

In mobile robotics and 3D visual reconstruction, the concept of *SLAM* (Simultaneous Localization and Map Building) is a very active research field, which goes a bit beyond the scope of this paper: this paper only does localization, and no map building. The Rao-Blackwellized particle filter [18] is often used for SLAM, where it dramatically improves the performance by estimating of the position of hundreds of landmarks independently using Kalman filters, while the position of the mobile robot is tracked using particle filters. However, for contact state estimation this type of filter cannot by used because (i) the geometrical parameters are not independent and (ii) the probability on a change in contact state not only depends on an input, but also on the geometrical parameters. Slaets et al. [19] already presented SLAM results in compliant motion, by building a geometric model of an unknown environment from a given number of primitives, and recognizing the contact transitions. Their approach is based on the Non-Minimal State Kalman Filter [20], and is valid if the estimation has converged to a unimodal Gaussian before a contact transition. They only allow new contact constraints to be added gradually, and no contact constraints to be removed. Recently Gadeyne et al. [21] developed a *particle filter* [22] to recognize, simultaneously, contact transitions, and estimate geometric pose parameters of a *known* geometric model in an unknown pose. Their approach is able to estimate (continuous) geometric parameters with a large uncertainty, and simultaneously recognize (discrete) contact transitions in an experiment consisting of six initially known possible contacts. The low number of possible contacts allows them to use a simple prediction step in the particle filter, in combination with a small number of particles.

This paper generalizes and scales the approach of Gadeyne et al. to cope with *all possible contacts* between two polyhedral objects. To cope with this increased complexity, a more accurate prediction step is used, based on the *topological information* contained in a contact state graph, [23], [24], and the pose of the contacting objects. This paper also presents efficient algorithms for the pose and consistency measurement equations, allowing the estimators to be used in *realtime*.

The paper is organized as follows. Section II briefly reviews the concepts of contact formations and the contact state graph. Section III describes the demonstration tool which is used to collect sensor data during human demonstration in compliant motion. The interpretation of this sensor data, using Bayesian estimation techniques, is covered in Section IV, while the efficient algorithms for the estimation are discussed in Section V. Section VI describes the real world experiments that validate the presented approach. Section VII discusses the strengths and limitations of the approach. Finally, Section VIII contains conclusions and future work.

II. Contact Formations and the Contact State Graph

A. Contact Formations

The notion of *principal contacts* (PCs) was introduced [25] to describe a contact primitive between two surface elements of two polyhedral objects in contact, where a surface element can be a face, an edge or a vertex. The *boundary elements* of a face are the edges and vertices bounding it, and the boundary elements of an edge are the vertices bounding it. Formally, a PC denotes the contact between a pair of surface elements which are not boundary elements of other contacting surface elements. Fig. 2 shows the six non-degenerate[1] PCs that can be formed between two polyhedral objects. Each non-degenerate PC is associated with a *contact plane*, defined by a contacting face or the two contacting edges at an edge-edge PC.

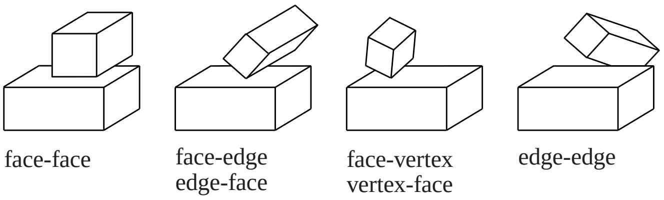

Fig. 2. The six possible non-degenerate principal contacts (PCs) between two polyhedral objects.

A general contact state between two objects can be characterized topologically by the set of PCs formed, called a *contact formation* (CF). Each configuration of two objects, i.e. their relative pose in space, compliant to the constraints of a CF, is called a *CF-compliant configuration*, denoted by a pose $\boldsymbol{X}$. Any motion formed by a sequence of CF-compliant configurations is called a *CF-compliant motion*. A homogeneous transformation matrix represents the pose of an object a relative to an object b:

$$\boldsymbol{X}_a^b = \begin{bmatrix} \boldsymbol{R}_a^b & \boldsymbol{p}_a^b \\ \mathbf{0} & 1 \end{bmatrix}, \tag{1}$$

where $\boldsymbol{p}_a^b$ represents the position vector from a to b, and $\boldsymbol{R}_a^b$ represents a the orientation matrix between a and b.

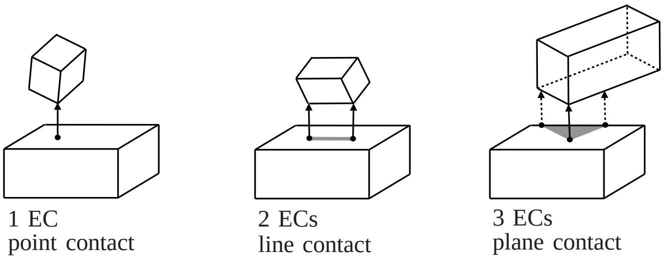

Fig. 3. A principal contact (PC) can be decomposed into one or more elementary contacts (ECs), which are associated with a contact point and a contact normal. The dotted arrows indicate the edge-edge ECs, and the full arrows indicate the vertex-face or face-vertex ECs.

A PC can be decomposed into one or more *Elementary Contacts* (ECs), providing a lower level description of the CF, as shown in Fig. 3. The three types of ECs (face-vertex, vertex-face and edge-edge) are shown in the two examples at the right of Fig. 2. An EC is a point contact and is associated with a *contact point* and a *contact normal*. For the decomposition of a PC into ECs, we use the boundary points of the *contacting area*, as shown by the gray areas in Fig. 3. The contacting area can be a single point (for a vertex-face, face-vertex or edge-edge contact), a line (for a face-edge or edge-face contact) or a polygon (for a face-face contact).

B. Contact State Graph

Xiao and Ji developed a divide-and-merge approach [23], [24] to generate a compact, simplified representation of the contact state space between two polyhedral objects, as a *contact state graph* G. In G a node represents a CF, and an arc connecting two nodes represents the adjacency relationship between the CFs of the nodes. Two CFs CF_i and CF_j are adjacent if a compliant motion from a CF_i-compliant configuration to CF_j-compliant configuration exists, which only includes CF_i and CF_j-compliant configurations. Fig. 4 shows an example of a contact state graph containing seven different CFs and their adjacency relationships. The approach generates a contact state graph from a given set of locally most constrained CFs, using a relaxation of the contact constraints. The approach was implemented to automatically generate a contact state graph that can contain hundreds of contact formations.

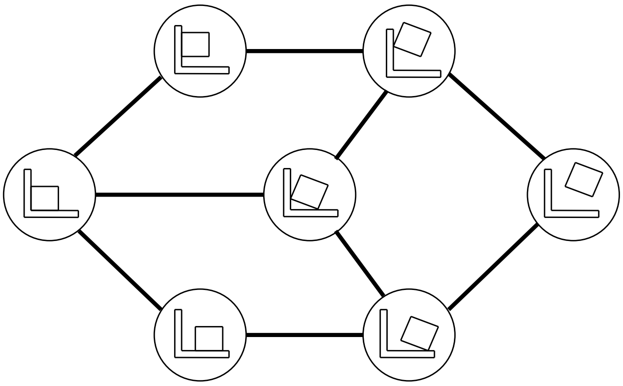

Fig. 4. A complete contact state graph shows all possible CFs (nodes) and transitions between neighboring CFs (arcs). While this figure shows a simplified example that only contains 7 CFs, a real contact state graph of two 3-dimensional polyhedral objects contains hundreds of CFs.

III. Demonstration Tool

In programming by human demonstration, a task specification for a compliant task involving a manipulated object and its environment is obtained by observing a human demonstrate the desired task. In this research the human demonstrator directly interacts with the manipulated object, using a *demonstration tool* which is mounted onto the manipulated object.

A. Design

A CAD model of the tool's design is shown in Fig. 5, while Fig. 10 shows the tool during an experiment. A handle on top provides an easy grasp for the human demonstrator to manipulate the demonstration tool and the object attached to it.

The six-dimensional pose of the demonstration tool is measured by a Krypton K600 6D optical system (Fig. 6) measuring the spatial positions of LEDs attached to the demonstration tool, at 100 $[Hz]$ or more, with a volumetric accuracy of 90 $[\mu m]$. The demonstration tool itself has a hollow cylinder-like shape, consisting of nine faces in 40 $[degrees]$ increments.

[1] The vertex-vertex, vertex-edge, edge-vertex and edge-edge-parallel PCs are called degenerate, as it is difficult to achieve a stable contact that includes one of these PCs. Therefore only non-degenerate PCs are considered in this paper.

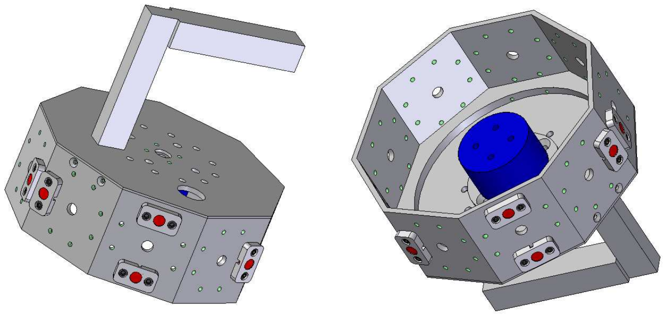

Fig. 5. The design of the demonstration tool, seen from above (left) and below (right). The red LED markers are used to track the pose of the demonstration tool in space, while a wrench sensor mounted inside the demonstration tool measures the interaction forces with the environment.

On each of the faces, up to four LED markers can be mounted on eight different positions. Inside the demonstration tool, a *JR3* wrench sensor is mounted between the demonstration tool and the manipulated object, in order to measure the wrench $\boldsymbol{w}_m$ applied by the human demonstrator to the manipulated object:

$$\boldsymbol{w}_m = \left[\begin{array}{cccccc} f_x & f_y & f_z & \tau_x & \tau_y & \tau_z \end{array} \right]^T . \tag{2}$$

f denotes a linear force, and τ is a moment.

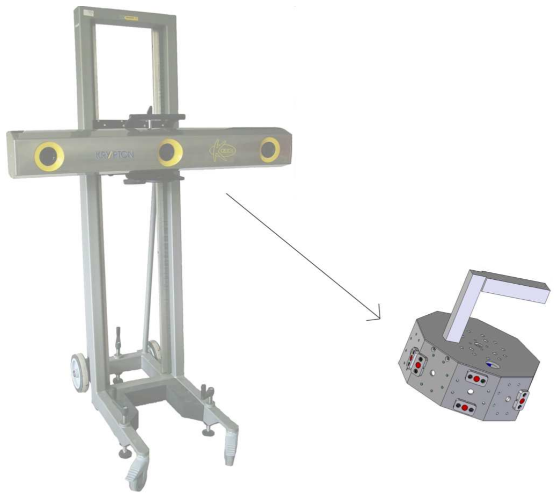

Fig. 6. The Krypton K600 6D optical system uses three cameras and triangulation algorithms to accurately measure the spacial position of each of the LED markers on the demonstration tool.

B. Pose and Twist Estimation

While the wrench $\boldsymbol{w}_m$ is directly measured by a physical sensor, the pose $\boldsymbol{X}_m$ and twist $\boldsymbol{t}_m$ between the contacting objects are indirectly measured through the position of the LED markers. A twist is a screw vector containing both translational and rotational velocities:

$$\boldsymbol{t}_m = \left[\begin{array}{cccccc} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{array} \right]^T . \tag{3}$$

When $b \geq 4$ non-collinear LED markers $L_1 \ldots L_b$ on the demonstration tool are visible to all three cameras, the relative pose $\boldsymbol{X}_t^c$ between the demonstration tool (t) and the camera (c) can be calculated using the method described below. The positions of the visible LED markers relative to the camera are represented by $\boldsymbol{p}^c_{L_1} \ldots \boldsymbol{p}^c_{L_b}$. The positions of the visible LED markers relative to the demonstration tool are constants obtained during an initial calibration phase (Section III-C), and are represented by $\boldsymbol{p}^t_{L_1} \ldots \boldsymbol{p}^t_{L_b}$. The pose matrix $\boldsymbol{X}_c^t$ transforms the positions of the LED markers from the camera to the demonstration tool [26]:

$$\boldsymbol{X}_c^t \left[\begin{array}{ccc} \boldsymbol{p}^t_{L_1} & \ldots & \boldsymbol{p}^t_{L_b} \\ 1 & \ldots & 1 \end{array} \right] = \left[\begin{array}{ccc} \boldsymbol{p}^c_{L_1} & \ldots & \boldsymbol{p}^c_{L_b} \\ 1 & \ldots & 1 \end{array} \right] . \tag{4}$$

Reducing the pose matrix to a rotation $\boldsymbol{R}_c^t$ and a translation $\boldsymbol{p}_c^t$, it can be calculated by:

$$\left[\begin{array}{cc} \boldsymbol{R}_c^t & \boldsymbol{p}_c^t \end{array} \right] = \left[\begin{array}{ccc} \boldsymbol{p}^c_{L_1} & \ldots & \boldsymbol{p}^c_{L_b} \end{array} \right] \left[\begin{array}{ccc} \boldsymbol{p}^t_{L_1} & \ldots & \boldsymbol{p}^t_{L_b} \\ 1 & \ldots & 1 \end{array} \right]^\dagger \tag{5}$$

in which † represents the Moore Penrose pseudo-inverse [27] of a matrix. The pose defined by $\boldsymbol{R}_c^t$ and $\boldsymbol{p}_c^t$ can be transformed into a minimal six-dimensional pose representation:

$$\boldsymbol{x}_m = \left[\begin{array}{cccccc} x & y & z & \alpha & \beta & \gamma \end{array} \right]^T , \tag{6}$$

using α, β and γ ZYX-Euler angles[2]. This "measured" pose $\boldsymbol{x}_m$ is an input to a linear estimation problem to obtain both the demonstration tool's pose $\boldsymbol{X}_m$ and twist $\boldsymbol{t}_m$, based on a constant acceleration model [28]. A constant acceleration model assumes the demonstration tool is moving with a constant acceleration, but the acceleration is constantly adapted based on the pose measurements. This results in a smooth estimation for the pose and twist of the demonstration tool, while deriving the twist from the pose measurements would result in a noisy and inaccurate twist estimation. The Kalman filter [29] is the preferred tool for this linear estimation problem with low uncertainties. The filter uses an 18-dimensional state vector which contains the pose $\boldsymbol{x}$, the velocity $\dot{\boldsymbol{x}}$ and the acceleration $\ddot{\boldsymbol{x}}$. All three 6-dimensional parameters are estimated from the measured pose $\boldsymbol{x}_m$.

The filter's system update extrapolates the estimated pose $\hat{\boldsymbol{x}}$, velocity $\hat{\dot{\boldsymbol{x}}}$ and acceleration $\hat{\ddot{\boldsymbol{x}}}$ at time step k, to make a prediction of the pose $\tilde{\boldsymbol{x}}$, velocity $\tilde{\dot{\boldsymbol{x}}}$ and acceleration $\tilde{\ddot{\boldsymbol{x}}}$ at time step $k+1$, using a constant acceleration model:

$$\left[\begin{array}{c} \tilde{\boldsymbol{x}} \\ \tilde{\dot{\boldsymbol{x}}} \\ \tilde{\ddot{\boldsymbol{x}}} \end{array} \right]_{k+1} = \left[\begin{array}{ccc} \boldsymbol{I} & \boldsymbol{\Delta t} & \boldsymbol{\Delta t}^2/2 \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{\Delta t} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{array} \right] \left[\begin{array}{c} \hat{\boldsymbol{x}} \\ \hat{\dot{\boldsymbol{x}}} \\ \hat{\ddot{\boldsymbol{x}}} \end{array} \right]_{k} . \tag{7}$$

The filter's measurement update uses the difference between the measured pose $\boldsymbol{x}_m$ and the predicted pose $\tilde{\boldsymbol{x}}$ at time step $k+1$, to update the estimated pose $\hat{\boldsymbol{x}}$, velocity $\hat{\dot{\boldsymbol{x}}}$ and acceleration $\hat{\ddot{\boldsymbol{x}}}$ at time step $k+1$:

$$\left[\begin{array}{c} \hat{\boldsymbol{x}} \\ \hat{\dot{\boldsymbol{x}}} \\ \hat{\ddot{\boldsymbol{x}}} \end{array} \right]_{k+1} = \left[\begin{array}{c} \tilde{\boldsymbol{x}} \\ \tilde{\dot{\boldsymbol{x}}} \\ \tilde{\ddot{\boldsymbol{x}}} \end{array} \right]_{k+1} + \boldsymbol{K} \left(\boldsymbol{x}_m - \tilde{\boldsymbol{x}} \right) , \tag{8}$$

[2] The singularities of the representation can be avoided by using a different representation (like Roll Pitch Yaw angles) depending on the position.

in which $\boldsymbol{K}$ is called the Kalman gain [29], which is a function of the Gaussian uncertainty on the estimated state, and the additive Gaussian noise on the measured pose $\boldsymbol{x}_m$.

The minimal representation of the estimated pose $\hat{\boldsymbol{x}}$, which uses ZYX Euler angles, can be transformed into the pose $\boldsymbol{X}_m$ which uses a rotation matrix [30]:

$$\boldsymbol{X}_m = f\left(\hat{\boldsymbol{x}}\right). \tag{9}$$

Its derivative $\dot{\hat{\boldsymbol{x}}}$ represents the change of ZYX Euler angles over time. For given ZYX Euler angles α, β and γ, this derivative of ZYX Euler angles can be transformed into the twist $\boldsymbol{t}_m$ using [30]:

$$\boldsymbol{t}_m = \begin{bmatrix} \boldsymbol{I} & & \mathbf{0} & \\ & 0 & -sin(\alpha) & cos(\alpha)cos(\beta) \\ \mathbf{0} & 0 & cos(\alpha) & sin(\alpha)cos(\beta) \\ & 1 & 0 & -sin(\beta) \end{bmatrix} \dot{\hat{\boldsymbol{x}}}. \tag{10}$$

C. Calibration

Using the Krypton camera system, the pose of the demonstration tool relative to the camera is accurately measured. However, the pose of the camera relative to a world reference is initially unknown. Also, to derive the contact wrench from the total measured wrench, a gravity compensation algorithm is used, which requires the knowledge of the mass and the center of gravity of the manipulated object, as well as the offsets on the measured wrench. These parameters are estimated during a calibration procedure, previously presented by Rutgeerts in [31]. The calibration procedure is based on the *Non-Minimal State Kalman Filter* [20] to cope with the non-linear measurement models.

IV. Simultaneous Recognition of Contact Formations and Estimation of Geometric Parameters

A compliant motion task can be segmented into a sequence of CFs. At each CF, different contact constraints apply. Therefore, to estimate uncertain geometric parameters of the objects involved in a compliant motion task, the knowledge of the current CF model is required. This means that the estimation problem for compliant motion tasks, consists of two connected sub-problems: the recognition of the (discrete) CF and the estimation of (continuous) geometric parameters. For the simultaneous recognition of CFs and the estimation of geometric parameters, a hybrid *Probability Density Function* (PDF) is required.

In this paper particle filters [22] are used to implement this hybrid estimation problem. Gadeyne et al. [21] show that while Kalman filter variants cannot cope with the cross-dependency between discrete and continuous variables, using a hybrid PDF, particle filters can. The particle filter algorithm updates the discrete CF and continuous geometric parameters in a two step approach. In the first step the system model makes a prediction for the next CF and the geometric parameters. In the second step the measurement model corrects this prediction based on sensor data. This section describes the models that are used in the estimation problem. Section V describes the implementation details.

A. Hybrid Probability Density Function

A hybrid PDF contains both continuous and discrete variables. A time-invariant variable is called a *parameter*, while a time-dependent variable is called a *state*. The continuous parameters in this paper are called the geometric parameters, denoted by $\boldsymbol{\Theta}$, and represent the pose of the manipulated object relative to the demonstration tool and the pose of the environmental object relative to a world reference. Note that while the pose of the objects is unknown, their geometry is known. The discrete state in this paper represents the CF at time step k, denoted by $\mathcal{CF}_k$. Fig. 7 shows an example of a hybrid PDF, for a one-dimensional continuous parameter $\boldsymbol{\Theta}$ and a one-dimensional discrete state $\mathcal{CF}$. The hybrid PDF represents the belief that, at time step k, the discrete state $\mathcal{CF}_k$ is j, with $0 \leq j <$ # CFs, and the continuous parameters $\boldsymbol{\Theta}$ have a certain value $\boldsymbol{\theta}$, given that the measurements $\boldsymbol{Z}_{1\ldots k}$ have a certain value $\boldsymbol{z}_{1\ldots k}$:

$$P\left(\boldsymbol{\Theta} = \boldsymbol{\theta}, \mathcal{CF}_k = j \mid \boldsymbol{Z}_{1\ldots k} = \boldsymbol{z}_{1\ldots k}\right).^{3} \tag{11}$$

Each $\mathcal{CF}_k = j$ of the hybrid PDF has its own continuous PDF:

$$P\left(\boldsymbol{\theta} \mid \mathcal{CF}_k = j, \boldsymbol{z}_{1\ldots k}\right). \tag{12}$$

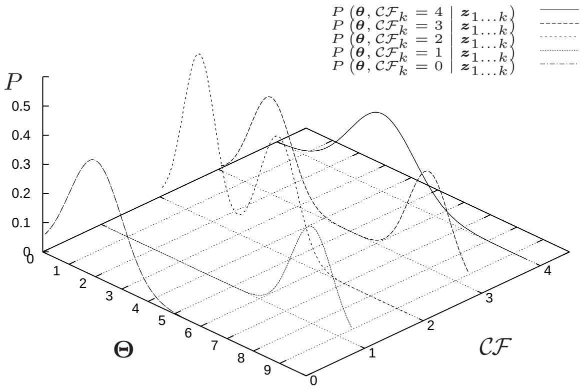

Fig. 7. An example of a probability density function (PDF) of a hybrid joint density, with a one-dimensional continuous geometric parameter $\boldsymbol{\Theta}$ and a one-dimensional discrete state $\mathcal{CF}$. All the information about the system is contained in this one hybrid PDF, while traditional Kalman filters do not have sufficient degrees of freedom in their Gaussian PDFs to represent the same information.

The presented approach is based on particle filters, and therefore the hybrid PDF is represented by a number of discrete samples or *particles*. Each particle corresponds to one possible value of the hybrid state: one of the possible values of each discrete state, and one value for each of the continuous parameters. With enough particles, it is possible to make a good approximation of any discrete, continuous or hybrid PDF. This allows particle filters to deal with virtually any type of PDF, while Kalman filters can only deal with Gaussian PDFs. However there is always a trade-off between the number of

[3] In the rest of the paper, the notation $\boldsymbol{A} = \boldsymbol{a}$ is shortened into $\boldsymbol{a}$, wherever the distinction between a stochastic variable and an actual value is unambiguous.

particles and the performance and memory requirements of the filter.

B. System Model – Prediction

The prediction step uses the system model to make a prediction for the hybrid joint density at time step k, given the hybrid joint density at time step $k-1$:

$$P\left(\boldsymbol{\theta}, \mathcal{CF}_k = j \mid \boldsymbol{z}_{1\ldots k-1}\right) = \sum_i P\left(\mathcal{CF}_k = j \mid \boldsymbol{\theta}, \mathcal{CF}_{k-1} = i\right) P\left(\boldsymbol{\theta}, \mathcal{CF}_{k-1} = i \mid \boldsymbol{z}_{1\ldots k-1}\right). \tag{13}$$

This simplified prediction step is valid because the estimated geometric parameters $\boldsymbol{\Theta}$ are time-invariant, and only the state $\mathcal{CF}$ changes during time [21]. The system model is a state transition prediction function that expresses the belief in a CF transition from a $\mathcal{CF}_{k-1} = i$ at time step $k-1$, to a $\mathcal{CF}_k = j$ at time step k, given the geometric parameters $\boldsymbol{\theta}$, and is defined by:

$$P\left(\mathcal{CF}_k = j \mid \boldsymbol{\theta}, \mathcal{CF}_{k-1} = i\right). \tag{14}$$

To predict the next CF out of hundreds of theoretically possible next CFs, the topological information from the objects' contact state graph is used. Between each two CFs that are connected in the contact state graph, a direct CF transition is possible, while between two unconnected CFs a transition is only possible through one or more other CFs. This topological information drastically reduces the number of possible next CFs; many of the transition probabilities in (14) are zero, and hence (13) contains fewer non-zero terms.

The accurate prediction of CF transitions is important for the filter's performance, since the number of particles needed for a given uncertainty on the geometric parameters is directly linked to the quality of the prediction step.

C. Measurement Model – Correction

The correction step uses the measurement model to calculate the hybrid joint density at time step k, given the prediction (13) for the hybrid joint density at time step k:

$$P\left(\boldsymbol{\theta}, \mathcal{CF}_k = j \mid \boldsymbol{z}_{1\ldots k}\right) \propto P\left(\boldsymbol{z}_k \mid \boldsymbol{\theta}, \mathcal{CF}_k = j\right) P\left(\boldsymbol{\theta}, \mathcal{CF}_k = j \mid \boldsymbol{z}_{1\ldots k-1}\right). \tag{15}$$

The measurement model represents the belief in a measurement $\boldsymbol{z}_k$, given the geometric parameters $\boldsymbol{\theta}$ and the $\mathcal{CF}_k = j$, and is defined by:

$$P\left(\boldsymbol{z}_k \mid \boldsymbol{\theta}, \mathcal{CF}_k = j\right). \tag{16}$$

In this research, two different measurement models are used to update the hybrid PDF. The first model is based on the pose measurement $\boldsymbol{X}_m$ in (9), and the second model is based on the wrench and twist measurements $\boldsymbol{w}_m$ in (2) and $\boldsymbol{t}_m$ in (10). Both measurements models are applied in every correction step.

1) Contact distance measurement model based on pose measurements: The contact distance measurement model expresses that when the manipulated object is in contact with the environmental object, the distance between the objects at the contact points should be zero, thereby closing the kinematic chain between the objects. The distance between the objects at non-contact points should be greater than zero, expressing that the objects do not penetrate nor contact.

The decomposition of a general CF into ECs (Section II) allows the automatic generation of the contact distance measurement equation for different CF models [32]. Say the number of possible ECs between the manipulated object and the environmental object at *all possible* CFs is p. This number is a constant that only depends on the geometry of the two contacting objects, not on the current CF. Each CF in the contact state graph can be decomposed into one or more ECs, therefore, the total number of possible ECs is ever greater than the number of CFs, and typically in the range of 10^2 to 10^3. At each $EC_1 \ldots EC_p$ the distance between the objects is calculated. A new measurement variable is created by combining all these distances into one distance vector:

$$\boldsymbol{d}_m = \left[\begin{array}{ccc} d_1 & \ldots & d_p \end{array}\right]^T. \tag{17}$$

This new distance measurement variable is a nonlinear function of the pose measurement $\boldsymbol{X}_m$ and the geometric parameters $\boldsymbol{\Theta}$. The contact distance measurement equation expresses the belief in the distance vector, given the current state $\mathcal{CF}_k = j$ and the geometric parameters $\boldsymbol{\theta}$:

$$P\left(\boldsymbol{d}_m \mid \boldsymbol{\theta}, \mathcal{CF}_k = j\right). \tag{18}$$

2) Residue measurement model based on wrench-twist measurements: The residue measurement model expresses the consistency between the contact constraints, and the wrench and twist measurements $\boldsymbol{w}_m$ and $\boldsymbol{t}_m$ [33], [34]. The consistency is expressed by a residue vector $\boldsymbol{r}_m$, which is the part of the measured twist and wrench that is not explained by the first order kinematics of an ideal frictionless contact; it should vanish when the measurements and the model are consistent. For a given pose and CF, the first order kinematics are represented by a wrench space $\mathcal{W}$ and a twist space $\mathcal{T}$. The wrench space contains all possible wrenches that can be applied between the contacting objects at the current pose, and is represented by a matrix $\boldsymbol{W}$ which spans the wrench space. The twist space contains all possible instantaneous twists that maintain the contact, and is represented by a matrix $\boldsymbol{T}$ which spans the twist space. The consistency between $\boldsymbol{w}_m$, $\boldsymbol{t}_m$ and $\mathcal{W}$, $\mathcal{T}$, is expressed by the 12-dimensional residue vector $\boldsymbol{r}_m$, containing a six-dimensional wrench residue and a six-dimensional twist residue:

$$\boldsymbol{r}_m = \left[\begin{array}{cc} \boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^{\dagger \boldsymbol{K}_w} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} - \boldsymbol{T}\boldsymbol{T}^{\dagger \boldsymbol{K}_t} \end{array}\right] \left[\begin{array}{c} \boldsymbol{w}_m \\ \boldsymbol{t}_m \end{array}\right]. \tag{19}$$

The operator ${}^{\dagger \boldsymbol{K}_w}$ represents the weighted pseudo-inverse [35], [36] of a matrix using a positive definite weighting matrix $\boldsymbol{K}_w = \boldsymbol{L}_w^T \boldsymbol{L}_w$, and is defined by:

$$\boldsymbol{W}^{\dagger \boldsymbol{K}_w} = \left(\boldsymbol{L}_w^T \boldsymbol{W}\right)^{\dagger} \boldsymbol{L}_w. \tag{20}$$

The residue $\boldsymbol{r}_m$ contains the difference between the measured wrench (twist) and its projection in the wrench (twist) space. It is a nonlinear function of the pose $\boldsymbol{X}_m$, the CF, the geometrical parameters $\boldsymbol{\Theta}$, and the measured wrench $\boldsymbol{w}_m$ and twist $\boldsymbol{t}_m$. The residue measurement equation expresses the belief in the residue vector, given the current state $\mathcal{CF}_k = j$ and the geometric parameters $\boldsymbol{\theta}$:

$$P\left(\boldsymbol{r}_m \mid \boldsymbol{\theta}, \mathcal{CF}_k = j\right). \tag{21}$$

V. Implementation Details

This section describes the implementation details of the system model in (14) and the measurement models in (18) and (21), presented in the previous section. The presented implementation is capable of processing $90,000$ particles[4] per second, on a 2 $[GHz]$ AMD 64 laptop, sufficient for realtime discrimination between 245 CFs and estimation of uncertain geometrical parameters. This performance is achieved by exploiting application-specific knowledge and using application-specific "shortcuts", such as:

- assuming probabilities to be independent,
- not calculating or only approximating probabilities that are not relevant, such as the probabilities of contact distances at ECs not adjacent to the current CF,
- developing numerically efficient algorithms that take advantage of the orthonormal nature of the matrices obtained from a singular value decomposition, and
- choosing easy to evaluate PDFs based on normal distributions and uniform distributions.

The particle filter implementation uses a 13-dimensional hybrid joint density PDF, consisting of a 12-dimensional continuous parameter and one discrete state. The continuous parameter $\boldsymbol{\Theta}$ contains geometric parameters that represent the unknown pose of the environmental object relative to a world reference, and the unknown pose of the manipulated object relative to the demonstration tool. The geometry of both the environmental object and the manipulated object is known. The discrete state $\mathcal{CF}$ contains the CF between the manipulated object and the environment, and can be any of the many hundreds of CFs in a complete contact state graph of the contacting objects.

A. Contact Distance Measurement Equation

The pose measurement model in (18) expresses the belief in the distance vector $\boldsymbol{d}_m$, which contains the distances between the involved objects at all possible ECs. Calculating all these distances for each pose between the objects would be numerically expensive. Therefore, only the distances at ECs that are relevant in the assumed CF[5] are calculated. The relevant distances are the distances at the ECs of the assumed CF, as well as the ECs directly connected to the assumed CF by an edge. When in the example in Fig. 8 the assumed CF (say CF_a) only includes EC_2, the distances $d_1 \ldots d_3$ are calculated, and not the distances $d_4 \ldots d_6$. When the assumed CF (say CF_b) includes EC_2 as well as EC_4, the distances $d_1 \ldots d_5$ are calculated, and not the distance d_6. In the former case, distance d_4 is not calculated, and thus the probability on CF_a does not decrease when d_4 gets smaller. However, when d_4 gets smaller, the probability on CF_b (which is a neighbor of CF_a) increases, and CF_b still becomes more probable than CF_a, although d_4 is not calculated in the case of CF_a.

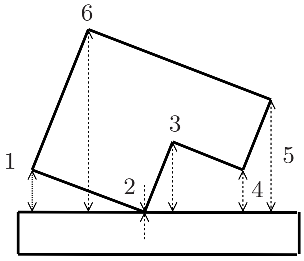

Fig. 8. For the calculation of the distance vector $\boldsymbol{d}_m$ only the distances at the ECs of assumed CF and the ECs that are directly connected to the assumed CF by an edge, are calculated.

The probabilities of the distances $d_1 \ldots d_p$ are not independent, but are all linked by the probability of the measured pose $\boldsymbol{X}_m$. The nonlinear relation between the probabilities of the pose and the distances is expensive to calculate, and therefore the distances d_i are assumed to be independent. The probability of the distance vector, as expressed in (18), can then be calculated by:

$$P\left(\boldsymbol{d}_m \mid \boldsymbol{\theta}, \mathcal{CF}_k = j\right) = \prod_{i=1}^{p} P\left(d_i \mid \boldsymbol{\theta}, \mathcal{CF}_k = j\right), \tag{22}$$

where $P\left(d_i \mid \boldsymbol{\theta}, \mathcal{CF}_k = j\right)$ expresses the belief in a distance measurement at the i^{th} EC.

The distance calculation at the ECs is helped by the use of spherical boundary boxes around the elements of the ECs. Only when two boundary boxes intersect, the exact distance at an EC is calculated. When the boundary boxes do not intersect (meaning that the elements of the EC are far apart) the exact distance has no real influence on the filter's behavior, and is approximated by the distance between the boundary boxes.

In this paper, the PDF of a distance d_i at an EC that is part of the current CF, is shown by the dashed line in Fig. 9, while the PDF for a distance d_i at an EC that is not part of the current PC, is shown by the continuous line.

B. Residue Measurement Equation

The residue measurement model in (21) expresses the belief in the residue vector $\boldsymbol{r}_m$, which is a measure for the consistency between the first order kinematics of the contact model, and the measured twist and wrench. This section presents an efficient approach to calculate the residue vector. It exploits the orthonormal nature of the matrices obtained from a *singular value decomposition* (SVD), to replace two weighted pseudo-inverses by two more efficient transposes, for a specific choice of the weighting matrices.

[4] Processing one particle includes one system update, two measurement updates and the overhead of the particle filter such as re-sampling.

[5] The assumed CF is the CF that is given in (18), the $\mathcal{CF}_k = j$

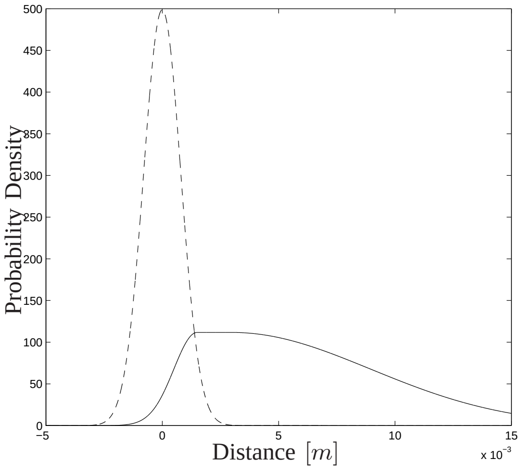

Fig. 9. The probability density function on the distance at an EC between two objects in contact (dashed line) and at an EC between two objects not in contact (continuous line).

1) Twist and Wrench Space: The first order kinematics of an ideal frictionless contact are represented by the local wrench and twist space at the CF. The wrench space of a CF is calculated by taking the union of the wrench spaces at each individual EC of the CF. The wrench space at an EC is always one-dimensional and defined by a wrench vector $\boldsymbol{w}_{EC}$ with zero torque when expressed at the EC itself. For a CF with q ECs, the wrench space of the CF is represented by:

$$\boldsymbol{W}_{CF} = \left[\begin{array}{ccc} \boldsymbol{w}_{EC_1} & \ldots & \boldsymbol{w}_{EC_q} \end{array} \right]. \qquad (23)$$

Using the SVD of this representation of the wrench space $\boldsymbol{W}_{CF}$, a base $\boldsymbol{W}$ and a base $\boldsymbol{T}$ for its dual space are obtained [10]:

$$\boldsymbol{W}_{CF} = [\boldsymbol{W} \quad \boldsymbol{T}]_{6\times 6}\, \boldsymbol{S}_{6\times q} \boldsymbol{V}^T_{q\times q}, \qquad (24)$$

in which the diagonal elements of $\boldsymbol{S}$ are the singular values, and the matrix $\boldsymbol{V}$ is orthonormal. The matrix $[\boldsymbol{W} \quad \boldsymbol{T}]$ is also orthonormal, however, all columns of the wrench space should be reciprocal to all columns of the twist space [37], meaning that any possible twist $\boldsymbol{t}$ of $\boldsymbol{T}$ produces no work in the interaction with any possible wrench $\boldsymbol{w}$ of $\boldsymbol{W}$:

$$\boldsymbol{W}^T \boldsymbol{T} = \boldsymbol{0}. \qquad (25)$$

The notion of orthogonality is often used to interpret the reciprocity condition, but is not applicable because the orthogonality can only be defined between elements of the same space, and twist and wrench spaces are distinct vector spaces [38]. To interpret the orthogonal columns of $[\boldsymbol{W} \quad \boldsymbol{T}]$ as reciprocal wrenches and twists, we assign compatible units to forces, torques, rotational velocities and translational velocities.

2) Implementation with a single SVD: The calculation of the residue $\boldsymbol{r}_m$ in (19) includes one SVD of the wrench space $\boldsymbol{W}_{\mathcal{CF}}$ (24), and two weighted pseudo-inverses $\boldsymbol{W}^{\dagger \boldsymbol{K}_w}$ and $\boldsymbol{T}^{\dagger \boldsymbol{K}_t}$ that each require another SVD. These calculations are numerically expensive. However, a smart choice of the weighting matrices $\boldsymbol{K}_w$ and $\boldsymbol{K}_t$ reduces the numerical cost to a single SVD, using the following method.

Choosing a diagonal weighting matrix $\boldsymbol{K}_w$, the product of a wrench space with this weighting matrix corresponds to a change of units[6] of the wrench space:

$$\boldsymbol{W}'_{CF} = \boldsymbol{K}_w \boldsymbol{W}_{CF}, \qquad (26)$$

in which $\boldsymbol{W}'_{CF}$ represents the same wrench space as $\boldsymbol{W}_{CF}$, using different units. The reciprocity condition in (25) imposes that when the units of the wrench space change with $\boldsymbol{K}_w$ between $\boldsymbol{W}$ and $\boldsymbol{W}'$, the units of the twist space change with $\boldsymbol{K}_w^{-1}$ between $\boldsymbol{T}$ and $\boldsymbol{T}'$:

$$\boldsymbol{W}'^T \boldsymbol{T}' = (\boldsymbol{K}_w \boldsymbol{W})^T (\boldsymbol{K}_w^{-1} \boldsymbol{T}) = \boldsymbol{0}. \qquad (27)$$

Therefore, when choosing the weighting matrices:

$$\boldsymbol{K}_t = \boldsymbol{K}_w^{-1}, \qquad (28)$$

the SVD of $\boldsymbol{W}'_{CF}$ instead of (24), results in a wrench space $\boldsymbol{W}'$ weighted with $\boldsymbol{K}_w$ and a twist space $\boldsymbol{T}'$ weighted with $\boldsymbol{K}_w^{-1}$:

$$\boldsymbol{W}'_{CF} = \left[\boldsymbol{W}' \quad \boldsymbol{T}' \right] \boldsymbol{S}' \boldsymbol{V}'^T, \qquad (29)$$

The calculation of the residue vector is independent of the bases used to represent the wrench and twist space. However, choosing the numerically orthonormal matrices $\boldsymbol{W}'$ and $\boldsymbol{T}'$ to represent the wrench and twist space, the pseudo-inverse of the wrench and twist space is reduced to a transpose:

$$\begin{aligned} \boldsymbol{W}'^{\dagger} &= \boldsymbol{W}'^T, \\ \boldsymbol{T}'^{\dagger} &= \boldsymbol{T}'^T. \end{aligned} \qquad (30)$$

This results in the calculation of the residue in (19) with a numerical cost of only one SVD in (29):

$$\boldsymbol{r}_m = \boldsymbol{K} \left[\begin{array}{cc} \boldsymbol{I} - \boldsymbol{W}'\boldsymbol{W}'^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} - \boldsymbol{T}'\boldsymbol{T}'^T \end{array} \right] \boldsymbol{K}^{-1} \left[\begin{array}{c} \boldsymbol{w}_m \\ \boldsymbol{t}_m \end{array} \right], \qquad (31)$$

in which

$$\boldsymbol{K} = \left[\begin{array}{cc} \boldsymbol{K}_w^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_w \end{array} \right]. \qquad (32)$$

The importance of reducing the number of SVDs required becomes clear when using profiling tools to measure the computational time spent on each of the sub-algorithms. Reducing the number of SVDs from 3 to 1 decreased the computational cost of the overall filter with 55%. Using the efficient algorithm still 60% of the overall computation time is spent on this one SVD.

In this paper, the PDF of the residue $\boldsymbol{r}_m$ is represented by two 6-dimensional Gaussians, one for the wrench residue and one for the twist residue.

[6] Example of changing the units of a wrench vector using a diagonal weighting matrix:

$$\left[\begin{array}{c} \boldsymbol{w}_1[kN] \\ \boldsymbol{w}_2[mN] \\ \boldsymbol{w}_3[N] \\ \boldsymbol{w}_4[Ndm] \\ \boldsymbol{w}_5[Nkm] \\ \boldsymbol{w}_6[Nm] \end{array} \right] = \left[\begin{array}{cccccc} 0.001 & & & & & \\ & 1000 & & & & \\ & & 1 & & & \\ & & & 10 & & \\ & & & & 0.001 & \\ & & & & & 1 \end{array} \right] \left[\begin{array}{c} \boldsymbol{w}_1[N] \\ \boldsymbol{w}_2[N] \\ \boldsymbol{w}_3[N] \\ \boldsymbol{w}_4[Nm] \\ \boldsymbol{w}_5[Nm] \\ \boldsymbol{w}_6[Nm] \end{array} \right]$$

C. System Update

The system update calculates the prediction density at time step k, given the estimated variables at time step $k-1$, as described by equation (13). This prediction step is based on the probability of a transition from a $\mathcal{CF}_{k-1} = i$ to a $\mathcal{CF}_k = j$. The adjacency relationship between CFs, which is defined by arcs connecting CFs in the contact state graph, indicates if a direct transition between two CFs is possible without passing through any other CF (see Fig. 4). If two CFs i and j are not adjacent, the probability of a transition is defined by:

$$P\left(\mathcal{CF}_k = j \mid \boldsymbol{\theta}, \mathcal{CF}_{k-1} = i\right) = 0. \tag{33}$$

For two adjacent CFs i and j, the probability of a transition is defined by the distance vector $\boldsymbol{d}_m$ between the two objects, as defined in equation (17). The closer the two objects are, the more likely a transition will occur:

$$P\left(\mathcal{CF}_k = j \mid \boldsymbol{\theta}, \mathcal{CF}_{k-1} = i\right) = P\left(\boldsymbol{d}_m \mid \boldsymbol{\theta}, \mathcal{CF}_k = j\right) \tag{34}$$

The probability is calculated identical to the contact distance probability in (18). Because the probability of a CF transition only depends on the involved CFs and the pose measurement, the approach does not require a separate training phase to assign probabilities to CF transitions.

D. Software

The algorithms presented in this paper are implemented within the framework offered by the open source C++ Bayesian Filtering Library (BFL) [39]. BFL offers a unifying framework for all recursive Bayesian filters, such as Kalman filters, extended Kalman filters and particle filters. It provides efficient implementations of various filter algorithms.

The software for the automatic generation of a contact state graph, given a geometric description of two polyhedral objects, is provided by Jing Xiao of the University of North Carolina at Charlotte and her research group. Their algorithms generate a complete contact state graph containing hundreds of CFs, within seconds.

VI. Experiments

This section reports on the real world experiment to validate the presented approach. In the experiment, a human demonstrator manipulates a cube through a complex sequence of CFs in an environment consisting of three perpendicular faces. Fig. 10 shows the experimental setup and Fig. 11 shows the sequence of CFs.

During the demonstration, sensors mounted on the demonstration tool measure its pose, twist and contact wrench. The initial uncertainty on the 12-dimensional continuous geometric parameters, which are the poses of the environmental and manipulated objects, is represented by uniform distributions. The uniform distribution on the pose of the environmental object relative to a world reference, has a width of 15 $[mm]$ on the x and y positions, 130 $[mm]$ on the z position and 0.5 $[rad]$ on the roll-pitch-yaw orientation. The uniform distribution on the pose of the manipulated object relative to the demonstration tool, has has a width of 5 $[mm]$ on the x, y and z positions and 0.5 $[rad]$ on the roll-pitch-yaw orientation. The discrete state can be any of the 245 possible CFs between the cube and the planes, and there is initially no contact between the objects.

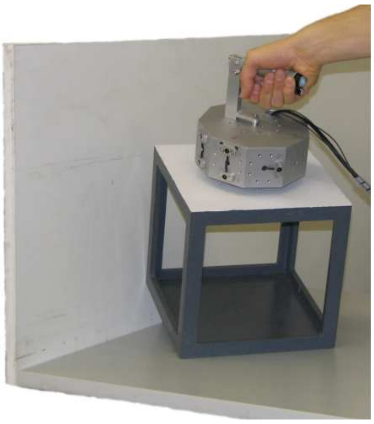

Fig. 10. In the experiment to validate the presented approach a cube is manipulated in contact with three perpendicular faces.

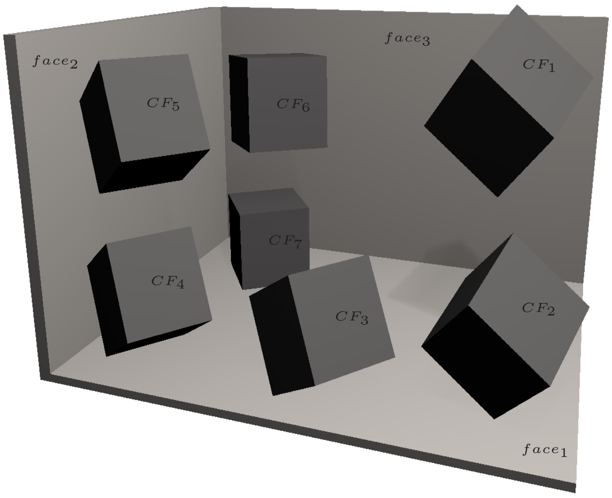

Fig. 11. The CF evolution of a human demonstration where a cube is manipulated in contact with two perpendicular faces.

The Gaussian PDF on the wrench residue has a sigma boundary of 10.0 $[N]$ for the forces and 1.0 $[Nm]$ for the torques, while the Gaussian PDF on the twist residue has a sigma boundary of 0.001 $[m/s]$ for the translational velocity and 0.01 $[rad/s]$ for the rotational velocities. The Gaussian PDF on the distance at an EC of the current CF a sigma boundary of 0.005 $[m]$. The PDF on the distance at an EC that is not part of the current CF is the only one with a non-zero expected value of $0.1[m]$. The wrench and twist weighting matrices are chosen:

$$\boldsymbol{K}_w = \boldsymbol{K}_t^{-1} = \begin{bmatrix} 1 & & & & & \\ & 1 & & & \mathbf{0} & \\ & & 1 & & & \\ & & & 10 & & \\ & \mathbf{0} & & & 10 & \\ & & & & & 10 \end{bmatrix}. \qquad (35)$$

Based on multiple experiments it was shown that 20.000 particles are sufficient to obtain a robust recognition of the CFs. Although for a single experiment it is sometimes possible to lower the number of particles to as little as 5000 and still obtain the same CF segmentation, at least 20.000 particles are needed to ensure that the CF recognition for all similar experiments with a comparable uncertainty is successful. The joint posterior PDF is dynamically re-sampled using importance sampling, once the effective number of particles drops below 12.000.

A. Approach to First Contact

In the first part of the experiment, the cube has no contact with the environment, and approaches one of the planes of the environment. Fig. 12 shows the time evolution of the uncertainty on the z position of the environment. The uncertainty on this position, represented by a histogram, is 1 component of the 12-dimensional continuous parameter; it is obtained by integrating[7] over the 11 other components, for a given CF. The first five sub-figures show the position parameter for the no-contact CF, while the sixth sub-figure shows the position parameter for the first vertex-face CF. Initially the position is represented by a uniform distribution, indicating that there is little knowledge about its value. When the cube approaches the plane, the probability decreases on the left side of the distribution. This shows that the cube "penetrated" one of the possible positions of the plane without detecting a contact, thus proving that possible position invalid. This evolution continues until the cube makes a vertex-face CF with the plane. The CF transition is detected due to the inconsistency between the measured wrench and the assumed no-contact CF. The knowledge about the vertex-face CF allows accurate estimation of the position of the plane, also decreasing the probability on the right side of the uniform distribution. When the probability density decreases on one side of the distribution, the probability density increases for the rest of the distribution, keeping the total probability unchanged.

B. Sequence of Contact Formations

In the rest of the experiment, the cube is manipulated in contact with the environment, through a complex sequence of CFs. The demonstrated sequence of CFs and the dimension of the wrench and twist spaces is shown in Table I and Fig. 11.

The experiment includes complex CFs and CF transitions, such as:

- adding contact constraints (CF_1 to CF_2 to CF_3),
- removing contact constraints (CF_5 to CF_6),
- adding many contact constraints at once (CF_6 to CF_7),
- removing many contact constraints at once (CF_4 to CF_5), and
- simultaneous contacts with the two planes (CF_4).

Fig. 13 shows the evolution of the estimated probability on each of the 245 possible CFs. Each CF is one possible value of the discrete state, and its probability is obtained by integrating over all 12 components of the continuous parameter, for each

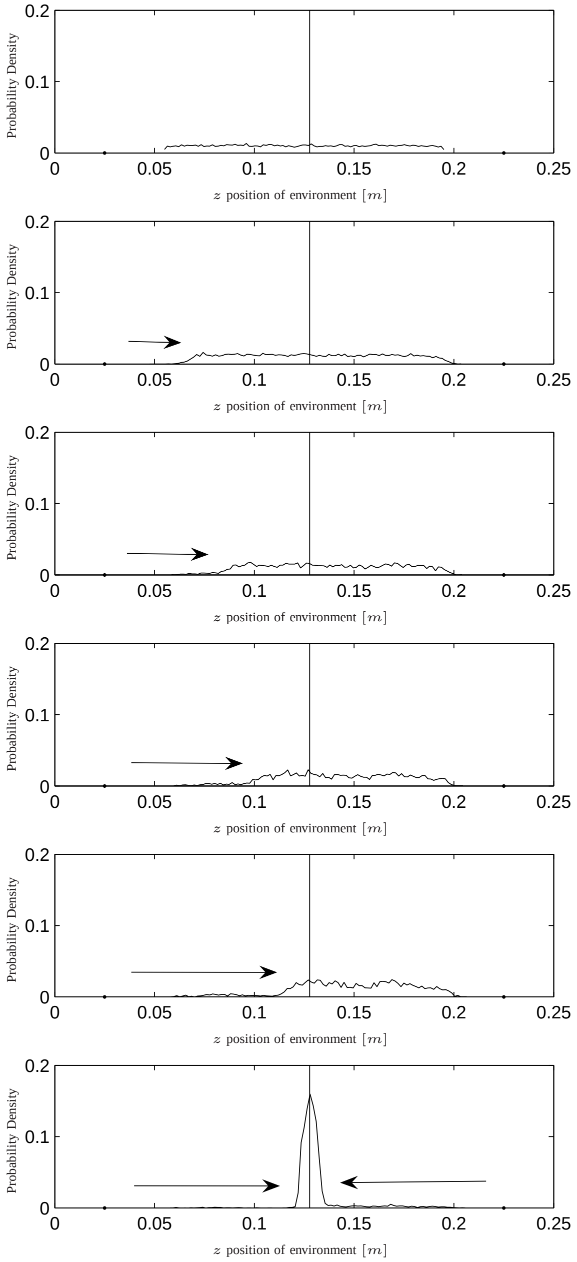

Fig. 12. The time evolution of the probability density on the position of a plane, when approaching the plane with a cube. The vertical line shows the true position of the plane. The probability density decreases gradually when the cube approaches. The last figure shows how the probability density suddenly decreases when the cube makes contact with the plane.

[7] The hybrid PDF is represented by discrete samples; the integral over the parameters of the PDF is approximated by the summation over the discrete samples.

TABLE I

THE SEQUENCE OF CFS BETWEEN THE CUBE AND ITS ENVIRONMENT.

CF	Principal Contacts	Wrench	Twist
CF_1	no contact	0	6
CF_2	face1-vertex1	1	5
CF_3	face1-edge1	2	4
CF_4	face1-edge1 face2-vertex2	3	3
CF_5	face2-vertex2	1	5
CF_6	no contact	0	6
CF_7	face1-face3	3	3
CF_8	face2-edge3	3	3
CF_9	face1-edge4	3	3

of the possible values of the discrete state. At each time step only a few CFs have a probability greater than zero. The particle filters successfully assign the highest probability to the CF that corresponds to the true CF in the experiment. Notice that CF_8 and CF_9 have a relevant probability in the respective measurement intervals $[120-150]$ and $[184-186]$. Both these CFs are neighboring to the true CF in the measurement interval.

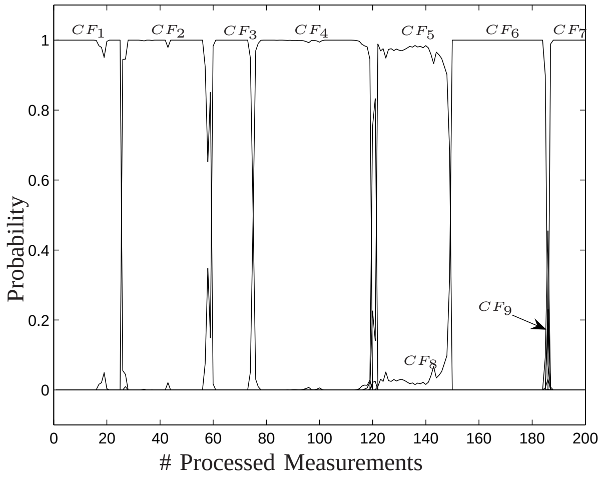

Fig. 13. The CF evolution of a human demonstration where a cube is manipulated in contact with two perpendicular faces. The evolution is shown by the probability on each of the CFs. The sequence is listed in Table I.

C. Comparison with previous approach

The effectiveness of the presented approach is shown when comparing the obtained experimental results with the results previously obtained by Gadeyne et al. in [21]. The main improvements with respect to the appraoch in [21] are found in the number of particles needed to estimate the hybrid state, and the performance of the presented algorithms.

- **Number of particles** The approach in [21] does not use the topological information of a contact state graph, and therefore requires to consider all 245 possible CFs at each timestep. The approach in this paper only requires to evaluate the *neighboring* CFs, which reduces the number of CFs to consider to an average of 5 CFs at each timestep. To allow the same uncertainty on the geometrical parameters, the approach in [21] would require the same number of particles per CF, resulting in 49 times more particles. For the presented experiment this would result in $980,000$ required particles.
- **Performance** The approach in [21] is able to process $3,000$ particles per second on a 1.1 $[GHz]$ laptop. Assuming this older hardware is a factor 4 slower than the 2 $[GHz]$ laptop used for the experiments in this paper, the presented approach is still 7.5 times faster in processing particles ($90,000$ particles per second compared to $4\times 3,000$ particles).

Combining both the decreased number of required particles and the increased performance, the presented approach is 367.5 times faster than the approach in [21], which for the presented experiment results in processing particles in realtime at 4.5 $[Hz]$ instead of at $\frac{1}{82}$ $[Hz]$. In addition to this increased processing speed, the presented appraoch considers the geometric features of the object (faces and edges) to be finite, and applies an improved measurement model (see Section VII-C).

VII. DISCUSSION

A. Contact State Graph

The contact state graph contains the adjacency relation between CFs. Two CFs are adjacent when a direct transition between them is possible, without going through any other CF. The system model directly applies this adjacency relation by only allowing a CF transition between adjacent CFs in (33). This knowledge about the adjacency relation drastically reduces the search field for possible next CFs. However, one could imagine a situation where the intermediate CF between two not-adjacent (but almost adjacent) CFs is skipped. In the example in Fig. 14, this could be a jump between CF_1 and CF_3, without going through CF_2. The system model will not be able to directly recognize the new CF, since it is not adjacent to the current CF. However, if the path in the contact state graph between the current and the new CF has a constantly increasing or decreasing number of contact constraints, the estimation will converge to the new CF. In this case, given the pose wrench and twist measurements at the new CF, each intermediate CF on the path between the current and new CF will be more and more probable, guiding the estimation to the new CF. When this is not the case and the

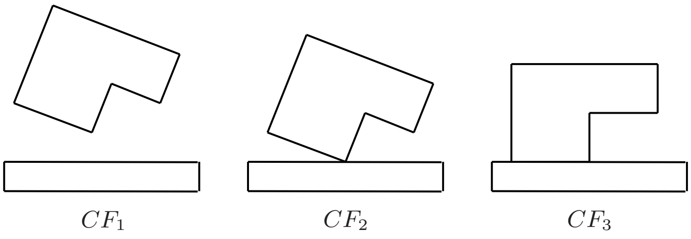

Fig. 14. Moving directly from CF_1 to CF_3 corresponds to a jump in the contact state graph. However, the estimated CF will converge to CF_3 by first passing through CF_2.

jump in the contact state graph is too large, the localization in the contact state graph is lost, encountering the “kidnapped robot” problem [40]. Adding random possible CFs to the particle filters can help to cope with this problem.

B. Probability Density Functions

The PDFs on the distance vector in (18) and the residue vector in (21) indirectly define the belief in the pose, wrench and twist measurements. The choice of these PDFs is always somewhat arbitrary, however, it is possible to make a problem specific but physically founded choice, considering the accuracy of (i) the physical sensors used, (ii) the calibration of the sensors, and (iii) the models used to describe the real world. The PDF on the wrench residue is dominated by the inaccuracy of the contact model that does not consider friction nor inertia forces. The PDFs on the twist residue and the contact distance are mainly influenced by the inaccuracy on the calibration of the demonstration tool, and the deformations of the demonstration tool. The actual values of the PDFs are given in Section VI.

Applying a model that differs from the real world not only affects the chosen uncertainty on the sensor measurements, but also the number of particles needed for the filter, and hence the performance of the filter. Using the presented method when manipulating e.g. a flexible object will require more particles then when manipulating e.g. a rigid object, because a rigid object better matches the used contact model.

C. Consistency versus Reciprocity

To link the wrench and twist measurements to the geometric parameters in a given CF, two different models where previously presented in literature: the consistency model (used in this paper) and the reciprocity model [33], [34].

The reciprocity model expresses that a measured twist (wrench) produces no work against the wrench (twist) vectors of the wrench (twist) space, and is expressed by the 6-dimensional power vector:

$$\boldsymbol{e}_m = \begin{bmatrix} \boldsymbol{T} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{W} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_m \\ \boldsymbol{t}_m \end{bmatrix}. \tag{36}$$

The power vector contains n components resulting from $\boldsymbol{T} \cdot \boldsymbol{w}_m$ and $(6-n)$ components from $\boldsymbol{W} \cdot \boldsymbol{t}_m$, with n the dimension of the twist space. The reciprocity model falls short when comparing two models with two different CFs, when one CF has a higher dimensional twist space than the other CF. For example, when the measured twist and wrench are zero, and the measured twist has a lower (higher) uncertainty than the measured wrench, the reciprocity model will always prefer a CF with the higher dimensional twist (wrench) space.

The consistency model, which expresses that the measured twist (wrench) is a linear combination of the twist (wrench) base vectors of the twist (wrench) space, does not suffer from the same shortcoming. Its 12-dimensional residue vector in (19) always contains 6 components from the measured twist, and 6 components from the measured wrench, thus avoiding favoring a higher dimensional twist or wrench space. Therefore, while the reciprocity model is only appropriate to distinguish between two models in the same CF, the consistency model can also be used to distinguish between different CFs.

VIII. Conclusions

This paper presents a contribution to the task specification process for sensor-controlled robot systems that physically interact with the environment, with a focus on compliant motion tasks. As shown in Fig. 1, previous research uses an automated off-line compliant path planner to generate a compliant path. This compliant path is then automatically converted into a task specification for a hybrid controller, using the compliant task generator. This paper presents another approach to obtain a compliant path, using programming by human demonstration.

In the demonstration step, a human demonstrates a compliant task by directly manipulating an object in contact with its environment, using a demonstration tool. During the demonstration, the krypton 6D optical system measures the pose and twist of the manipulated object, while a wrench sensor measures the interaction forces between the contacting objects. Subsequently, in the interpretation step, these measurements are interpreted using sequential Bayesian estimation techniques.

Similar to the more familiar localization, tracking and recognition problems in mobile robotics, the approach *simultaneously* recognizes CF transitions and estimates geometrical parameters. The approach is based on the Bayesian sequential Monte Carlo method, or particle filter, and can cope with hybrid (partly discrete, partly continuous) joint posterior variables containing both the CF and geometrical parameters of the environment. While previously presented results in this field only allowed certain CFs or certain CF transitions, this approach scales the search space to *all possible CFs* between the contacting objects, using the *topological information* contained in a contact state graph. This extension in combination with new efficient algorithms, allow the *realtime* simultaneous recognition of CFs and estimation of geometric parameters. The approach applies to convex as well as to concave polyhedral objects with a known geometry, but at an unknown pose, and is able to efficiently recognize the CF at each step of a human demonstration out of many hundreds of possible CFs. Experimental results verify the effectiveness of the method. While in [13] the probability of CF transitions is trained, the experimental results presented in this paper do not use any prior knowledge about the probability of CF transitions. Training the CF transitions in the contact state graph using sensor data gathered during multiple demonstrations is an interesting topic for future research.

In this paper the presented particle filters are applied to sensor data gathered during a human demonstration. However, the estimators can also be used to process sensor data collected during the execution of a compliant motion task. When processing this data online, during the execution, the estimators can provide feedback to the robot controller about the current CF and estimate geometrical parameters related to the robot manipulator and its environment. The performance of the presented algorithms is sufficient to process all sensor data in realtime. Using the estimators online to provide feedback for the robot controller will be the subject of further research and experiments.

ACKNOWLEDGMENT

All authors gratefully acknowledge the financial support by K.U.Leuven's Concerted Research Action GOA/05/10. The authors also acknowledge Jing Xiao of the university of North Carolina at Charlotte and her research group for providing their software for the automatic generation of a contact state graph. The authors also thank all the reviewers for their time and their constructive comments.

REFERENCES

[1] J. De Schutter and H. Van Brussel, "Compliant Motion I, II," *Int. J. Robotics Research*, vol. 7, no. 4, pp. 3–33, Aug 1988.

[2] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-11, no. 6, pp. 418–432, 1981.

[3] H. Bruyninckx and J. De Schutter, "Specification of force-controlled actions in the "Task Frame Formalism": A survey," *IEEE Trans. Rob. Automation*, vol. 12, no. 5, pp. 581–589, 1996.

[4] J. Chen and A. Zelinsky, "Programing by demonstration: Coping with suboptimal teaching actions," *Int. J. Robotics Research*, vol. 22, no. 5, pp. 299–319, 2003.

[5] Q. Wang, J. De Schutter, W. Witvrouw, and S. Graves, "Derivation of compliant motion programs based on human demonstration," in *Int. Conf. Robotics and Automation*, Minneapolis, MN, 1996, pp. 2616–2621.

[6] T. Kröger, B. Finkemeyer, and F. M. Wahl, "A task frame formalism for practical implementations," in *Int. Conf. Robotics and Automation*, New Orleans, U.S.A., 2004, pp. 5218–5223.

[7] T. Kröger, B. Finkemeyer, M. Heuck, and F. M. Wahl, "Adaptive implicit hybrid force/pose control of industrial manipulators: Compliant motion experiments," in *Proc. IEEE/RSJ Int. Conf. Int. Robots and Systems*, Sendai, Japan, 2004, pp. 816–821.

[8] J. De Schutter, J. Rutgeerts, E. Aertbelien, F. De Groote, T. De Laet, T. Lefebvre, W. Verdonck, and H. Bruyninckx, "Unified constraint-based task specification for complex sensor-based robot systems," in *Int. Conf. Robotics and Automation*, Barcelona, Spain, 2005, pp. 3618–3623.

[9] X. Ji and J. Xiao, "Planning motions compliant to complex contact states," *Int. J. Robotics Research*, vol. 20, no. 6, pp. 446–465, July 2001.

[10] W. Meeussen, J. De Schutter, H. Bruyninckx, J. Xiao, and E. Staffetti, "Integration of planning and execution in force controlled compliant motion," in *Proc. IEEE/RSJ Int. Conf. Int. Robots and Systems*, Edmonton, Canada, 2005, pp. 2550–2555.

[11] A. Bicchi, K. Salisbury, and D. L. Brock, "Contact sensing from force measurements," *Int. J. Robotics Research*, vol. 12, no. 3, pp. 249–262, 1993.

[12] A. O. Farahat, B. S. Graves, and J. C. Trinkle, "Identifying contact formations in the presence of uncertainty," in *Proc. IEEE/RSJ Int. Conf. Int. Robots and Systems*, Pittsburgh, PA, 1995.

[13] B. S. Eberman and J. K. Salisbury, Jr., "Application of change detection to dynamic contact sensing," *Int. J. Robotics Research*, vol. 13, no. 5, pp. 369–394, 1994.

[14] B. J. McCarragher and H. Asada, "Qualitative template matching using dynamic process models for state transition recognition of robotic assembly," *Trans. ASME J. Dyn. Systems Meas. Control*, vol. 115, pp. 261–269, 1993.

[15] D. Schulz, W. Burgard, and D. Fox, "People tracking with mobile robots using sample-based joint probabilistic data association filters," *Int. J. Robotics Research*, vol. 22, no. 2, pp. 99–116, 2003.

[16] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[17] A. J. Davison, Y. G. Cid, and N. Kita, "Real-time 3D SLAM with wide-angle vision," in *Proc. 5th IFAC Symposium on Intelligent Autonomous Vehicles*, 2004.

[18] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using fastslam," in *Int. Conf. Robotics and Automation*, Taipeh, Taiwan, 2003, pp. 1985–1991.

[19] P. Slaets, T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Incremental building of a polyhedral feature model for programming by human demonstration of force controlled tasks," *Int. J. Robotics Research*, 2006.

[20] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Exact nonlinear Bayesian parameter estimation for autonomous compliant motion," *Advanced Robotics*, vol. 18, no. 8, pp. 787–800, 2004.

[21] K. Gadeyne, T. Lefebvre, and H. Bruyninckx, "Bayesian hybrid model-state estimation applied to simultaneous contact formation recognition and geometrical parameter estimation," *Int. J. Robotics Research*, vol. 24, no. 8, pp. 615–630, 2005.

[22] A. Doucet, N. J. Gordon, and V. Krishnamurthy, "Particle Filters for State Estimation of Jump Markov Linear Systems," *IEEE Trans. Signal Processing*, vol. 49, no. 3, pp. 613–624, march 2001.

[23] J. Xiao and X. Ji, "A divide-and-merge approach to automatic generation of contact states and planning of contact motions," in *Int. Conf. Robotics and Automation*, San Francisco, CA, 2000, pp. 750–756.

[24] ——, "On automatic generation of high-level contact state space," *Int. J. Robotics Research*, vol. 20, no. 7, pp. 584–606, 2001.

[25] J. Xiao, "Automatic determination of topological contacts in the presence of sensing uncertainty," in *Int. Conf. Robotics and Automation*, Atlanta, GA, 1993, pp. 65–70.

[26] N. J. Higham, "Computing the polar decomposition—with applications," *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 4, pp. 1160–1174, 1986.

[27] R. Penrose, "A generalized inverse for matrices," *Proc. Cambridge Philos. Soc.*, vol. 51, pp. 406–413, 1955.

[28] Y. Bar-Shalom and X. Li, *Estimation and Tracking, Principles, Techniques, and Software*. Artech House, 1993.

[29] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, vol. 82, pp. 34–45, 1960.

[30] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*. McGraw-Hill, 1996.

[31] J. Rutgeerts, P. Slaets, F. Schillebeeckx, W. Meeussen, B. Stallaert, P. Princen, T. Lefebvre, H. Bruyninckx, and J. De Schutter, "A demonstration tool with Kalman Filter data processing for robot programming by human demonstration," in *Proc. IEEE/RSJ Int. Conf. Int. Robots and Systems*, Edmonton, Canada, 2005, pp. 3918–3923.

[32] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Polyhedral contact formation modeling and identification for autonomous compliant motion," *IEEE Trans. Rob. Automation*, vol. 19, no. 1, pp. 26–41, 2003.

[33] H. Bruyninckx, J. De Schutter, and S. Dutré, "The "reciprocity" and "consistency" based approaches to uncertainty identification for compliant motions," in *Int. Conf. Robotics and Automation*, Atlanta, GA, 1993, pp. 349–354.

[34] M. S. Ohwovoriole and B. Roth, "An extension of screw theory," *Trans. ASME J. Mech. Design*, vol. 103, no. 4, pp. 725–735, 1981.

[35] K. L. Doty, C. Melchiorri, and C. Bonivento, "A theory of generalized inverses applied to robotics," *Int. J. Robotics Research*, vol. 12, no. 1, pp. 1–19, 1993.

[36] Y. Nakamura, *Advanced robotics: redundancy and optimization*. Reading, MA: Addison-Wesley, 1991.

[37] R. S. Ball, "The theory of screws—a geometrical study of the kinematics, equilibrium, and small oscillations of a rigid body," *Trans. of the Royal Irish Academy*, vol. 25, pp. 137–217, 1871.

[38] J. Duffy, "The fallacy of modern hybrid control theory that is based on "orthogonal complements" of twist and wrench spaces," *J. Robotic Systems*, vol. 7, no. 2, pp. 139–144, 1990.

[39] K. Gadeyne, "BFL: Bayesian Filtering Library," http://people.mech.kuleuven.ac.be/~kgadeyne/bfl, 2001.

[40] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo localization: Efficient position estimation for mobile robots," in *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99).*, July 1999.