Contact-State Classification in Human-Demonstrated Robot Compliant Motion Tasks Using the Boosting Algorithm

Stefano Cabras, María Eugenia Castellanos, and Ernesto Staffetti

***Abstract*—Robot programming by demonstration is a robot programming paradigm in which a human operator directly demonstrates the task to be performed. In this paper, we focus on programming by demonstration of compliant motion tasks, which are tasks that involve contacts between an object manipulated by the robot and the environment in which it operates. Critical issues in this paradigm are to distinguish essential actions from those that are not relevant for the correct execution of the task and to transform this information into a robot-independent representation. Essential actions in compliant motion tasks are the contacts that take place, and therefore, it is important to understand the sequence of contact states that occur during a demonstration, called contact classification or contact segmentation. We propose a contact classification algorithm based on a supervised learning algorithm, in particular on a stochastic gradient boosting algorithm. The approach described in this paper is accurate and does not depend on the geometric model of the objects involved in the demonstration. It neither relies on the kinestatic model of the contact interactions nor on the contact state graph, whose computation is usually of prohibitive complexity even for very simple geometric object models.**

***Index Terms*—Contact classification, contact segmentation, programming by human demonstration, stochastic gradient boosting (SGB), supervised learning.**

I. Introduction

MOST ROBOT programming languages work at a low level of abstraction and need a complete specification of the sensor-based strategy to perform a task. Some languages permit a higher level of abstraction and automatically generate a task plan based on a description of the crucial aspects of the task such as initial and final configurations, constraints on the path, on the twists of the manipulated object, or on the wrenches to be applied to the environment during the task. However, they need additional information such as the geometrical, kinematic, or dynamic models of the robot manipulator, of the manipulated objects, and of the environment in which the robot operates, which need to be specified as well. This means that, in general, these robot programming methods require specialist knowledge and, most importantly, they are time consuming.

Robot programming by demonstration is a paradigm for teaching a robot a task by demonstrating it directly. It consists of a set of techniques to generate robot task plans from user examples. In programming by demonstration systems, this skill transfer process from humans to robots relies on the following:

1) a real or virtual environment to acquire the crucial information about a task from one or more demonstrations;
2) a methodology to transform the acquired information into a robot-independent representation;
3) a strategy to convert this knowledge learned through observation into a sequence of robot actions that accomplish the same task.

Critical aspects of robot programming by demonstration are to capture the intention of the user and to distinguish essential actions of the task demonstration from actions that are not relevant for its correct execution. Knowledge of the redundant part of the task is important because it permits its optimization. In this paper, we focus on point 2) applied to robot programming by demonstration of compliant motion tasks. In [1], an approach related to point 3) to automatically convert a geometrical contact path of a polyhedral object into a force-based task specification for a hybrid robot controller has been presented.

Compliant motion tasks are manipulation tasks that involve contacts between the object manipulated by a robot and the environment in which it operates. In general, interaction wrenches are crucial for the correct execution of the task, and they have to be controlled along the task. Even if the interaction wrenches are not relevant for a particular compliant motion task, these operations must be carried out using passive or active force control to cope with uncertainties because small errors in the object models can give rise to high interaction forces. We suppose that objects involved in the compliant motion are rigid. In general, the surface of a rigid object can be represented as a set of surface patches, curves, and vertices in a 3-D space with their adjacency relationships. Each surface patch defines a face of the object, and the intersection curve between two faces is called an edge. A vertex of an object is the intersection point between three or more faces. Faces of polyhedral objects are planar polygons, and edges are line segments.

Any contact state between two polyhedra can be described as a combination of the so-called basic contacts, i.e., in terms of

Manuscript received April 16, 2009; revised September 9, 2009; accepted November 23, 2009. Date of publication January 26, 2010; date of current version September 15, 2010. This work was supported in part by the Spanish Ministerio de Ciencia e Innovación under Grants MTM2007-61554 and TIN2008-06796-C04-01/TSI, by the Visiting Professor Program of the Regione Autonoma della Sardegna, and by the Italian Ministero dell'Istruzione, dell'Università e della Ricerca. This paper was recommended by Associate Editor C.-T. Lin.

S. Cabras is with the Department of Mathematics, Università degli Studi di Cagliari, 09124 Cagliari, Italy (e-mail: scabras@unica.it).

M. E. Castellanos and E. Staffetti are with the Department of Statistics and Operational Research, Universidad Rey Juan Carlos, 28933 Móstoles, Spain (e-mail: maria.castellanos@urjc.es; ernesto.staffetti@urjc.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2009.2038492

vertex–face (v−f) and edge–edge (e−e) contacts [2]. A more concise description of contact states is based on the notion of principal contacts (PCs) which was introduced 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. Formally, a PC describes a contact between a pair of surface elements which are not boundary elements of other contacting surface elements. In this context, 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. The contacting area can be a single point, as in v−f or e−e contacts; a line, as in an edge–face (e−f) contact; or a polygon, as in a face–face (f−f) contact. Thus, a general contact state between two polyhedral objects can be characterized by a set of PCs, called contact formations (CFs) [3].

Task frame formalism [4] is an intuitive and manipulator-independent formalism to specify force-controlled robot tasks in the hybrid control paradigm [5], [6]. It is useful to specify many elementary compliant motion tasks, but it cannot cope with more complex operations involving multiple simultaneous contacts, even in the case of polyhedral objects. In this formalism, the programmer not only has to specify the task but also has to foresee the input sensor signal to control it. For more complex tasks, involving multiple contacts, this can be extremely difficult. Recently, a constraint-based task specification framework, which overcomes these limitations, has been presented in [7].

For robot programming by human demonstration of compliant motion tasks, we gather the pose of the manipulated object and the interaction wrenches between the object manipulated by the user and the environment in which he operates. To collect sensor data, we used a modular demonstration tool specifically designed for compliant motion tasks [8], [9]. It is attached to the object to be manipulated and located between the hand of the human demonstrator and the object itself. It is equipped with a JR3[1] wrench sensor and a probe for accurate pose tracking with a METRIS K600 measurement system.[2] Twists of the manipulated objects are estimated from pose measurements. Data are collected at a frequency of 100 Hz.

As stated earlier, to successfully transfer skills from humans to robots, it is necessary to recognize and to interpret human actions. Humans tend to perceive tasks as a sequence of operations, each of them directly achieving a specific subgoal. These operations are representable as a sequence of elementary actions. An elementary action can be thought of as the smallest symbolic entity that can be used to describe tasks such as touching a table with the manipulated object or moving it in contact with the table surface. Thus, in order to reproduce tasks, it is essential to recognize sequences of elementary actions. This recognition is called segmentation of the task, and it has been studied in several works.

Early deterministic approaches to contact identification mainly exploited geometric knowledge of the contacting objects [10]–[12], whereas first stochastic approaches used hidden Markov models to recognize CFs and detect contact state transitions [13]–[17].

[1]http://www.jr3.com.

[2]http://www.metris.com.

Other works perform contact recognition together with estimation of other unknown quantities, such as the pose of the manipulated object with respect to the gripper of the robot, the pose of the objects in the environment in which the robot operates, and some geometric features of the manipulated object. These quantities are referred to as geometric parameters. By using pose and twists of the manipulated object and the interaction wrenches, they combine estimation of unknown geometrical parameters and contact state transition detection when the sequence of contact states is known or, more generally, contact state recognition when the sequence of contact states is unknown. Under the hypothesis that the sequence of contacts is known, De Schutter *et al.* [18] combine estimation of unknown geometrical parameters and contact state transition detection. Given the geometry of the objects, a kinestatic model [19] is built for each possible type of contact. A Kalman filter model is then derived for each expected contact with implicit measurement equations derived from the kinestatic model. However, a Kalman filter fails to give statistically good estimates if the assumptions under which it is constructed are not valid. This fact is used to detect contact transitions, and the compatibility between sensor data and contacts is assessed by a statistical test χ^2 on the sum of the k latest normalized innovation squared (SNIS) [20]. In [21], which focuses on polyhedral objects, some improvements are introduced to automatically derive the kinestatic models of the contacts. An extended Kalman filter is used for estimation, whereas the SNIS test is used for detecting contact transitions. The same statistical test is used in [22] to detect contact transitions, whereas a nonminimal extended Kalman filter is used for estimation.

In [23], an interacting multiple model estimator is used to recognize CFs. In this approach, a set of models is considered, one for each possible contact state. With these models, several Kalman filters are run in parallel, and the estimation of the state is obtained as a weighted sum of the state estimates from each Kalman filter.

Reference [24] proposes a particle filter (see, for example, [25]) to simultaneously recognize CFs and estimate geometric parameters. This method is able to estimate continuous geometric parameters with a large uncertainty, and it simultaneously recognizes discrete contact states in an experiment consisting of six possible CFs whose sequence is not known. Explicit measurement equations needed by particle filters are also derived from the kinestatic model of the contacts. To cope with all possible contacts between two polyhedral objects, which are in general hundreds even for very simple objects, the contact segmentation method described in [9] uses a more accurate prediction step. It is based on the topological information contained in the CF Graph[3], which is converted into the probability of being in a certain CF at certain time t, CF_t, conditionally on CF_{t-1}. In [26], a method for constructing a model of polyhedral features from pose, twists, and wrench measurements in programming by demonstration of compliant motion tasks is described.

In [27] and [28], a fuzzy classifier and a neural network classifier are proposed to identify single-ended CFs, using only force and torque sensor data without taking into account the geometric models of the objects. The concept of single-ended CFs is similar to the notion of CF and provides a qualitative description of the contact without distinguishing among

features of the environment in contact with a certain feature of the manipulated object. The idea of analyzing force and torque sensor information for contact state identification was introduced in [29]. Another work based on a fuzzy classifier is in [30].

Other approaches to contact identification that model uncertainties are in [31]–[33].

In [34] and [35], a gesture-based robot programming system is presented. The system is assumed to have a set of basic skills called sensorimotor primitives by means of which the system can compose programs. Human demonstration is observed through a gesture recognition system, and the correct skills are selected based on eigenvalue analysis.

Recent works describe other aspects of programming by demonstrations such as action recognition [36], task learning and representation [37], [38], and task generalization over multiple demonstrations [39]. For some tasks, there are underlying constraints that must be fulfilled, and knowing only the final goal is not sufficient. In [40] and [41], techniques to deal with task constraints are described.

A recent comprehensive reference on programming by demonstration is in [42, Ch. 59].

To perform the segmentation process, we propose a statistical machine learning algorithm, in particular a supervised algorithm using twist and wrench sensor data from the task demonstration tool. This consists in looking at the characteristic features of a sample of sensor data $\mathbf{x}$ to build a function $F(\mathbf{x})$ that maps sensor data $\mathbf{x}$ to CFs between the manipulated object and its environment, y. This is done by supposing known all $(\mathbf{x}, y)$ values in a training experiment. The algorithm that learns from this training sample classifies a new sample $\mathbf{x}'$ by assigning a value y'. The classifier we propose belongs to the class of ensemble methods in which many simple classifiers are used to obtain a classifier that is more efficient than one single relatively complex classifier. In our approach, an experiment to train the classifier to recognize all the CFs that, potentially, can take place during the task demonstration is needed. However, the CFs in this training experiment do not need to follow a particular order.

The main difference between our method and the previous supervised methods for the classification of contact states present in the literature [28], [35] is that our method is completely nonparametric, i.e., it does not need the specification of parametric functions, for example, the membership function for a fuzzy classifier.

The advantages of our approach compared to the state of the art are substantial because of the following.

1) A description of the geometry of the objects is not needed. Formally, they do not enter explicitly into the stochastic gradient boosting (SGB) algorithm illustrated in this paper. However, geometry enters via the demonstration experiment. Therefore, a change of geometry at the demonstration level would induce another $F(\mathbf{x})$ that applies to those objects.
2) The calculation of the CF Graph is not needed. This is a great advantage because its computation has prohibitive complexity even for very simple geometric object models [3].
3) The specification of the kinestatic model is totally avoided, while in other approaches, it must be built for every type of contact, which is very complicated in the case of multiple contacts [19].

Moreover, we give the following points.

1) Our classifier works in real time, whereas a sequential Monte Carlo analysis takes several hours on a standard computer.
2) Our approach to CF classification is based on a nonparametric statistical model. Statistical models of this type are more accurate than parametric ones when substantial information on parametric models is unavailable, as the case considered here.
3) Our classifier can recognize both specific CFs, i.e., in which a certain feature of the manipulated object is in contact with another feature of the environment, and families of CFs of the same type, e.g., v–f CFs, regardless of the specific features that are in contact.
4) Our classification method returns the most influent components of the training set to recognize a certain contact state. This information can be used during the demonstration to increase the effectiveness of the programming by demonstration process.

The rest of this paper is organized as follows. In Section II, the setup used for the experiment is described. In Section III, the problem of contact classification is stated, whereas Section IV contains the description of the SGB algorithm used to build the contact classifier. In Section V, the effectiveness of the method for contact segmentation is proven by experimental results. In Section VI, a comparison is made between the results obtained with our classifier and those obtained with a fuzzy classifier. Some important remarks are given in Section VII, and finally, Section VIII presents the conclusions.

II. Experimental Setup

The experimental setup is composed of a cubic manipulated object and an environment consisting of three perpendicular faces.

The human demonstrator moves the manipulated object using a demonstration tool which is assembled onto the manipulated object. A CAD model of the tool design is shown in Fig. 1, while Fig. 2 shows the tool during an experiment. The human demonstrator manipulates the demonstration tool and the object attached to it by means of a handle. The 6-D pose of the demonstration tool is measured by a METRIS K600 optical measurement system [Fig. 1(b)], measuring the spatial positions of a probe integrated in the demonstration tool, at 100 Hz or more, with a volumetric accuracy of 90 μm. The demonstration tool has a hollow prismatic shape whose basis is a regular nonagon. A JR3 wrench sensor [Fig. 1(a)] is assembled inside the demonstration tool between the demonstration tool and the manipulated object, in order to measure the wrench applied by the human demonstrator to the manipulated object. While the wrench is directly measured by the sensor, the pose and the twist of the manipulated object are estimated. More precisely, they are estimated by a linear estimator from the measured sequence of pose of the demonstration tool based on a constant acceleration model [20]. This results in a smooth estimation for the pose and twist of the demonstration tool, while deriving the twist from the pose measurements would

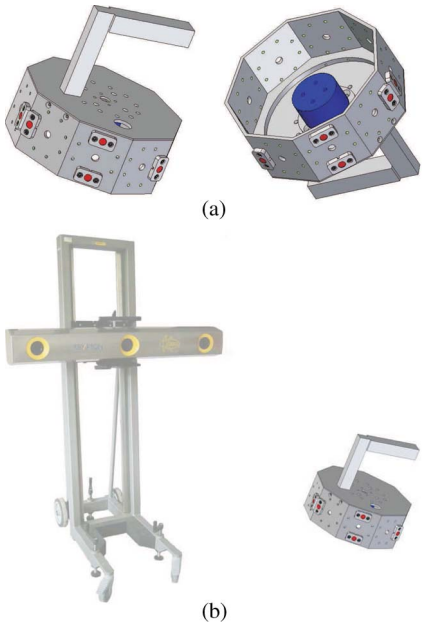

Fig. 1. METRIS K600 measurement system is used to measure the pose of the demonstration tool and to estimate the pose and the twist of the manipulated object. The probe, composed of several LED markers, has been integrated in the demonstration tool. A JR3 wrench sensor, assembled inside the demonstration tool, measures the wrench applied by the human demonstrator to the manipulated object [8], [9]. (a) Demonstration tool. (b) Demonstration equipment.

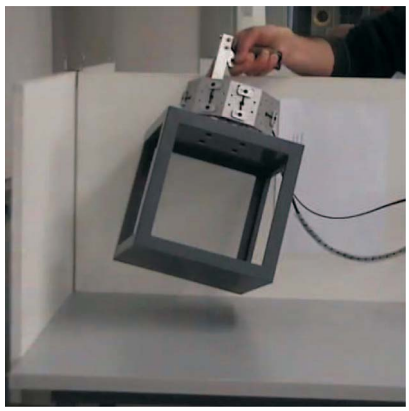

Fig. 2. Demonstration tool during an experiment.

result in a noisy and inaccurate twist estimation. A Kalman filter [43] is used for this linear estimation problem.

III. Description of the Problem

The problem of contact state classification of human-demonstrated robot compliant motion tasks can be stated as follows: Given the sensor data collected during the demonstration of a compliant motion task, identify the CFs that took place among a specified set of CFs.

In this paper, we suppose that the location of the manipulated object with respect to the demonstration tool is uncertain, as well as the location of the objects in the environment in which the demonstration takes place. We also suppose that the geometric parameters of the manipulated object are unknown.

The demonstration setup collects a sequence of poses and twists, i.e., linear and angular velocities of the manipulated object, and wrenches, i.e., forces and torques that arise during the contact interaction with the environment [44], [45]. We show that, in the presence of this kind of uncertainties, we can successfully classify contacts without taking into account the pose information. This means that what identifies CFs is, in this case, the relation between twists and wrenches during the contact [12].

We suppose that, for each CF that can take place during a demonstration of a compliant motion task, the human demonstrator carries out a training experiment by means of the demonstration tool making the CF and recording the corresponding sensor data. After this training phase, a statistical contact classifier is built. With this contact classifier, we are able to classify the CFs that will take place during future demonstrations of compliant motion tasks.

For the sake of clarity of exposition, we describe a CF classifier to recognize a single CF. The generalization to classify several CFs is described in Section VII.

IV. SGB Classifier

Consider the observation of $p = 12$ variables collected during the training experiment. Six of them represent wrenches between the object manipulated by the human demonstrator and the environment in which the demonstration takes place. A wrench

$$\mathbf{w} = (f_x, f_y, f_z, m_x, m_y, m_z)$$

is composed of three forces f_x, f_y, and f_z along the X-, Y-, and Z-axes, respectively, and three torques m_x, m_y, and m_z about the X-, Y-, and Z-axes, respectively. The other six variables represent twists of the manipulated object. A twist

$$\mathbf{t} = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$$

is composed of three linear velocities v_x, v_y, and v_z along the X-, Y-, and Z-axes, respectively, and three angular velocities ω_x, ω_y, and ω_z about the X-, Y-, and Z-axes, respectively.

These observations, along with the indication of the corresponding CFs, are called training set, and they are used to build a statistical model which acts as a CF classifier. To create it, we assume that each CF made in the training experiment is known without uncertainty. By using the same notation as in [46], we denote by

$$\mathbf{x}_i = \left(x_i^1, x_i^2, \ldots, x_i^{12}\right) \tag{1}$$

the ith observation, by $\{\mathbf{x}_i\}_{i=1}^n$ the sequence of n observations in the training experiment, and by $\{y_i\}_{i=1}^n$ the corresponding indicator variables, where

$$y_i = \begin{cases} 1, & \text{if the CF is present} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

We want to estimate the function $y = F(\mathbf{x})$ that represents the relation between sensor data from the demonstration tool $\mathbf{x}$ and CF to identify y. In our problem, F is a highly nonlinear function in which $\mathbf{x}$ can be thought of as a set of random inputs and y as the random output. Given a training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ of known $(\mathbf{x}, y)$ values, the goal is to find a function $F^*(\mathbf{x})$ such that the expected value of some specified loss function,

$L(y, F(\mathbf{x}))$, is minimized. The loss function indicates the amount of loss incurred when there is a wrong classification, i.e., when $y \neq F^*(\mathbf{x})$.

The approximation of F is obtained using the SGB method proposed by Friedman [46] and further discussed in [47]. This method consists in approximating the function F through an additive sum of M basis functions $\{h(\mathbf{x}; \mathbf{a}_m)\}_{m=1}^M$, known as classification trees [48, Ch.9], weighted by a set of scalars $\{\beta_m\}_{m=1}^M$, i.e.,

$$F(\mathbf{x}) \approx F^*(\mathbf{x}) = \sum_{m=1}^{M} \beta_m h(\mathbf{x}; \mathbf{a}_m).$$

This is a nonparametric approximation of the function F and allows F to be expressed in terms of a broad class of basis functions. Once the estimation of $F(\mathbf{x})$ has been built, it is used to recognize whether a CF is present or not in new demonstration tool data $\mathbf{x}$. This is done by computing the weighted majority of M trees, where each tree classifies the CF as present or not.

More precisely, the mth classification tree with K terminal nodes may be represented as the sum of indicator functions $\mathbf{I}(\mathbf{x} \in R_k)$ for $\mathbf{x}$ belonging to a certain region R_k, i.e.,

$$h(\mathbf{x}; \mathbf{a}_m) = h\left(\mathbf{x}; \{c_k, R_k\}_{k=1}^K\right) = \sum_{k=1}^{K} c_k \mathbf{I}(\mathbf{x} \in R_k) \quad (3)$$

where $\{R_k\}_{k=1}^K$ is a partition of the space spanned by $\mathbf{x}$ into K regions R_k and $\mathbf{I}$ takes the value one if its argument is true, and zero otherwise. Likewise, c_k is one or zero depending on whether in the training set the majority of observations falling into region R_k have $y = 1$ or not. The parameters of the mth tree are the coefficients $\{c_k\}_{k=1}^K$ and the regions $\{R_k\}_{k=1}^K$ that we jointly denote by $\mathbf{a}_m$.

Once F^* has been built based on the training set, we want to establish if $y = 1$ or $y = 0$ in a future observation $\mathbf{x}$. Therefore, in the mth tree, $\mathbf{x}$ will fall into one and only one R_k region, and $h(\mathbf{x}; \mathbf{a}_m)$ will vote either one or zero depending on c_k. Finally, by using M classification trees, we have that $F^*(\mathbf{x}) = 1$ if the weighted majority of tree votes are one, or $F^*(\mathbf{x}) = 0$ otherwise. Alternatively, we can give the probability of CF being observed under $\mathbf{x}$, $\Pr(y = 1|\mathbf{x})$, by just computing the weighted proportion of ones. The calculation of this probability is not computationally demanding, and it may be evaluated online leading to an online contact recognition system.

As stated before, in the SGB algorithm, we compute $F^*(\mathbf{x})$ by building the best set of M classification trees and weights $\{\beta_m\}_{m=1}^M$. This is done using a steepest descent optimization algorithm applied to the function space which F belongs to and where function gradients are derived from $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$. Here, we provide a brief summary of the SGB algorithm.

An essential ingredient of this algorithm is the loss function $L(y, F(\mathbf{x}))$, which changes if we want to recognize more than one CF as explained in Section VII.

For the recognition of one CF, we use, as a loss function, the deviance of the Bernoulli model, defined as $D(y, p) = -2\log[p^y(1-p)^{1-y}]$, where

$$p = \Pr(y = 1|\mathbf{x}) = \frac{\exp(F(\mathbf{x}))}{1 + \exp(F(\mathbf{x}))} \text{ (logit link)}.$$

Substituting this expression of p in $D(y, p)$, we have

$$L(y, F) = -2\left(yF(\mathbf{x}) - \log\left[1 + \exp(F(\mathbf{x}))\right]\right). \quad (4)$$

From (4), we derive the empirical risk function over the observed sequence in the training experiment

$$\Phi(F) = \sum_{i=1}^{n} L\left(y_i, F(\mathbf{x}_i)\right) \quad (5)$$

which is the total amount of loss in the training set for a given F. The SGB algorithm consists in choosing an

$$F^* = \arg\min_F \Phi(F) \quad (6)$$

where the solution has the form

$$F^* = \sum_{m=0}^{M} f_m(\mathbf{x})$$

and $\{f_m(\mathbf{x})\}_0^M$ are the weighted basis functions known as boosts, while f_0 is an initial guess set to $f_0 = \log(\sum_{i=1}^n y_i/(n - \sum_{i=1}^n y_i))$, which is equal to the logit of the proportion of $y = 1$ in the training set.

The central idea of this optimization method is to reduce the search space for F by allowing classification trees to be fitted in the steepest descent algorithm, in which the empirical gradients at step m used to solve the optimization problem (6)

$$\tilde{y}_i = \left[\frac{\partial L\left(y_i, F(\mathbf{x}_i)\right)}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}$$

are called pseudoresponses. At each step for the pseudoresponses $\tilde{y}_i$, the gradient boosting algorithm calculates the most parallel gradient (based on classification trees) to these empirical gradients by solving the following equation:

$$\mathbf{a}_m = \arg\min_{\mathbf{a},\beta} \sum_{i=1}^{n} \left(\tilde{y}_i - \beta h(\mathbf{x}_i; \mathbf{a})\right)^2. \quad (7)$$

Note that β only appears in this minimization step and that it affects the calculation of $\mathbf{a}$. It is no longer used in the update of F. This negative constrained gradient $h(\mathbf{x}; \mathbf{a})$ is used to solve the original minimization problem (6) that turns out to be a univariate minimization problem solved by calculating the length of the step ρ_m in the direction fixed by $\mathbf{a}_m$

$$\rho_m = \arg\min_{\rho} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(\mathbf{x}_i) + \rho h(\mathbf{x}_i; \mathbf{a}_m)\right) \quad (8)$$

leading to the following update at step m:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \lambda\rho_m h(\mathbf{x}; \mathbf{a}_m) \quad (9)$$

where $\lambda > 0$ is called learning rate and its value is chosen to optimize the performance of the CF classifier as explained hereinafter.

An extension of the gradient boosting algorithm is its stochastic version where, at each step, a random fraction of the training set is selected to update F_m. More precisely, letting $\{\pi(i)\}_{i=1}^n$ be a random permutation of the indices $1, \ldots, n$, then a random subsample of size $\tilde{n} < n$ is obtained by selecting

the first $\tilde{n}$ elements, i.e., $\{\mathbf{x}_{\pi(i)}, y_{\pi(i)}\}_{i=1}^{\tilde{n}}$. Algorithm 1 summarizes the SGB algorithm that classifies CFs.

Algorithm 1 Stochastic Gradient Boosting
Require: Initialize f_0 as above.
1: **for** $m = 1$ to M **do**
2: permute the n indices $\{\pi(i)\}_{i=1}^{n}$;
3: compute the pseudoresponses

$$\left\{ \tilde{y}_{\pi(i)} = -\left[\frac{\partial L\left(y_{\pi(i)}, F\left(\mathbf{x}_{\pi(i)}\right)\right)}{\partial F\left(\mathbf{x}_{\pi(i)}\right)} \right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})} \right\}_{i=1}^{\tilde{n}};$$

4: calculate the parameters from the tree by solving
$\mathbf{a}_m = \arg\min_{\mathbf{a},\,\beta} \sum_{i=1}^{\tilde{n}} (\tilde{y}_{\pi(i)} - \beta h(\mathbf{x}_{\pi(i)}; \mathbf{a}))^2$;
5: calculate step length with
$\rho_m = \arg\min_{\rho} \sum_{i=1}^{n} L(y_i, F_{m-1}(\mathbf{x}_i) + \rho h(\mathbf{x}_i; \mathbf{a}_m))$;
6: update F using
$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \lambda \rho_m h(\mathbf{x}; \mathbf{a}_m)$.
7: **end for**

A. Choice of λ and M and Variable Importance

In this section, we would like to highlight two important facts.

1) If M is large, the SGB will perfectly fit the training set used for update (9), but it will poorly fit a new observation. Therefore, the original training set is randomly split at each step m in two sets: The first set composed of $\tilde{n} = [n/2]$ observations is used for the update (9) as explained earlier, and the second one, called test set, made by $n - \tilde{n}$ observations is used to estimate the classification error of $F_M(\mathbf{x})$ by the empirical risk (5). In other words, loss function (4) at a certain M will increase in the test set. Therefore, the value of M should be chosen to minimize (4) in the test set. In general, best results are obtained by minimizing (5) with small λ and large M that correspond to many small steepest descent steps. A practical way to choose λ is based on classifier performances. One can reduce, but not eliminate, the classification error by decreasing λ and increasing M. For our purposes, we found $\lambda = 0.01$ small enough.
2) Only some of the p variables are used to build regions R_k in the M trees. Indeed, only those variables that assure the largest decreasing of the loss function (4) are used to build the M trees. Therefore, we can derive a measure of variable importance, the relative influence, by simply adding up the decrements in (4) globally induced in the M trees for each of the p variables. In the contact classifiers, the importance of the variables depends on the CF to be recognized and the CFs (not on their sequence) made in the demonstration experiment. Relative influence is not really the object of this work, because the SGB is a statistical model mainly oriented to prediction rather than interpretation of the "real world." As long as F^* is a good predictor of CF (i.e., a good classifier), the interpretation of the relative influence may be of use in planning further training experiments. For instance, one may emphasize certain movements or forces that the relative influence suggested to be good predictors for the CF of interest.

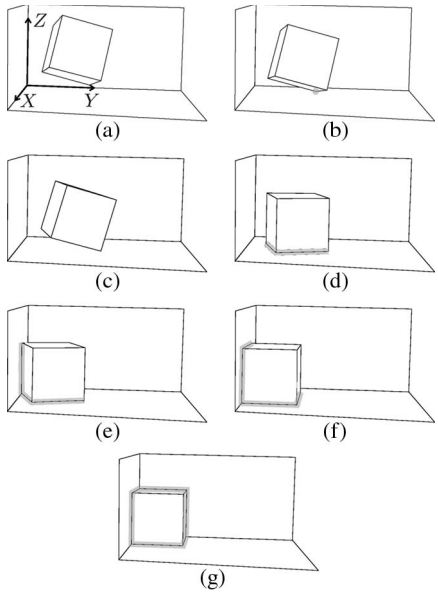

Fig. 3. Sequence of CFs made during the test experiment for the recognition of a three face–face CF. Visible elements of the manipulated object that are in contact with the environment are highlighted in gray. (a) No Contact. (b) Vertex–face. (c) Edge–face. (d) Face–face. (e) Face–face and edge–face. (f) Face–face and face–face. (g) Three face–face.

V. Experimental Validation

We apply the proposed contact classifier to recognition of CFs between a cubic manipulated object and an environment composed of three orthogonal faces, using the experimental setup described in Section II.

It is important to point out that form and dimensions of the manipulated object are not relevant for our contact classifier. It can classify contacts between convex and nonconvex polyhedral objects, and it is invariant to scale transformations of the object manipulated by the human demonstrator and of the objects present in the environment in which the demonstration takes place.

In particular, we want to recognize the following:

1) when the cube is in the corner, i.e., when it makes three f−f contacts with the environment (the sequence of CFs made during the test experiment is shown in Fig. 3);
2) when a vertex of the cube is in contact with the horizontal face of the environment, i.e., when the cube makes a v−f contact with the environment (the sequence of CFs made during the test experiment is shown in Fig. 4);
3) when an edge of the cube is in contact with the horizontal face of the environment, i.e., when the cube makes an e−f contact with the environment (the sequence of CFs made during the test experiment is shown in Fig. 5).

While, in the first case, we are interested in recognizing a specific CF, in the last two cases, we test our classifier on the more difficult problem of recognizing classes of CFs, namely, the v−f and e−f contacts that take place during a demonstration.

For each case, we use two different experiments: The first one is the training experiment, and the second one is the test experiment used to validate the classifier. We consider that the contact state corresponds to the CF we want to recognize if $\Pr(y = 1|\mathbf{x}) > 0.9$. Note that this threshold is rather conservative

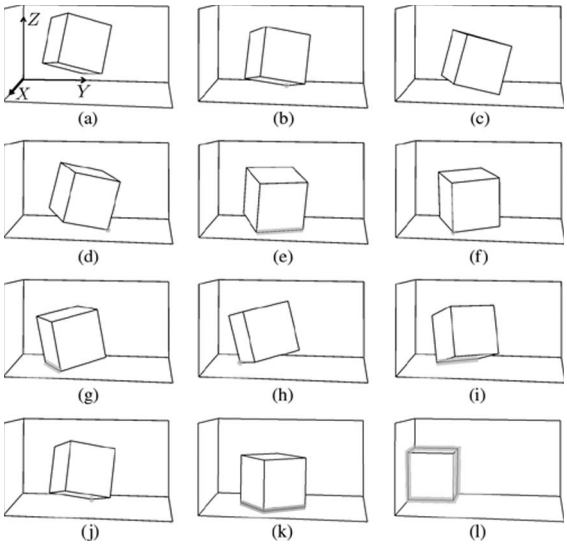

Fig. 4. Sequence of CFs made during the test experiment for the recognition of vertex–face CFs. Visible elements of the manipulated object that are in contact with the environment are highlighted in gray. (a) No contact. (b) Vertex–face. (c) Edge–face. (d) Vertex–face. (e) Edge–face. (f) Vertex–face. (g) Edge–face. (h) Vertex–face. (i) Edge–face. (j) Vertex–face. (k) Face–face. (l) Three Face–face.

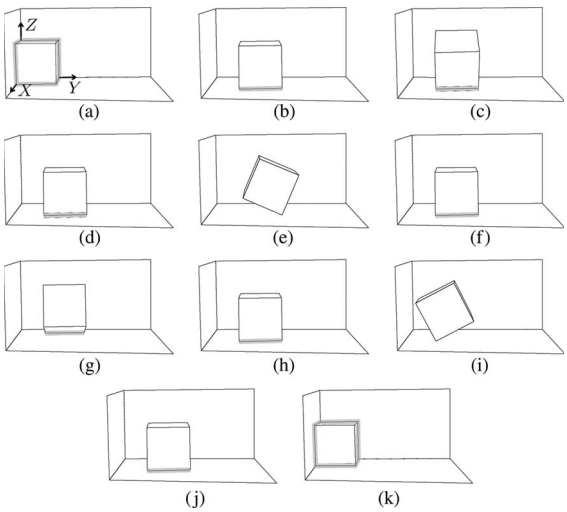

Fig. 5. Sequence of CFs made during the test experiment for the recognition of edge–face CFs. Visible elements of the manipulated object that are in contact with the environment are highlighted in gray. (a) Three face–face. (b) Face–face. (c) Edge–face. (d) Face–face. (e) Edge–face. (f) Face–face. (g) Edge–face. (h) Face–face. (i) Edge–face. (j) Face–face. (k) Three face–face.

because to recognize a CF, it would be enough $\Pr(y = 1|\mathbf{x}) > 0.5$. In general, let $Pr(y = 1|\mathbf{x}) > c$, with $c \in (0, 1)$. If $c \to 1$, then the SGB will be conservative in recognizing the CF of interest. This happens because a contact state is declared to be that CF if the probability that this is true is almost one. The opposite occurs if $c \to 0$. In summary, if $c \to 0$, the number of false detections of the CF of interest is high, but the number of missed detections is low, while the contrary occurs when $c \to 1$.

A. Recognition of a Three Face–Face Contact Formation

In this case, we want to distinguish the specific CF in Fig. 3(g), denoted by cc, in which the cube makes three f−f contacts with the environment, from the other CFs. The sequence of CFs made during the test experiment for the recognition of this CF is shown in Fig. 3. In this case, we built a classifier based on a training set composed of $n = 1001$ demonstration tool sensor data in which $\sum_{i=1}^{n} y_i = 169$ corresponds to the cc CF and the rest to other CFs. We validated the CF classifier with a second experiment with different CFs having 1422 demonstration tool sensor data, 500 of which corresponding to the cc CF as shown in Fig. 9 (top). Fig. 6 shows the demonstration tool sensor data $\mathbf{x}$ for the training experiment. Note that, for the sake of readability, $\mathbf{x}$ data shown in the figures are low pass filtered with a moving average at 1 lag, whereas $\mathbf{x}$ data used in the SGB algorithm are not filtered at all. This makes our classifier independent of a specific filtering algorithm.

Consider Fig. 7. It contains an example of classification tree $h(\mathbf{x}, \mathbf{a})$ applied to the data set shown in Fig. 6. For these data, which differ from the M random subset in the SGB algorithm, we can see that the most important splitting variable is f_x. The cut point of 12.93 N induces the largest decrement in the loss function L. This variable clearly interacts with m_y. In fact, looking at the right branch, when $f_x > 12.97$ N and $m_y > 1.73$ N · m, the state is classified as cc CF. This decision is supported by the fact that this terminal node has 67 states, where 63 of them correspond to the cc CF. To estimate the optimal number of boosts, we use the value of M that corresponds to a minimum of the deviance. We can see in Fig. 8 (left) that the error in the test set increases after about 2000 iterations; therefore, $M = 2000$ is enough to estimate $F(\mathbf{x})$.

The interpretation of the final model $F^*(\mathbf{x})$ is very difficult because it is made of thousands of classification trees like that in Fig. 7. Nonetheless, we can affirm that to recognize the cc CF, the most important variables are the forces along the Z- and Y-axes together with the translation velocity along the X-axis as we can observe in Fig. 8 (right). In order to appreciate prediction accuracy, we can consider that a cc CF in the test experiment has occurred if its probability, conditioned on the observed $\mathbf{x}$, is greater than 0.9. Fig. 9 shows, at the top, the true contact state in the test experiment and, at the bottom, their respective probability estimation to be in the cc CF.

We can see that the contact classifier estimates the true cc CF very accurately, with 4% being the classification error in this experiment. Only 54 out of 1422 sensor data have not been correctly classified.

B. Recognition of Vertex–Face Contact Formations

In this case, we tested our contact classifier in the recognition of a class of CFs rather than of a specific CF, namely, all the v−f CFs that took place in the test experiment. For v−f CF recognition, we used demonstration tool sensor data collected from a training experiment with 2691 samples, in which 1099 correspond to v−f CFs. The sequence of CFs that took place during the test experiment of v−f CF recognition is shown in Fig. 4, and the demonstration tool sensor data collected in the training experiment are shown in Fig. 10. In this case, we want to distinguish v−f CFs shown in Fig. 4(b), (d), (f), (h), and (j) from the rest of CFs.

From Fig. 11, we can see that the optimal number of boosts is $M = 1000$ and the most important variable used to recognize the v−f CF is the wrench about the Y-axis, m_y.

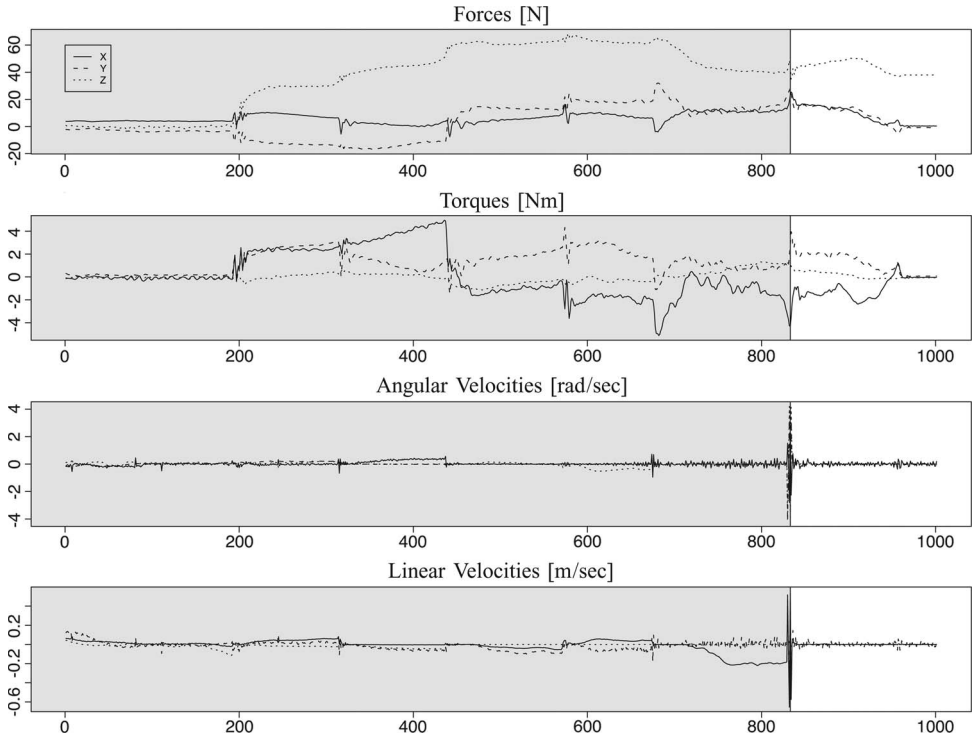

Fig. 6. Demonstration tool sensor data collected during the training experiment for the recognition of a three face–face CF. On the horizontal axis, the index of the samples is reported. Data in the nonshaded interval correspond to the three face–face CF.

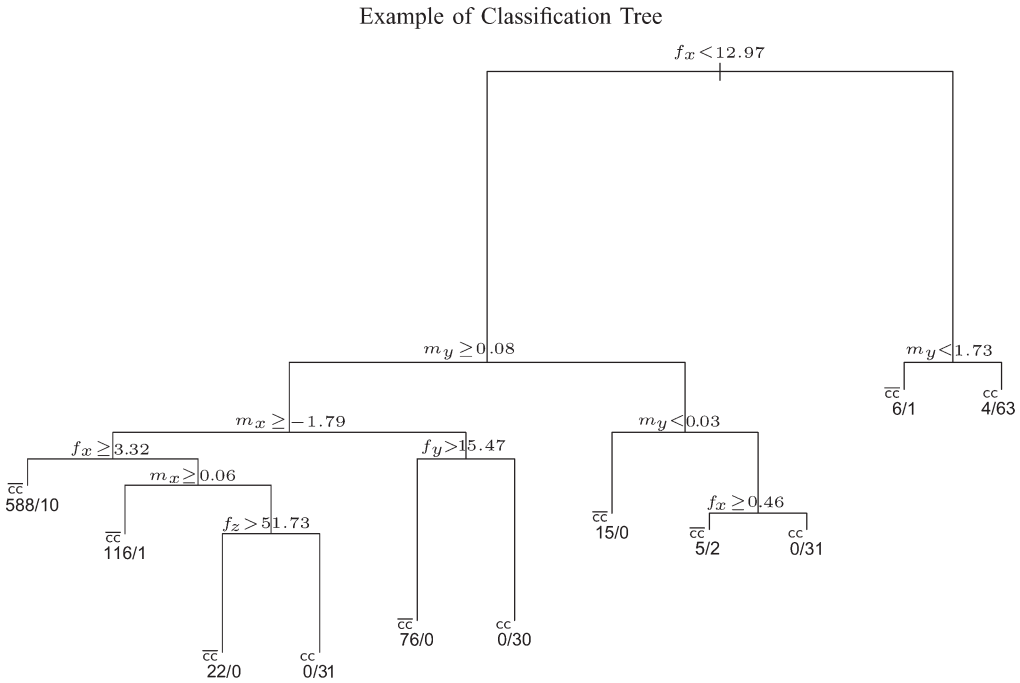

Fig. 7. Example of a classification tree applied to demonstration tool data shown in Fig. 6. We can see that the most important variable is, in this case, f_x because it induces the largest decrement in the loss function L (larger vertical segments). This variable clearly interacts with m_y. In fact, looking at the right branch, when $f_x > 12.97$ N and $m_y > 1.73$ N · m, the state is classified as three face–face CF (cc) with probability $\Pr(y = 1|x) = 63/67$ as 67 training states belong to the region defined by $f_x > 12.97$ N and $m_y > 1.73$ N · m, where 63 of them are three face–face CF. Splitting variables and splitting points are chosen according to (7). Note that, in the SGB algorithm, we use $M = 2000$ classification trees, each fitted on M random subsamples of the demonstration tool sensor data shown in Fig. 6.

Fig. 12 shows, at the top, the true v−f CF for the test experiment with 2655 sensor data and, at the bottom, the estimated probability of v−f CF. We can see that there is a strong consistency between the CFs estimated by the proposed contact classifier and the CFs that took place in the test experiment. In fact, a v−f CF is almost persistent in the central part of the experiment being broken by short e−f CFs as shown in Fig. 4(c), (e), (g), and (i). The classifier we propose is able to recognize

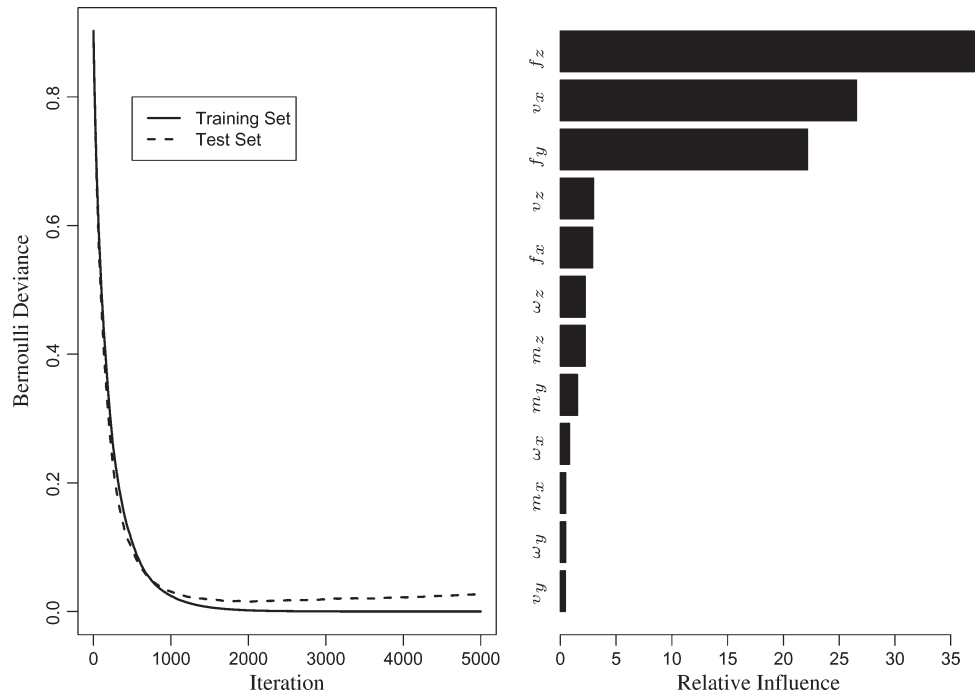

Fig. 8. Recognition of a three face–face CF. To estimate the optimal number of boosts, we use the value that corresponds to the minimum of the deviance. We can see in the left graph that, after about 2000 boosts, the deviance increases in the test set. The right graph reports the relative importance of the components of the demonstration tool data $\mathbf{x}$. It is clear that the most influent components to recognize the three face–face CF based on the training experiment are f_z, v_x, and f_y.

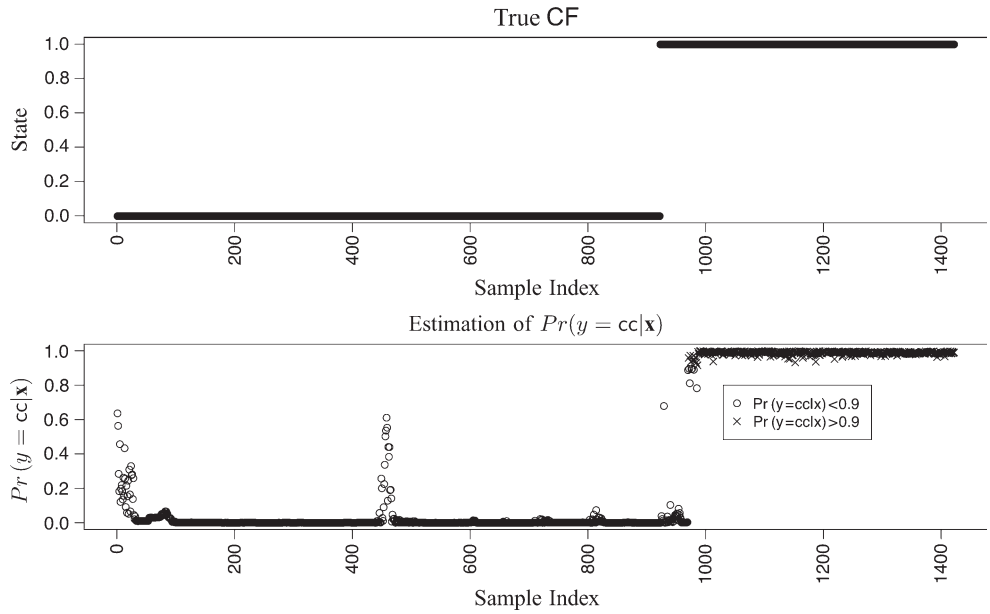

Fig. 9. Recognition of the three f−f CF shown in Fig. 3(g). This CF is denoted by cc. Top figure shows the true CF, and the bottom figure shows the estimation of $Pr(y = \mathsf{cc}|\mathbf{x})$. From sample 550 on the CF corresponds to this CF.

this type of CF, by significantly decreasing the probability of v−f CF, as expected. In this case, the classification error is around 10%, i.e., 254 out of 2655 sensor data have not been correctly classified.

C. Recognition of Edge–Face Contact Formations

In this case, we tested our contact classifier in the recognition of a class of e−f CFs that took place in the test experiment. For e−f recognition, we used demonstration tool sensor data collected from a training experiment with 2454 samples in which 905 correspond to e−f CFs. The sequence of CFs that took place during the training experiment of e−f CF recognition is shown in Fig. 5, and the demonstration tool sensor data are shown in Fig. 13. In this case, we want to distinguish e−f CFs shown in Fig. 5(c), (e), (g), and (i) from the rest of CFs.

From Fig. 14, we can see that the optimal number of boosts is $M = 1353$, and the most important variable used to recognize the e−f CF is the linear velocity along the Z-axis, v_z.

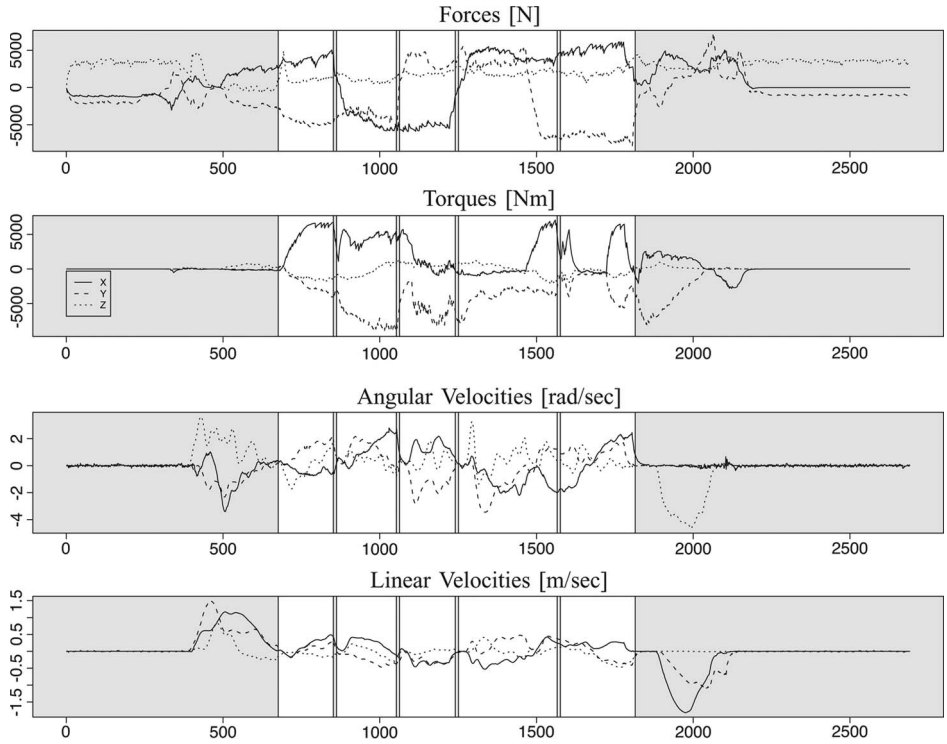

Fig. 10. Demonstration tool sensor data collected during the training experiment for the recognition of vertex–face CFs. Data in the nonshaded intervals correspond to vertex–face CFs.

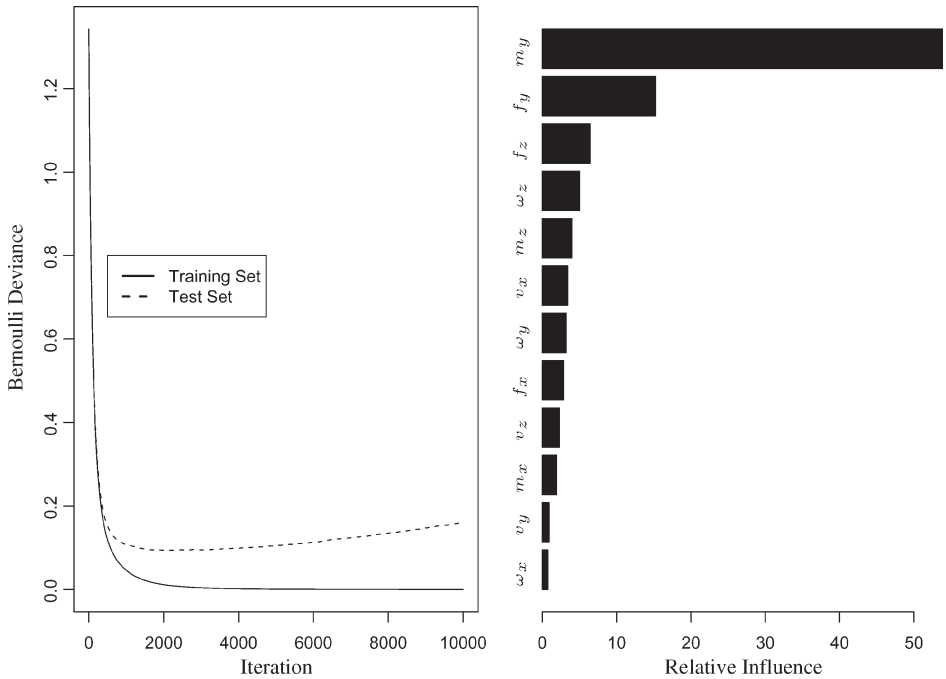

Fig. 11. Recognition of vertex–face CFs. We can see in the left graph that, after about 1000 boosts, the deviance increases in the test set. The relative importance of the components of $\mathbf{x}$ is shown in the right graph. It is clear that the most influent variable to recognize vertex–face CFs, based on the training experiment, is the wrench about the Y-axis, m_y.

Again, we compare the performance of the proposed online classifier by means of Fig. 15. It shows, at the top, the true CFs taking place in the test experiment, with 2415 sensor data, and, at the bottom, the estimated ones. We can see that the classifier recognizes almost all the CFs of interest. In particular, in this experiment, we have a classification error of just 4%, i.e., only 108 out of 2415 sensor data have not been correctly classified.

VI. Comparison With a Fuzzy Classifier

In this section, we describe the results of a comparison between our SGB classifier and the fuzzy classifier proposed in [28]. This fuzzy classifier uses only normalized forces and torques, and it was originally tested on data coming from experiments performed with a robot manipulator rather than a human demonstrator. Experiments done with robots produce

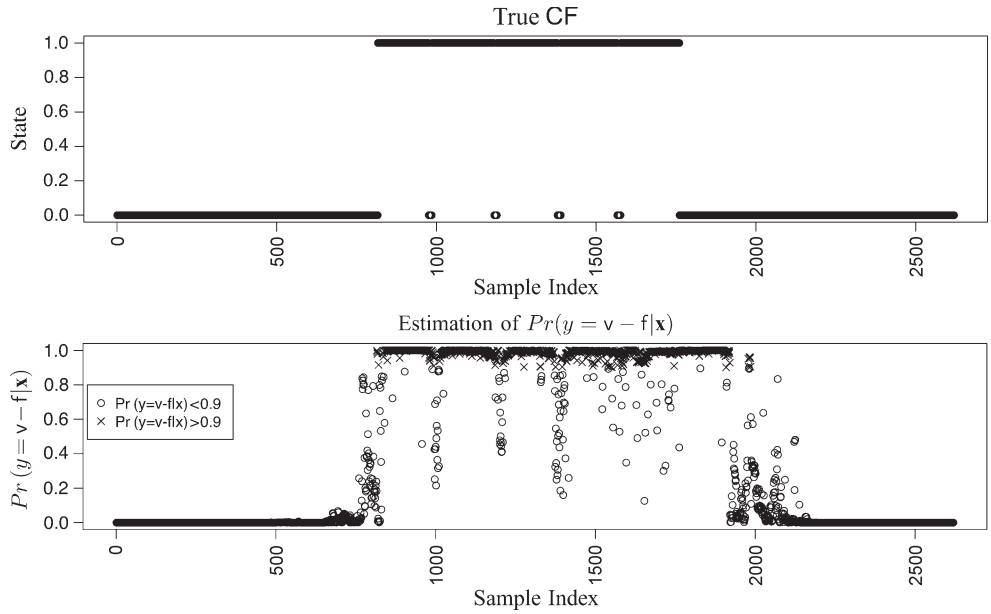

Fig. 12. Recognition of vertex–face CFs. Top figure shows the true contact state, and the bottom figure shows the estimation of $Pr(Y = \mathsf{v} - \mathsf{f}|\mathbf{x})$. The sequence of vertex–face CFs from sample 800 to sample 1750 corresponds to those shown in Fig. 4(b), (d), (f), (h), and (j). Isolated zero points correspond to the edge–face CFs shown in Fig. 4(c), (e), (g), and (i).

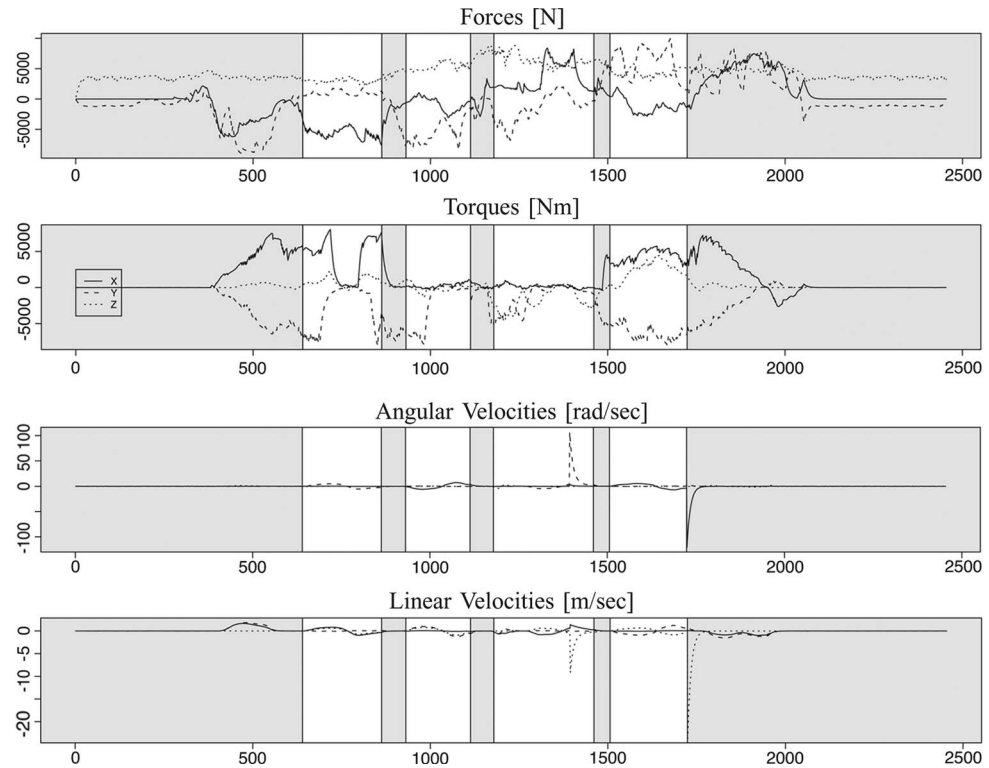

Fig. 13. Demonstration tool sensor data collected during the training experiment for the recognition of edge–face CFs. Data in the nonshaded intervals correspond to edge–face CFs.

data with less noise than those obtained from a human demonstration. Our SGB classifier also uses twists, although for the comparison only, wrench information has been used. To recognize a specific CF, the fuzzy classifier needs a complete segmentation of the CFs that took place in the training phase, whereas in our approach, we only need to distinguish the CF of interest from the rest of CFs that have occurred. Furthermore, the SBG classifier can recognize both specific CFs, i.e., in which a certain feature of the manipulated object is in contact with another concrete feature of the environment, and families of CFs of the same type, e.g., v–f CFs.

The training experiment in the fuzzy classifier is used to calculate the mean μ and standard deviation σ of the sensor data for each contact state. As a consequence, we have six membership functions, one for each variable, defined as $\pi(r) = 1 - \mathrm{e}^{(-3\sigma/|\mu - r|)^3}$. Finally, a fuzzy conjunction operator

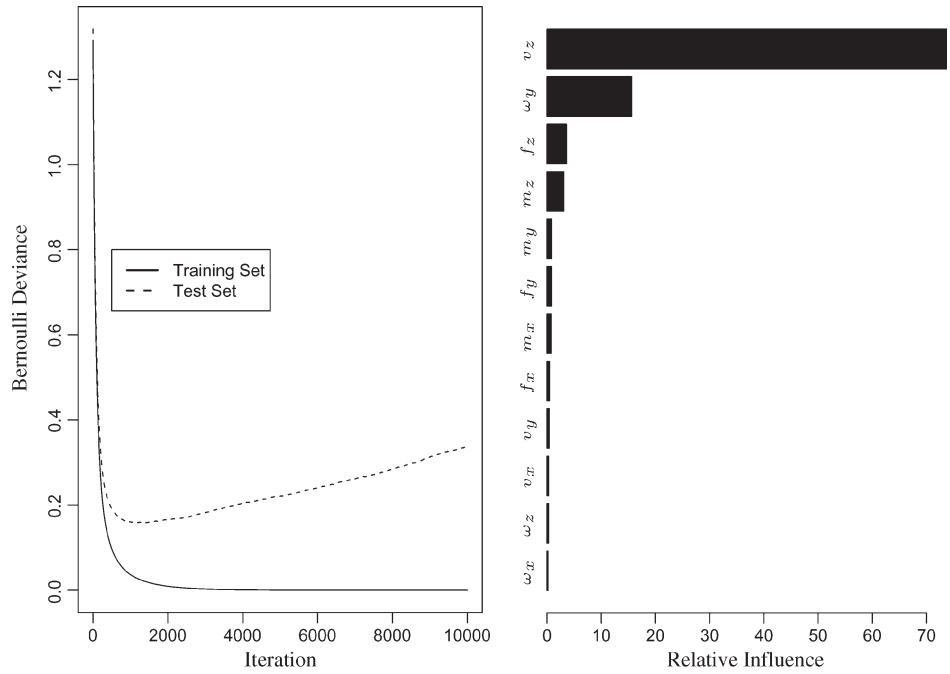

Fig. 14. Recognition of edge–face CFs. We can see in the left graph that, after about 1300 boosts, the deviance increases in the test set. The relative importance of the components of $\mathbf{x}$ is shown in the right graph. It is clear that the most influent variable to recognize edge–face CFs, based on the training experiment, is the linear velocity along the Z-axis, v_z.

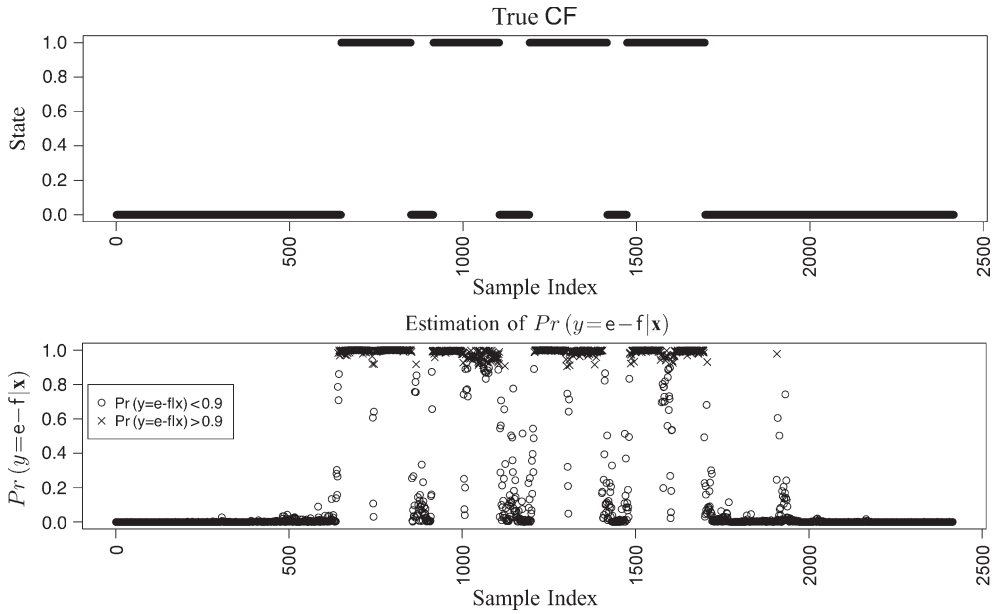

Fig. 15. Recognition of edge–face CFs. Top figure shows the true contact state, and the bottom figure shows the estimation of $Pr(y = \mathrm{e} - \mathrm{f}|\mathbf{x})$. The sequence of edge–face CFs from sample 600 to sample 1700 corresponds to those shown in Fig. 5(c), (e), (g), and (i). Isolated zero points correspond to the face–face CFs shown in Fig. 5(d), (f), (h), and (j).

is used to combine the six membership functions for each contact state. We use the Hamacher product as a conjunction operator, as proposed in [28]. The classifier returns possibility values for each class where the highest value represents the identified contact state. As in the original paper, we also use the threshold value 0.5 to determine when a classification can be made. If there are no possibility values above the threshold value 0.5, then the data vector remains unclassified.

We compare the results obtained with the fuzzy classifier and the SGB classifier on a subset of sensor data that corresponds to a subset of CFs of the three experiments described earlier. In order to make the results comparable, we consider pairs of specific contacts in each of the three experiments.

A. Recognition of a Three Face–Face and an Edge–Face Contact Formations

From the data set analyzed in the previous section and shown in Fig. 6, we extracted a subset of data in which two CFs occur. We focused on the recognition of a three f−f CF, which

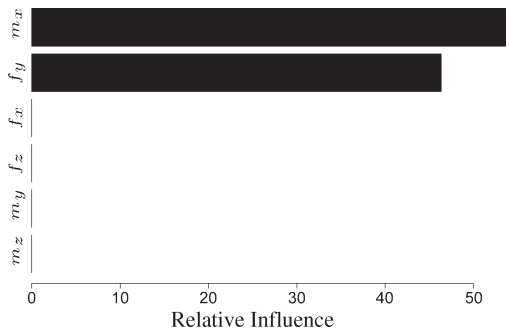

Fig. 16. Relative influence of the components of $\mathbf{x}$, which in this case is composed only of forces and torques, for the recognition of the edge–face CF in Fig. 3(c) and the three face–face CF in Fig. 3(g). The most influent variables to be distinguished between these two CFs, based on the training experiment, are the torque about the X-axis, m_x, and the force along the Y-axis, f_y.

occurs when the cube is in the corner as in Fig. 3(g), and the specific e−f contact shown in Fig. 3(c). Once both classifiers were trained, we used another data set as a test set to recognize the CFs of interest. The results show that both classifiers are quite accurate, with the SGB classifier being superior, since the classification error is 0.001 for the SGB classifier, while for the fuzzy classifier, it is 0.01, i.e., ten times higher. In this case, the fuzzy classifier does not leave any data unclassified. As we can see in Fig. 16, there are two important variables in order to distinguish between the two considered CFs, the torque about the X-axis, m_x, and the force along the Y-axis, f_y. While our classifier automatically selects the most important variables, the fuzzy classifier is based on the convolution of six membership functions, carried out without evaluating which variables are more important for the classification. This results in a worse accuracy.

B. Recognition of Two Different Vertex–Face Contact Formations

In this case, we used a subset of data in Fig. 10 to train both classifiers. We focused on the recognition of the two v−f CFs in Fig. 4(b) and (d). As before, once both classifiers were trained, we used another data set as a test set to recognize the CFs of interest. In this case, the fuzzy classifier leaves some states without being classified. This occurs to 11% of data, while the classification error is 3%. This corresponds to an 86% of successful classifications, while the SGB classifier is clearly superior with a classification success of 97%. Fig. 17 shows the importance of each of the six variables. Again, not all components of $\mathbf{x}$ are important in order to distinguish between these two contacts, and variables other than the force along the X-axis, f_x, just add noise to the classifier.

C. Recognition of an Edge–Face and a Face–Face Contact Formation

For the last comparison, we considered a subset of data shown in Fig. 13 to train both classifiers. We focused on the recognition of the e−f CF and f−f CF corresponding to the contacts shown in Fig. 5(c) and (d), respectively. As before, we used another data set as a test set to recognize the CFs of interest after training both classifiers. The fuzzy classifier is unable to classify 10% of data. Again, the SGB classifier is superior

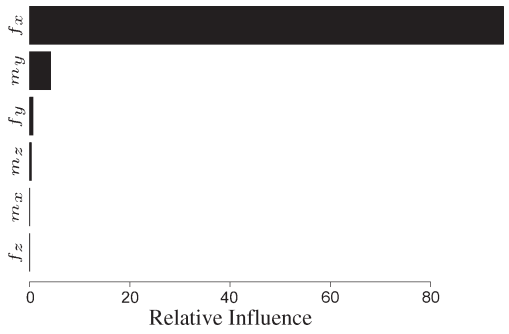

Fig. 17. Relative influence of the components of $\mathbf{x}$, which in this case is composed only of forces and torques, for the recognition of the vertex–face CFs in Fig. 4(b) and (d). The only influent variable to be distinguished between these two CFs is the force along the X-axis, f_x.

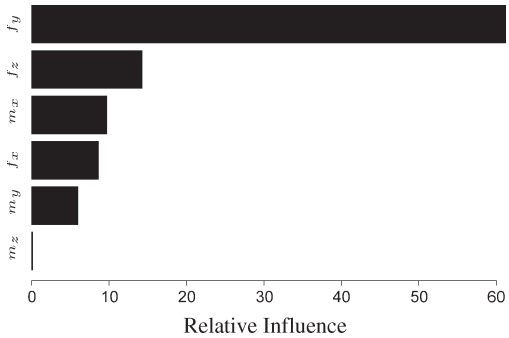

Fig. 18. Relative influence of the components of $\mathbf{x}$, which in this case is composed only of forces and torques, for the recognition of the edge–face CF in Fig. 5(c) and the face–face CF in Fig. 5(d). The two most influent variables to be distinguished between these CFs, based on the training experiment, are the force along the Y-axis, f_y, and the force along the Z-axis, f_z.

with a classification success of 94%, while the fuzzy one has a classification success of 84%. As we can see in Fig. 18, the most important variable in order to distinguish between these two contacts is the force along the Y-axis, f_y. However, in this case, there are other important variables such as f_z, m_x, f_x, and m_y.

VII. Further Remarks

The classifier we propose is based on an accurate segmentation of a test experiment. We segmented the data sets collected during the training and test experiments by inspection; with a small modification of the demonstration tool, this segmentation could be performed online at demonstration time by manually indicating when the CF we want to recognize is taking place.

Recognition of several contacts may be implemented by means of a suitable loss function (4). For example, if one wants to recognize jointly J different CFs, loss function (4) may be the log-likelihood of a multinomial distribution

$$L\left(\{y_j, F_j(\mathbf{x})\}_1^J\right) = -\sum_{j=1}^{J} y_j \log p_j(\mathbf{x})$$

where $y_j = 1$ if $\mathbf{x}$ has been observed under the jth CF and

$$p_j(\mathbf{x}) = \exp\left(F_j(\mathbf{x})\right) / \sum_{j=1}^{J} \exp\left(F_j(\mathbf{x})\right)$$

is the probability of $\mathbf{x}$ to be observed under the jth CF.

It is important to point out that the relative importance of the variables in recognizing a CF depends on the particular training experiment, and variable selection is not compelled in fuzzy classifiers. Adding unimportant variables just adds noise to the classifier, downgrading its performances.

The CF classifier proposed here is essentially a conditional forecast model for the true current CF given the observed sensor data. This stochastic model is not constrained by transitions from one contact to another as those of the approaches based on the contact state graph [3], such as that in [9].

The most important feature of our classifier is that it relies neither on the geometric models of the objects involved in the compliant motion nor on the kinestatic models of the CFs [19]. Kinestatic models of contacts are, in general, ideals models that do not take into account friction, whereas the model proposed here inherently considers the uncertainty caused by friction. Therefore, it may be extended to classification of contacts between rigid nonpolyhedral objects [49] and even to classification of contacts between nonrigid objects.

VIII. Conclusion

In this paper, a method for contact state classification in robot programming of compliant motion tasks by human demonstration has been described. This contact classifier is based on the supervised learning SGB algorithm trained by means of a training experiment. To the best knowledge of the authors, the SGB algorithm has never been applied to contact state classification.

The main advantage in using this method is that both the geometric models of the objects and the kinestatic models of their interaction are not needed. Moreover, it does not rely on the CF graph, whose computation has prohibitive complexity even for very simple geometric object models.

The accuracy of this approach to the classification of both specific CFs and CFs of the same type has been proven using real data. A comparison has been made between our SGB classifier and a fuzzy classifier whose results show that our method is more accurate in terms of classification error. One of the reasons of this superiority is that the SGB classifier automatically performs variable selection while the fuzzy classifier does not.

Acknowledgment

The authors would like to thank J. De Schutter and H. Bruyninckx of the Department of Mechanical Engineering, Katholieke Universiteit Leuven, for giving them the possibility of using their experimental setup for robot programming by demonstration.

References

[1] W. Meeussen, E. Staffetti, H. Bruyninckx, J. Xiao, and J. De Schutter, "Integration of planning and execution in force controlled compliant motion," *Robot. Auton. Syst.*, vol. 56, no. 5, pp. 437–450, May 2008.

[2] J. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer, 1991.

[3] J. Xiao and X. Ji, "On automatic generation of high-level contact state space," *Int. J. Robot. Res.*, vol. 20, no. 7, pp. 584–606, Jul. 2001.

[4] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, no. 6, pp. 418–432, Jun. 1981.

[5] M. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *ASME J. Dyn. Syst., Meas., Control*, vol. 102, no. 2, pp. 126–133, 1981.

[6] H. Bruyninckx and J. De Schutter, "Specification of force-controlled actions in the 'Task Frame Formalism': A survey," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 581–589, Aug. 1996.

[7] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decre, R. Smits, E. Aertbelien, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *Int. J. Robot. Res.*, vol. 26, no. 5, pp. 433–455, May 2007.

[8] J. Rutgeerts, P. Slaets, F. Schillebeeckx, W. Meeussen, W. Verdonck, 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. Intell. Robots Syst.*, Edmonton, AB, Canada, 2005, pp. 3592–3597.

[9] W. Meeussen, J. Rutgeerts, K. Gadeyne, H. Bruyninckx, and J. De Schutter, "Contact state segmentation using particle filters for programming by human demonstration in compliant motion tasks," *IEEE Trans. Robot.*, vol. 23, no. 2, pp. 218–231, Apr. 2007.

[10] S. Hirai and H. Asada, "Kinematics and statics of manipulation using the theory of polyhedral convex cones," *Int. J. Robot. Res.*, vol. 12, no. 5, pp. 434–447, Oct. 1993.

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

[12] H. Bruyninckx, S. Demey, S. Dutré, and J. De Schutter, "Kinematic models for model based compliant motion in the presence of uncertainty," *Int. J. Robot. Res.*, vol. 14, no. 5, pp. 465–482, Oct. 1995.

[13] B. Hannaford and P. Lee, "Hidden Markov model analysis of force/torque information in telemanipulation," *Int. J. Robot. Res.*, vol. 10, no. 5, pp. 528–539, Oct. 1991.

[14] B. J. McCarragher and H. Asada, "Qualitative template matching using dynamic process models for state transition recognition of robotic assembly," *ASME J. Dyn. Syst., Meas., Control*, vol. 115, no. 2A, pp. 261–269, Jun. 1993.

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

[16] B. McCarragher, G. Hovland, P. Sikka, P. Aigner, and D. Austin, "Hybrid dynamic modeling and control of constrained manipulation systems," *Robot. Autom. Mag.*, vol. 4, no. 2, pp. 27–44, Jun. 1997.

[17] G. E. Hovland and B. J. McCarragher, "Hidden Markov models as a process monitor in robotic assembly," *Int. J. Robot. Res.*, vol. 17, no. 2, pp. 153–168, Feb. 1998.

[18] J. De Schutter, H. Bruyninckx, S. Dutré, J. De Geeter, J. Katupitiya, S. Demey, and T. Lefebvre, "Estimating first-order geometric parameters and monitoring contact transitions during force-controlled compliant motions," *Int. J. Robot. Res.*, vol. 18, no. 12, pp. 1161–1184, Dec. 1999.

[19] E. Staffetti, "Analysis of rigid body interactions for compliant motion tasks using the Grassmann–Cayley algebra," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 1, pp. 80–93, Jan. 2009.

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

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

[22] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Online statistical model recognition and state estimation for autonomous compliant motion," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 1, pp. 16–29, Feb. 2005.

[23] L. Mihaylova, T. Lefebvre, E. Staffetti, H. Bruyninckx, and J. De Schutter, "Contact transitions tracking during force-controlled compliant motion using an interacting multiple model estimator," *Inf. Secur.*, vol. 9, pp. 114–129, 2002.

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

[25] A. Doucet, N. J. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Process.*, vol. 49, no. 3, pp. 613–624, Mar. 2001.

[26] P. Slaets, T. Lefebvre, J. Rutgeerts, H. Bruyninckx, and J. De Schutter, "Incremental building of a polyhedral feature model for programming by human demonstration of force-controlled tasks," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 20–33, Feb. 2007.

[27] L. J. Everett, R. Ravuri, R. A. Volz, and M. Skubic, "Generalized recognition of single-ended contact formations," *IEEE Trans. Robot. Autom.*, vol. 15, no. 5, pp. 829–836, Oct. 1999.

[28] M. Skubic and R. A. Volz, "Identifying single-ended contact formations from force sensor patterns," *IEEE Trans. Robot. Autom.*, vol. 16, no. 5, pp. 597–603, Oct. 2000.
[29] E. Cervera, A. P. Del Pobil, E. Marta, and M. A. Serna, "Perception-based learning for motion in contact in task planning," *J. Intell. Robot. Syst.*, vol. 17, no. 3, pp. 283–308, Nov. 1997.
[30] K. Hara and R. Yokogawa, "Recognition of state in peg-in-hole by fuzzy schema," *J. Adv. Autom. Technol.*, vol. 4, no. 3, pp. 134–139, 1992.
[31] J. Xiao and R. Volz, "On replanning for assembly tasks using robots in the presence of uncertainties," in *Proc. Int. Conf. Robot. Autom.*, 1989, pp. 638–645.
[32] T. Lozano-Perez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *Int. J. Robot. Res.*, vol. 3, no. 1, pp. 3–24, Mar. 1984.
[33] J. Rosell, L. Basañez, and R. Suárez, "Predicting planar motion behaviour under contact uncertainty," *Adv. Robot.*, vol. 19, no. 5, pp. 567–590, 2005.
[34] R. Voyles, J. Morrow, and P. Khosla, "Towards gesture-based programming: Shape from motion primordial learning of sensorimotor primitives," *J. Robot. Auton. Syst.*, vol. 22, no. 3/4, pp. 361–375, Dec. 1997.
[35] R. Voyles, J. Morrow, and P. Khosla, "Gesture-based programming for robotics: Human augmented software adaptation," *IEEE Intell. Syst.*, vol. 14, no. 6, pp. 22–29, Nov./Dec. 1999.
[36] V. Kruger, D. Kragic, A. Ude, and C. Geib, "The meaning of action: A review on action recognition and mapping," *Adv. Robot.*, vol. 21, no. 13, pp. 1473–1501, 2007.
[37] M. Pardowitz, S. Knoop, R. Dillmann, and R. D. Zöllner, "Incremental learning of tasks from user demonstrations, past experiences, and vocal comments," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 2, pp. 322–332, Apr. 2007.
[38] D. Kulic, W. Takano, and Y. Nakamura, "Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden Markov chains," *Int. J. Robot. Res.*, vol. 27, no. 7, pp. 761–784, Jul. 2008.
[39] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 2, pp. 286–298, Apr. 2007.
[40] S. Ekvall and D. Kragic, "Robot learning from demonstration: A task-level planning approach," *Int. J. Adv. Robot. Syst.*, vol. 5, no. 3, pp. 223–234, 2008.
[41] F. Guenter, M. Hersch, S. Calinon, and A. Billard, "Reinforcement learning for imitating constrained reaching movements," *Adv. Robot.—Special Issue Imitative Robots*, vol. 21, no. 13, pp. 1521–1544, 2007.
[42] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. New York: Springer-Verlag, 2008, ch. 59.
[43] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, vol. 82, pp. 34–45, 1960.
[44] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.
[45] J. M. Selig, *Geometric Fundamentals of Robotics*. New York: Springer-Verlag, 2005.
[46] J. H. Friedman, "Stochastic gradient boosting," *Comput. Stat. Data Anal.*, vol. 38, no. 4, pp. 367–378, Feb. 2002.
[47] P. Bühlmann and T. Hothorn, "Boosting algorithms: Regularization, prediction and model fitting," *Stat. Sci.*, vol. 22, pp. 477–505, 2007.
[48] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2001.
[49] P. Tang and J. Xiao, "Automatic generation of high-level contact state space between 3-D curved objects," *Int. J. Robot. Res.*, vol. 27, no. 7, pp. 832–854, Jul. 2008.

Stefano Cabras received the M.Sc. degree in statistics from the Carnegie Mellon University, Pittsburgh, PA, and the Ph.D. degree in applied statistics from the Università degli Studi di Firenze, Florence, Italy.

He is currently a Researcher with the Department of Mathematics, Università degli Studi di Cagliari, Cagliari, Italy. He has also been with the Universidad Carlos III de Madrid, Madrid, Spain. His main research interests are in objective Bayesian statistics and in nonparametric statistics.

María Eugenia Castellanos received the M.Sc. degree in mathematics from the Universidad de Valencia, Valencia, Spain, and the Ph.D. degree in statistics from the University Miguel Hernández de Elche, Elche, Spain, where she has also been working.

She is currently a Professor with the Department of Statistics and Operational Research, Universidad Rey Juan Carlos, Móstoles, Spain. Her research interests are in Bayesian and nonparametric statistics.

Ernesto Staffetti received the M.Sc. degree in electronic engineering from the Università degli Studi di Roma "La Sapienza," Rome, Italy, and the Ph.D. degree in electronic engineering from the Universitat Politècnica de Catalunya, Barcelona, Spain.

He is currently a Professor with the Department of Statistics and Operational Research, Universidad Rey Juan Carlos, Móstoles, Spain. He has also been with the Katholieke Universiteit Leuven, Louvain, Belgium, with the University of North Carolina at Charlotte, Charlotte, and with the Spanish Council of Scientific Research, Barcelona. His research interests include pattern recognition in computer vision, motion planning and control in robotics, and aircraft trajectory optimization in aeronautics.