# Integration of Planning and Execution in Force Controlled Compliant Motion

Wim Meeussen [a,*] Ernesto Staffetti [b] Herman Bruyninckx [a]
Jing Xiao [c] Joris De Schutter [a]

[a] *Department of Mechanical Engineering, Katholieke Universiteit Leuven Celestijnenlaan 300B, B3001 Leuven (Heverlee), Belgium.*

[b] *Department of Statistics and Operation Research, University Rey Juan Carlos Calle Tulipan s/n E-28933 - Madrid , Spain*

[c] *Computer Science Department, University of North Carolina at Charlotte 9201 University City Blvd, Charlotte, NC 28223, USA*

**Abstract**

This paper presents the *Compliant Task Generator:* a new approach for the automatic conversion of a geometric contact path into a force based task specification. A contact path planner generates a sequence of six-dimensional poses and corresponding contact formations, while a hybrid robot controller expects a desired wrench, twist and the local wrench and twist subspaces. Our approach automatically converts a geometric path description into a force based tasks specification for the hybrid controller, based on a user specified input of the magnitudes and the norms of the desired contact force and execution speed. The approach applies to all contact motions between known polyhedral objects, and is verified in real world experiments.

*Key words:* compliant motion control, compliant path planning, hybrid force control

## 1 Introduction

Compliant motion task are manipulation task that involve contacts between the object manipulated by a robot and the environment in which it operates. To cope with uncertainties these operations must be carried out using passive

* Corresponding author.
*Email address:* wim.meeussen@mech.kuleuven.be (Wim Meeussen).

or active force control. Indeed small errors in the object models can give rise to high interaction forces. Whereas passive force control relies on compliance elements placed at the wrist of the robot, in active force control the robot controller modifies the trajectory depending on the forces arising during the interaction [14].

To specify force controlled robot tasks M. T. Mason introduced the *Task Frame Formalism* (TFF) [30] an intuitive and manipulator independent formalism for the specification of force controlled robot tasks in the *Hybrid Control Paradigm* (HCP) [35]. Bruyninckx and De Schutter made an extensive catalog of TFF models and specifications [3]. While the TFF is useful to specify many elementary contact tasks [21], it cannot cope with more complex operations involving multiple simultaneous contacts, even in the case of polyhedral objects; the TFF is limited to translational and rotational components along the same axes, orthogonal to each other. 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 [13]. However, these approaches themselves have no "intelligence" such as planning or advanced sensor processing capabilities, but fully rely on human intervention and intuition to specify a task compatible with the framework. Indeed, 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 and changes in contact formations, this can prove to be extremely difficult.

Different approaches exist to automatically generate the desired sequence of contact formations in such a case. *Programming by Human Demonstration* [6, 17, 32, 36] gathers wrench, pose and contact data about a task, while a human demonstrates the task, in a virtual or real environment using for instance a haptic device or a demonstration tool. A different approach, as used in this paper, involves a geometric planner that automatically generates a path in the contact space of the manipulated object and its environment based on their geometric models. Xiao and Ji developed such a compliant planner [19]. Due to the complexity of the problem this planner finds such a path in two stages. In the first one the planner automatically generates the contact state space between two arbitrary polyhedral objects in terms of a contact state graph [39]. Even between two simple polyhedral objects, hundreds of contact formations are possible. In this graph the possible contact states and their adjacency relations are represented. In the second stage the planner generates a compliant path by compliant interpolation between neighboring contact states. However, this path only contains geometric and topological information. For the complete specification of the compliant task the forces to be applied to the environment along the path have to be described.

In this paper we present an approach to automatically generate a task speci-

fication for a hybrid controller, based on the output of the geometric planner. Therefore the user specifies desired magnitudes for the contact wrench and the manipulator twist. This allows us to generate a complex task plan based on the known geometric models of the objects, and execute it on a real robot manipulator under active force control without expert user intervention. The method has been implemented and real world experiments have been carried out to validate it. This work is complementary to —and can be integrated with— previous work of our research group, which focussed on the identification of contact states and the estimation of geometrical parameters using iterative stochastic estimation tools such as Kalman filters or particle filters [17,24,32]. To improve the identification and estimation, in [26] the active sensing problem is formulated and decoupling it into smaller optimization problems.

The remainder of the paper is organized as follows. Section 2 briefly reviews the concept of contact formations and the contact state graph. The next section discusses the output of the compliant planner and the input to the hybrid controller, in other words the planner and controller primitives. Section 4 describes the automatic generation of the controller primitives, using the planner primitives and information about the desired contact force level and execution speed. Section 5 explains the internals of the hybrid controller. Section 6 discusses the invariance and the robustness of the method. The experimental setup and the obtained results are discussed in Section 7. Finally, Section 8 contains the conclusions together with future extensions and improvements of the method.

## 2 Review: Contact Formations and Contact State Graph

### *2.1 Contact Formations*

The notion of *principal contacts* (PCs) was introduced [37] 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. 1 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

[1] The vertex-vertex, vertex-edge and edge-vertex 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.

contacting edges at an edge-edge PC.

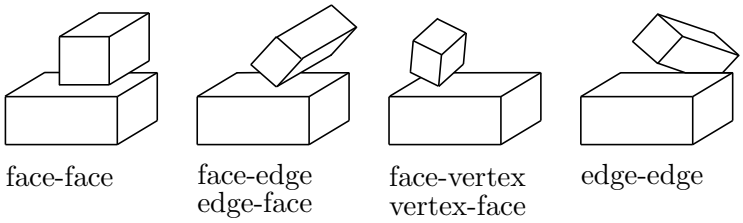



Fig. 1. Every non-degenerate contact formation between two polyhedral objects can be described by a combination of the six non-degenerate Principal Contacts (PCs)

A general contact state between two objects can be characterized topologically by the set of PCs formed, called a *contact formation* (CF). Each pose of two objects, this is their relative location in space, compliant to the constraints of a CF is called a *CF-compliant pose*, denoted by $\boldsymbol{X}$. Any motion formed by a sequence of CF-compliant poses is called a *CF-compliant motion*. We use a non-minimal representation for $\boldsymbol{X}$, defined by a homogeneous transformation matrix containing a rotation matrix $\boldsymbol{R}_{3\times3}$ and a translation vector $\boldsymbol{p}_{3\times1}$:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{R}_{3\times3} & \boldsymbol{p}_{3\times1} \\ \boldsymbol{0}_{1\times3} & 1 \end{bmatrix}. \tag{1}$$

A PC can be decomposed into one or more *Elementary Contacts* (ECs), providing a lower level description of the contact formation, as shown in Fig. 2. An EC is a point contact and is associated with a *contact point* and a *contact normal*. The three types of ECs (face-vertex, vertex-face and edge-edge) are shown in the two examples at the right of Fig. 1. For the decomposition of a PC into ECs, we use the *contacting area* of the PC, as shown by the gray areas in Fig. 2. 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). We position the ECs at the boundary points of the (polygonal) contacting area. The number of ECs at a PC, depends on the type of PC and the compliant pose $\boldsymbol{X}$ of the contacting objects at the PC. This is illustrated by the last three examples in Fig. 2, which all show the same two objects in the same face-face PC, but at a different pose $\boldsymbol{X}$. The contacting area at the first example has 3 *EC*s at its boundary points, the second has 4 *EC*s, while the last has 6 *EC*s.

### *2.2 Contact State Graph*

Xiao and Ji developed a divide-and-merge approach [38,39] to generate a compact, simplified representation of the contact state space between two poly-

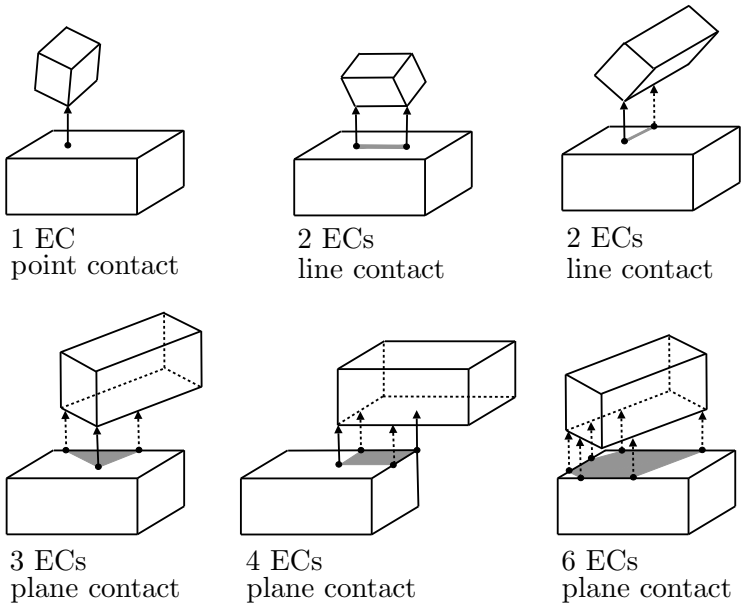



Fig. 2. A principal contact (PC) can be decomposed into one or more elementary contacts (ECs), which are associated with a contact point and a contact normal. The dotted arrows indicate the edge-edge ECs, and the full arrows indicate the vertex-face or face-vertex ECs.

hedral objects, as a *contact state graph* $G$. In $G$ a node represents a contact formation, and an arc connecting two nodes represents the adjacency relationship between the contact formations of the nodes. Two contact formations $CF_i$ and $CF_j$ are adjacent if a compliant motion from a $CF_i$-compliant pose to $CF_j$-compliant pose exists, which only includes $CF_i$ and $CF_j$-compliant poses. Fig. 3 shows an example of a contact state graph containing 9 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. It was implemented with algorithms to automatically generate a complete contact state graph, given the geometric models of two polyhedral objects and a locally most constrained pose. A contact state graph can contain hundreds of contact formations even for two simple polyhedral objects.

## 3 Compliant Planner and Hybrid Controller Primitives

This section first describes the output primitives of the compliant motion planner, which correspond to the input for the *Compliant Task Generator*. Next, the input primitives for the hybrid controller are discussed, which correspond to the output of the *Compliant Task Generator*. This is schematically represented in Fig. 4.

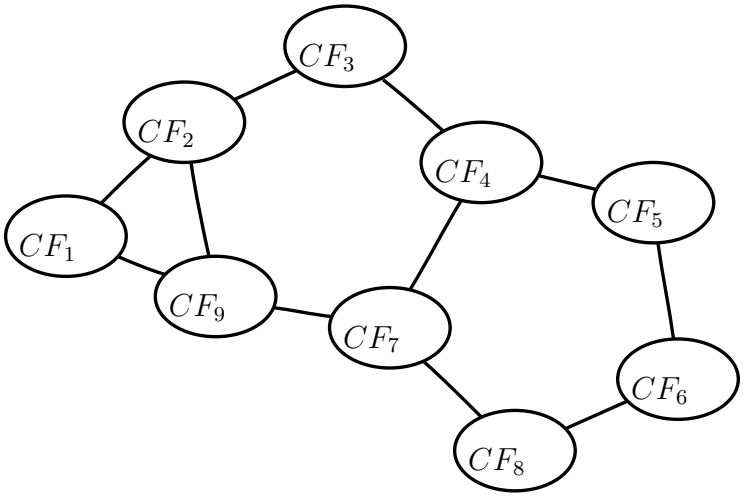



Fig. 3. The contact state graph shows all possible contact formations (nodes) and transitions between neighboring contact formations (arcs).

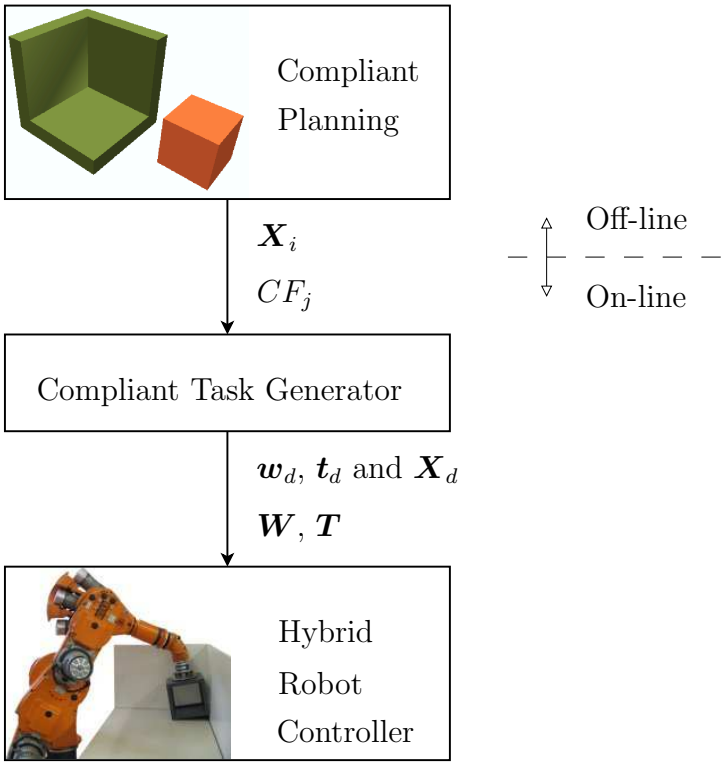



Fig. 4. Compliant task generation: the primitives of the off-line compliant path planner are converted into primitives for the hybrid robot controller.

### *3.1 Compliant Planner Primitives*

Compliant motion planning can be defined as follows [22]: given a $CF_1$-compliant start pose $\boldsymbol{X}_1$ and a $CF_m$-compliant end pose $\boldsymbol{X}_n$, find a path between them in the contact space of the manipulated object and the environment. The path

must be collision-free for the manipulator. Fig. 5 shows a simplified representation of the motion planning problem, for a 3-dimensional configuration space of two contacting objects. The dotted line represents the searched compliant path connecting $\boldsymbol{X}_1$ and $\boldsymbol{X}_n$. Ji and Xiao developed a two-level geometric approach to tackle this problem [19]. First a high-level graph search in the contact state graph results in a sequence of contact transitions between adjacent contact formations, connecting $CF_1$ and $CF_m$. Then a low-level motion planner, based on extending the *Probabilistic Roadmap Paradigm* [20], is used to interpolate between the contact formations of the high level planner. Within each contact formation $CF_j$ of the high level path, with $j = 1 \dots m$, it produces a sequence of $CF_j$-compliant poses. The first pose of $CF_j$ connects to the last pose of $CF_{j-1}$, and the last pose of $CF_j$ connects to the first pose of $CF_{j+1}$, resulting in the desired compliant path.

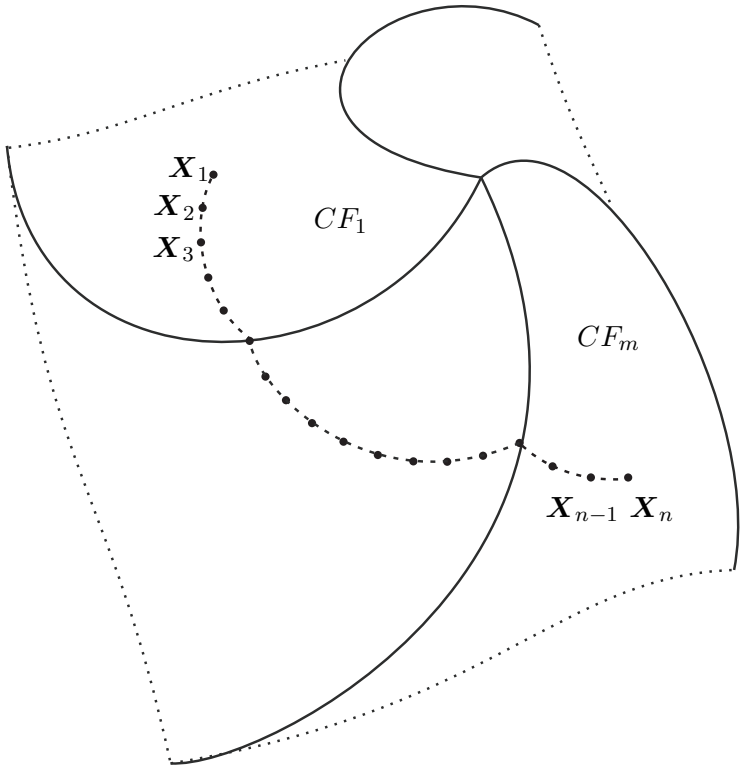



Fig. 5. A simplified 3-dimensional configuration space representation of a compliant path from $\boldsymbol{X}_1$ at $CF_1$, to $\boldsymbol{X}_n$ at $CF_m$.

The output primitives of this compliant planner only contain geometrical and topological information in the form of a sequence of poses $\boldsymbol{X}_1 \dots \boldsymbol{X}_n$ and their corresponding contact formations $CF_1 \dots CF_m$. Each two poses $\boldsymbol{X}_i$ and $\boldsymbol{X}_{i+1}$ are at the same or at neighboring contact formations, as shown in Fig. 5.

The controller primitives are the task specification for the controller. They specify the same path as the planner primitives, but in a form the controller understands. In our approach, we use the controller primitives of the *Hybrid Control Paradigm* (HCP) [30, 35], one of the three major force control paradigms together with *Impedance Control* [18] and the *Parallel Force Control* [7]. The HCP assumes a *geometric* interaction model. In HCP terminology, an object in contact with its environment has $h$ degrees of freedom (DOF) which are wrench controlled, and $(6-h)$ DOF which are twist controlled. A wrench is a six-vector containing a force and a torque:

$$\boldsymbol{w} = \begin{bmatrix} f_x & f_y & f_z & \tau_x & \tau_y & \tau_z \end{bmatrix}^T = \begin{bmatrix} \boldsymbol{f}^T & \boldsymbol{\tau}^T \end{bmatrix}^T. \tag{2}$$

A twist is a six-vector containing a translational and a rotational velocity:

$$\boldsymbol{t} = \begin{bmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T = \begin{bmatrix} \boldsymbol{v}^T & \boldsymbol{\omega}^T \end{bmatrix}^T. \tag{3}$$

The $h$ wrench controlled DOF are described by a $h$-dimensional wrench controlled subspace $\mathcal{W}$ and the $(6-h)$ twist controlled DOF are described by a $(6-h)$-dimensional twist controlled subspace $\mathcal{T}$. In the rest of the paper we use the terms wrench space and twist space for the wrench and twist controlled subspaces. The wrench and twist space model the first order kinematic constraints of a contact between two objects at a pose $\boldsymbol{X}$. All possible wrenches of $\mathcal{W}$ are reciprocal to all possible twists of $\mathcal{T}$ [27]. This means that the ideal [2] contact wrenches produce no work against the twists allowed by the contact.

The controller primitives are a desired wrench $\boldsymbol{w}_d$ to specify the contact wrench between the manipulated object and its environment, a desired twist $\boldsymbol{t}_d$ to specify the velocity of the manipulated object in its environment, a desired pose $\boldsymbol{X}_d$, and the local wrench and twist spaces $\boldsymbol{W}$ and $\boldsymbol{T}$. Fig. 4 gives a schematic overview of the primitives used by the compliant planner and the hybrid controller.

## 4 Compliant Task Generator

This section describes the core of our approach, the automatic conversion of a geometric path generated by the compliant path planner ($\boldsymbol{X}_1 \ldots \boldsymbol{X}_n$ and

[2] The Hybrid Control Paradigm models frictionless contacts.

$CF_1 \dots CF_m$), into a force based task specification for the hybrid controller ($\boldsymbol{w}_d$, $\boldsymbol{t}_d$, $\boldsymbol{X}_d$, $\boldsymbol{W}$ and $\boldsymbol{T}$). The *direction* of the desired twist $\boldsymbol{t}_d$ and the desired wrench $\boldsymbol{w}_d$ are derived from the planner output: $\boldsymbol{T}$ is defined by the pose setpoints $\boldsymbol{X}_1 \dots \boldsymbol{X}_n$, and $\boldsymbol{w}_d$ is defined by the contact formation setpoints $CF_1 \dots CF_m$. The *magnitudes* of the desired twist and wrench are specified by the user because the planner provides no information to derive a magnitude from. However, it is not possible to directly specify the magnitude of a twist or wrench, because a natural norm in the twist and wrench spaces does not exist; this is a natural function to map a twist or a wrench into a nonnegative number [28, 34]. The norm does not exist due to the nonexistence of a left and right invariant Riemennian metric in $SE(3)$, this is a metric that is invariant to changes of inertial and body-fixed reference frames, respectively. However the nonexistence of the invariant metric should not be interpreted as the nonexistence of a frame-invariant way to measure a twist or a wrench. For instance, the kinetic energy metric defined by $\boldsymbol{t}^t\boldsymbol{M}\boldsymbol{t}$, in which $\boldsymbol{M}$ is the generalized intertia matrix is leftinvariant and its invariance under a change of the body-fixed reference frame is assured if the matrix representation of the physical mapping $\boldsymbol{M}$ is properly transformed. Therefore, in addition to the two magnitudes, the user also needs to specify two norms to give a meaning to each of the specified wrench and twist magnitudes.

### *4.1 Specification of norms and magnitudes*

The desired contact force level and execution speed are specified by the user in the form of a *magnitude* and a *norm* for the desired twist and for the desired wrench. To obtain a specification that is invariant with respect to changes in reference frame, physical units or scale, we choose two norms with a physical meaning [1, 8, 15]. The norm for the desired twist is defined by a generalized inertia matrix $\boldsymbol{M}$, while the norm for the desired wrench is defined by a generalized compliance matrix $\boldsymbol{C}$. When the inertia and compliance matrices define a norm for the desired twist and wrench, the specified magnitudes have the physical meaning of a kinetic and a potential energy:

$$\|\boldsymbol{t}_d\|_{\boldsymbol{M}} = \frac{\boldsymbol{t}_d^t\boldsymbol{M}\boldsymbol{t}_d}{2} = E_{kin}, \tag{4}$$

$$\|\boldsymbol{w}_d\|_{\boldsymbol{C}} = \frac{\boldsymbol{w}_d^t\boldsymbol{C}\boldsymbol{w}_d}{2} = E_{pot}. \tag{5}$$

The hybrid controller uses two separate feedback loops, one for the measured twist and one for the measured wrench. The former controls the specified magnitude of the twist, and the latter controls the specified magnitude of the wrench. The magnitudes and norms can be directly specified by an inertia, a compliance and two energy levels, or indirectly by a magnitude for all

components of the desired twist and wrench.

### *4.1.1 Direct specification*

The inertia and compliance can be chosen to reflect the true dynamical properties of the system defined by the manipulator, the manipulated object and the environment. In this case the specified magnitudes for the desired twist and wrench reflect the real kinetic and potential energy stored in the system during the execution of the compliant path.

The inertia and compliance can also be chosen as an arbitrary norm for the desired wrench and twist, invariant with respect to changes of reference frame, physical units of scale. In this case, the inertia and compliance represent a virtual system. During the execution of the compliant path, the specified magnitudes have the meaning of energy levels in the virtual system.

### *4.1.2 Indirect specification*

In some cases it is more intuitive to define the magnitudes $\bar{v}$ and $\bar{\omega}$ of the translational and rotational components of the desired twist, and the magnitudes $\bar{f}$ and $\bar{\tau}$ of the force and torque components of the desired wrench. The magnitudes of the torque and translational velocity are specified at a given point on the manipulated object, because only when expressed at a certain point they have a meaning. From the specification of these magnitudes at a given point, we derive two invariant norms as an inertia and compliance matrix, and two magnitudes as the kinetic and potential energy.

The inertia matrix is defined by:

$$\boldsymbol{M} = m \cdot \begin{bmatrix} \boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & l_t^2 \cdot \boldsymbol{I}_{3\times3} \end{bmatrix}, \tag{6}$$

where $m = 1\ [kg]$ and where $l_t$ is the characteristic length for the twist:

$$l_t = \frac{\bar{v}}{\bar{\omega}}\ [m], \tag{7}$$

and the magnitude of the twist is defined by the kinetic energy:

$$E_{kin} = \bar{v}^2 \cdot m. \tag{8}$$

In the same way, the compliance matrix is defined by:

$$\boldsymbol{C} = c \cdot \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & 1/l_w^2 \cdot \boldsymbol{I}_{3\times3} \end{bmatrix}, \tag{9}$$

where $c = 1\ [m/N]$ and where $l_w$ is the characteristic length for the wrench:

$$l_w = \frac{\bar{\tau}}{\bar{f}}\ [m], \tag{10}$$

and the magnitude of the wrench is defined by the potential energy:

$$E_{pot} = \bar{f}^2 \cdot c. \tag{11}$$

### *4.2 Desired Twist, Pose and Wrench*

The desired twist $\boldsymbol{t}_d$, pose $\boldsymbol{X}_d$ and wrench $\boldsymbol{w}_d$ are calculated using the set of poses $\boldsymbol{X}_1 \ldots \boldsymbol{X}_n$ and their corresponding contact formations $CF_1 \ldots CF_m$, together with the specified magnitudes and norms for the desired twist and wrench. [3]

#### *4.2.1 Twist*

The desired twist $\boldsymbol{t}_d$ at time $t \in [t_i, t_{i+1}[$, to move from $\boldsymbol{X}_i$ to $\boldsymbol{X}_{i+1}$ with a magnitude $E_{kin}$, is calculated in two steps. First we define a constant twist $\boldsymbol{t}_i$ to move from $\boldsymbol{X}_i$ to $\boldsymbol{X}_{i+1}$:

$$\boldsymbol{t}_i = \begin{bmatrix} \boldsymbol{v}_i^T & \boldsymbol{\omega}_i^T \end{bmatrix}^T, \tag{12}$$

and

$$\begin{bmatrix} [\boldsymbol{\omega}_i\times] & \boldsymbol{v}_i \\ \boldsymbol{0} & 0 \end{bmatrix} = \log\left(\boldsymbol{X}_i^{-1} \cdot \boldsymbol{X}_{i+1}\right) / \left(t_{i+1} - t_i\right). \tag{13}$$

The logarithm of a homogeneous transformation matrix [33, Section 3.2] is used to interpolate between two discrete setpoints of the planner. The $[\ \times]$ operator is defined in [2]. In the second step we scale this constant twist $\boldsymbol{t}_i$ to the desired twist $\boldsymbol{t}_d$, so that its magnitude equals $E_{kin}$:

$$\boldsymbol{t}_d = s_t \cdot \boldsymbol{t}_i. \tag{14}$$

[3] Note that for all equations in this section where twists and wrenches are added or multiplied with an inertia or compliance matrix, it is necessary that both share the same reference frame and reference point. Transformations of reference frames and reference points are discussed in [2].

The scaling factor $s_t$ has no units and is defined by the magnitude $E_{kin}$:

$$E_{kin} = \frac{(s_t \cdot \boldsymbol{t}_i)^T \cdot \boldsymbol{M} \cdot (s_t \cdot \boldsymbol{t}_i)}{2}. \tag{15}$$

This results in:

$$s_t = \sqrt{\frac{2 \cdot E_{kin}}{\boldsymbol{t}_i^T \cdot \boldsymbol{M} \cdot \boldsymbol{t}_i}}. \tag{16}$$

The direction of the desired twist changes discontinuously between two planner setpoints, while its magnitude remains constant. Replacing the linear interpolation between the pose setpoints by a smooth interpolation would avoid the discontinuous changes of the direction.

### *4.2.2 Pose*

The desired pose $\boldsymbol{X}_d$ at time $t \in [t_i, t_{i+1}[$, between $\boldsymbol{X}_i$ and $\boldsymbol{X}_{i+1}$ is defined by the integration of the desired twist $\boldsymbol{t}_d$, using the exponential function [33, Section 3.2]

$$\boldsymbol{X}_d = \boldsymbol{X}_i \cdot \exp\left(\begin{bmatrix} [\boldsymbol{\omega}_d\times] & \boldsymbol{v}_d \\ \boldsymbol{0} & 0 \end{bmatrix}\right) \cdot (t - t_i)\,, \tag{17}$$

with

$$\boldsymbol{t}_d = \begin{bmatrix} \boldsymbol{v}_d^T & \boldsymbol{\omega}_d^T \end{bmatrix}^T. \tag{18}$$

### *4.2.3 Wrench*

The desired wrench $\boldsymbol{w}_d$ at time $t \in [t_i, t_{i+1}[$, is calculated in two steps. First we decompose all PCs of the contact formation into ECs, as described in Section 2. The number of ECs depends on $CF_{j+1}$ at the pose $\boldsymbol{X}_{i+1}$, and is called $p$. We position a unit wrench vector $\boldsymbol{w}_{unit_k}$ at each $EC_k$, with $k = 1 \ldots p$. In a local frame with the origin at the contact point and the x-axis along the contact normal, oriented from the environment towards the manipulated object, each unit wrench is represented by:

$$\boldsymbol{w}_{unit_k} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T. \tag{19}$$

The sum (see footnote 3) of all unit wrenches is called $\boldsymbol{w}_i$, and defines the direction of the desired wrench:

$$\boldsymbol{w}_i = \sum_{k=1}^{p} \boldsymbol{w}_{unit_k}. \tag{20}$$

In the second step we scale this wrench $\boldsymbol{w}_i$ to the desired wrench $\boldsymbol{w}_d$, so that its magnitude equals $E'_{pot}$:

$$\boldsymbol{w}_d = s_w \cdot \boldsymbol{w}_i. \tag{21}$$

The scaling factor $s_w$ has no units and is defined by the magnitude $E'_{pot}$:

$$E'_{pot} = \frac{(s_w \cdot \boldsymbol{w}_i^T) \cdot \boldsymbol{C} \cdot (s_w \cdot \boldsymbol{w}_i)}{2}. \tag{22}$$

This results in:

$$s_w = \sqrt{\frac{2 \cdot E'_{pot}}{\boldsymbol{w}_i^T \cdot \boldsymbol{C} \cdot \boldsymbol{w}_i}}. \tag{23}$$

The magnitude $E'_{pot}$ is chosen to depend on the *contact strength* of the contact formation $CF_{j+1}$ at $\boldsymbol{X}_{i+1}$ and the specified magnitude $E_{pot}$.:

$$E'_{pot} = \dim(\mathcal{W}) \cdot E_{pot}. \tag{24}$$

The contact strength is equal to the number of wrench controlled degrees of freedom, denoted by $\dim(\mathcal{W})$. This results in a "natural" magnitude for the desired wrench, where at a point contact (vertex-face, face-vertex or edge-edge) it is smaller than at a line contact (edge-face or face-edge), and at a line contact it is smaller than at a plane contact (face-face):

The desired wrench changes discontinuously between two different CFs, where different contact constraints apply.

### *4.3 Defining Wrench and Twist Controlled Subspaces*

To obtain bases $\boldsymbol{W}$ and $\boldsymbol{T}$ for the wrench space and the reciprocal twist space at time $t \in [t_i, t_{i+1}]$, when moving from $\boldsymbol{X}_i$ at $CF_j$, to $\boldsymbol{X}_{i+1}$ at $CF_{j+1}$, we choose to use the contact information of $CF_{j+1}$. This choice allows us to break unilateral contact constraints under velocity control, and add unilateral contact constraints under force control. When breaking a contact constraint, the twist space of $CF_{j+1}$ is higher dimensional than the twist space $CF_j$, allowing a velocity controlled motion to break the contact. When creating a new contact constraint, the wrench space of $CF_{j+1}$ is higher dimensional than the wrench space of $CF_j$, allowing a force controlled motion to add the contact.

To obtain a base for the wrench and twist spaces at $CF_{j+1}$, we use all unit wrenches $\boldsymbol{w}_{unit_k}$ of $CF_{j+1}$, with $k = 1 \ldots p$. The $p$ unit wrenches form a set of vectors that span $\mathcal{W}$; they are represented by:

$$\boldsymbol{W}_{CF} = \begin{bmatrix} \boldsymbol{w}_{unit_1} & \ldots & \boldsymbol{w}_{unit_p} \end{bmatrix}. \tag{25}$$

Using the *singular value decomposition* (SVD) of $\boldsymbol{W}_{CF}$, we obtain a base $\boldsymbol{W}$ and a base $\boldsymbol{T}$, representing the wrench and the twist space, respectively:

$$\boldsymbol{W}_{CF} = \boldsymbol{U}_{6\times 6} \cdot \boldsymbol{S}_{6\times 6} \cdot \boldsymbol{V}^T_{6\times p}, \tag{26}$$

where $\boldsymbol{V}$ and $\boldsymbol{U}$ are orthonormal, and

$$\boldsymbol{U} = \left[\boldsymbol{W}\ \boldsymbol{T}\right] \tag{27}$$

$$= \left[\boldsymbol{w}_1 \ldots \boldsymbol{w}_h\ \boldsymbol{t}_1 \ldots \boldsymbol{t}_{(6-h)}\right]. \tag{28}$$

$\boldsymbol{S}$ is a diagonal matrix containing the singular values $s_1 \ldots s_6$. The $h$ columns of $\boldsymbol{U}$ that correspond to singular values that are greater than a threshold $\epsilon \approx 0$ span the wrench space, while the $(6-h)$ columns that correspond to smaller singular values span the twist space:

$$s_1 \geq \ldots \geq s_h > \epsilon > s_{h+1} \geq \ldots \geq s_6 \geq 0. \tag{29}$$

The columns of matrix $\boldsymbol{U} = [\boldsymbol{W}\ \boldsymbol{T}]$, calculated by the numerical SVD algorithm, are orthogonal to each other[4]. However, all columns of the wrench space should be reciprocal to all columns of the twist space [16]. This means that any possible twist $\boldsymbol{t}$ of $\mathcal{T}$ produces no work in the interaction with any possible wrench $\boldsymbol{w}$ of $\mathcal{W}$:

$$\boldsymbol{W}^T \cdot \boldsymbol{T} = \boldsymbol{0}. \tag{30}$$

To interpret the orthogonal columns of $\boldsymbol{U}$ as reciprocal wrenches and twists, we assign compatible units to forces, torques, rotational velocities and translational velocities.

## 5 Implementation of the Hybrid Controller

This section discusses the hybrid controller that converts the desired twist $\boldsymbol{t}_d$, the desired pose $\boldsymbol{X}_d$ and the desired wrench $\boldsymbol{w}_d$ to a control twist $\boldsymbol{t}_c$ for the manipulator. The approach applies to a velocity controlled manipulator like in [10], which is industrial practice. A proportional feedback loop in the twist space controls the desired twist $\boldsymbol{t}_d$ and pose $\boldsymbol{X}_d$, while a second proportional feedback loop in the wrench space controls the desired wrench $\boldsymbol{w}_d$.

[4] The notion of orthogonality is often used to interpret the reciprocity condition, but is not applicable because orthogonality can only be defined between elements of the same space, and twist and wrench spaces are distinct vector spaces.

### 5.1 Pose and Twist Controller in $\mathcal{T}$

The desired twist $\boldsymbol{t}_d$ is directly applied in the twist space, while the desired pose $\boldsymbol{X}_d$ is used together with the measured pose $\boldsymbol{X}_m$ as a pose feedback in the twist space. The pose feedback prevents the robot manipulator to drift away from the desired trajectory in the twist space. The desired twist and pose together define the resulting velocity for the manipulated object in the $(6-h)$-dimensional twist space $\mathcal{T}$. The contributions of the desired velocity and the pose feedback are represented using $(6-h)$-dimensional coordinate vectors $\boldsymbol{u}_t$ and $\boldsymbol{u}_X$:

$$\boldsymbol{u}_t = \boldsymbol{T}^{\dagger_{\boldsymbol{M}_c}} \cdot \boldsymbol{t}_d \tag{31}$$

$$\boldsymbol{u}_X = K_X^{FB} \cdot \boldsymbol{T}^{\dagger_{\boldsymbol{M}_c}} \cdot \boldsymbol{t}_\Delta^{FB}. \tag{32}$$

The pose difference $\boldsymbol{t}_\Delta^{FB}$ between the measured pose $\boldsymbol{X}_m$ and the desired pose $\boldsymbol{X}_d$ is calculated similar to (13), using:

$$\boldsymbol{t}_\Delta^{FB} = \begin{bmatrix} \Delta\boldsymbol{p}^{FB} & \Delta\boldsymbol{\theta}^{FB} \end{bmatrix}^T, \tag{33}$$

and

$$\begin{bmatrix} \left[\Delta\boldsymbol{p}^{FB}\times\right] & \Delta\boldsymbol{\theta}^{FB} \\ \boldsymbol{0} & 0 \end{bmatrix} = \log\left(\boldsymbol{X}_n^{-1} \cdot \boldsymbol{X}_d\right). \tag{34}$$

The scalar $K_X^{FB}$ with units $\frac{1}{s}$ represents the proportional pose feedback constant in $\mathcal{T}$. Matrix $\boldsymbol{T}^{\dagger_{\boldsymbol{M}_c}}$ is the weighted pseudo-inverse of $\boldsymbol{T}$, and the weighting matrix $\boldsymbol{M}_c$ is an inertia matrix:

$$\boldsymbol{T}^{\dagger_{\boldsymbol{M}_c}} = (\boldsymbol{T}^T \cdot \boldsymbol{M}_c \cdot \boldsymbol{T})^{-1} \cdot \boldsymbol{T}^T \cdot \boldsymbol{M}_c. \tag{35}$$

Using this weighted pseudo-inverse, we minimize the projection error when projecting the 6-dimensional twist and pose difference into the $(6-h)$-dimensional twist space. The norm for the twist has the physical meaning of kinetic energy in an object with mass distribution $\boldsymbol{M}_c$ [3]. The weighted pseudo inverse is calculated numerically using the SVD algorithm.

### 5.2 Wrench Controller in $\mathcal{W}$

The desired wrench $\boldsymbol{w}_d$ is used together with the measured wrench $\boldsymbol{w}_m$ as a wrench feedback in the wrench space. While the controller in the twist space uses an integral action to prevent drift in the twist space, the contact constraints prevent drift in the wrench space. Therefore there is no integral action used in the wrench space. In the $h$-dimensional wrench space $\mathcal{W}$ we use

a $h$-dimensional coordinate vector $\boldsymbol{u}_w$ to represent the desired rate of wrench change, resulting from the wrench feedback:

$$\boldsymbol{u}_w = K_w^{FB} \cdot \boldsymbol{W}^{\dagger \boldsymbol{C}_c} \cdot (\boldsymbol{w}_d - \boldsymbol{w}_m). \tag{36}$$

The scalar $K_w^{FB}$ with units $\frac{1}{s}$ represents the proportional force feedback constant in $\mathcal{W}$. Matrix $\boldsymbol{W}^{\dagger \boldsymbol{C}_c}$ is the weighted pseudo-inverse of $\boldsymbol{W}$, and the weighting matrix $\boldsymbol{C}_c$ is a compliance:

$$\boldsymbol{W}^{\dagger \boldsymbol{C}_c} = (\boldsymbol{W}^T \cdot \boldsymbol{C}_c \cdot \boldsymbol{W})^{-1} \cdot \boldsymbol{W}^T \cdot \boldsymbol{C}_c. \tag{37}$$

Using this weighted pseudo-inverse, we minimize the projection error when projecting the 6-dimensional wrench error into the $h$-dimensional wrench space. The norm for the wrench has the physical meaning of potential energy in an object with compliance $\boldsymbol{C}_c$.

### *5.3 Resulting Manipulator Twist*

The $(6-h)$-dimensional coordinate vectors $\boldsymbol{u}_t$ and $\boldsymbol{u}_X$, together define the resulting control twist $\boldsymbol{t}_c^t$ in the twist space, and can be directly applied by a velocity controlled manipulator:

$$\boldsymbol{t}_c^t = \boldsymbol{T} \cdot (\boldsymbol{u}_t + \boldsymbol{u}_X)\,. \tag{38}$$

The $h$-dimensional coordinate vector $\boldsymbol{u}_w$ defines the desired rate of wrench change at the manipulated object, and can only be applied by a velocity controlled manipulator through a compliance. The compliance $\boldsymbol{C}_c$ in the system defines the relation between a control twist $\boldsymbol{t}_c^w$ in the wrench space and the rate of wrench change:

$$\boldsymbol{t}_c^w = \boldsymbol{C}_c \cdot \boldsymbol{W} \cdot \boldsymbol{u}_w. \tag{39}$$

Combining the manipulator twist from the control loops in both the twist and wrench space:

$$\boldsymbol{t}_c = \boldsymbol{t}_c^w + \boldsymbol{t}_c^t \tag{40}$$

$$= \begin{bmatrix} \boldsymbol{C}_c \cdot \boldsymbol{W} & \boldsymbol{T} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{u}_w \\ \boldsymbol{u}_t + \boldsymbol{u}_X \end{bmatrix}, \tag{41}$$

results into the control twist $\boldsymbol{t}_c$ for the manipulator.

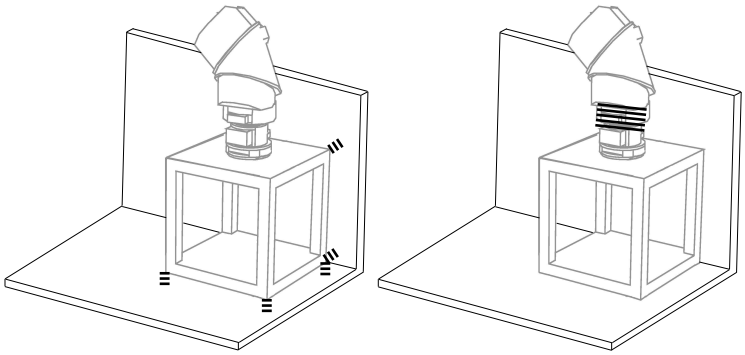



Fig. 6. The dominant compliance between the robot manipulator and the environment can be (a) the contact compliance, and (b) the compliance between the manipulator and the manipulated object.

## 6 Discussion

This section discusses the applicability of the approach, the invariance of the method, and the choice of the different parameters.

### *6.1 Applicability*

The compliant task generation method presented in this paper is integrated with a robot controller based on the hybrid control paradigm. The hybrid controller uses a separate closed feedback loop for the velocity and the force controlled degrees of freedom of the robot manipulator. This hybrid controller is capable of operating in the presence of small uncertainties on geometric parameters such as the location of the objects of the environment in which the robot operates or the location of the manipulated object with respect to the robot gripper. The approach is designed for a velocity controlled manipulator, and is applicable to plan compliant motion tasks involving convex as well as nonconvex polyhedral objects. It requires the knowledge of the geometric model of the manipulated object and the environment.

The compliant task generation method presented in this paper is intended to be integrated into a more complete compliant robot system capable of operating in the presence of large uncertainties on the geometric parameters [31]. Such a system includes components for the identification of the actual sequence of CFs that occur during the execution of the task. In this paper, uncertainties on the sequence of CFs are not considered; in particular it is assumed that the sequence of discrete states generated by the planner is the same as the actual sequence of CFs that occur during the execution of the task. Nevertheless, the closed force feedback loop in the adopted hybrid control

strategy gives this method some robustness to small geometric uncertainties as described in the next section.

### *6.2 Robustness to Geometric Uncertainty*

In an uncertain environment where the actual position and orientation of the environment slightly differ from the desired ones, the closed loop wrench controller corrects the motion of the manipulator so the actual CF corresponds to the desired CF. When for example a face-face CF is desired, but the error on the orientation of the environment is large enough to change the face-face CF into an edge-face CF, the measured wrench will include an undesired torque component. The closed loop wrench controller will then generate a motion for the manipulator to reduce this undesired torque component, and the face-face CF will be restored.

To undestand what the maximum uncertainty our method can cope with is, consider the motion between a pose $\boldsymbol{X}_i$ at a less constrained CF to a pose $\boldsymbol{X}_{i+1}$ at a more constrained CF. According to the presented approach, this motion is executed in the local twist and wrench spaces of $\boldsymbol{X}_{i+1}$. This means that the extra contact is added under active force control, because the wrench space of $\boldsymbol{X}_{i+1}$ is higher dimensional than the wrench space of $\boldsymbol{X}_i$. In the example in Fig. 7 the poses $\boldsymbol{X}_1$ to $\boldsymbol{X}_3$ contain one face-face contact, while pose $\boldsymbol{X}_4$ contains two face-face contacts. The motion from $\boldsymbol{X}_1$ to $\boldsymbol{X}_2$ to $\boldsymbol{X}_3$ assumes one face-face contact, and the rotations in the plane and translation perpendicular to the face of the next contact, are executed under velocity control. The motion from $\boldsymbol{X}_3$ to $\boldsymbol{X}_4$, where a new face-face contact is added, assumes two face-face contacts. Therefore, during the motion from $\boldsymbol{X}_3$ to $\boldsymbol{X}_4$, the translations and rotations in the plane are executed under active force control. The implication of this method is that the allowed geometric uncertainty is limited by the translational and rotational distance between $\boldsymbol{X}_3$ and $\boldsymbol{X}_4$. When the geometric uncertainty is higher, while adding the new face-face contact, it is possible that this new contact already occurs during the motion from $\boldsymbol{X}_2$ to $\boldsymbol{X}_3$, which is velocity controlled in the direction of the new contact. So, for the presented method to be robust, the geometric uncertainty must be smaller than the translational and rotational distance between the last pose $\boldsymbol{X}_i$ at a less constrained CF and the first pose $\boldsymbol{X}_{i+1}$ at a more constrained CF. This distance is directly defined by the step size of the compliant planner.

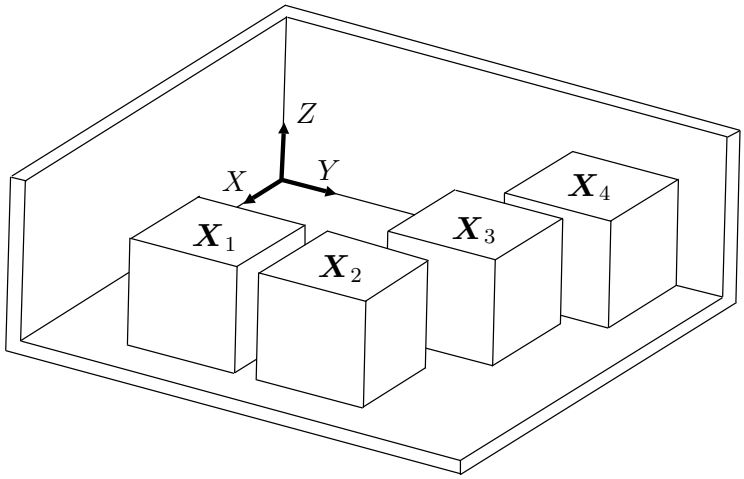



Fig. 7. Creating a new contact under active force control: The motion from $\boldsymbol{X}_1$ to $\boldsymbol{X}_2$ to $\boldsymbol{X}_3$ is velocity controlled in the horizontal plane. During the motion from $\boldsymbol{X}_3$ to $\boldsymbol{X}_4$ an new contact is added under active force control instead of velocity control.

### *6.3 Invariance of the Approach*

For a compliant motion task, there is not always an obvious or best choice for the mathematical formalism to represent the problem: for the location of the reference frames, for the physical units to express the magitudes and for the scale in which the models are represented. This implies that a method to solve a problem must be insensitive to changes in this kind of arbitrary choices that have to be made.

#### *6.3.1 Specification in Terms of Physical Properties*

To ensure invariance with respec to to the above mentioned arbitrary choices, a good strategy is to use magnitudes with a physical meaning. This avoids the use of magnitudes that are only mathematical constructs. Therefore the presented method is based on physical properties of rigid bodies such as magnitudes defined by a kinetic and potential energy, and norms defined by a inertia and a compliance [1, 8, 15, 29]. Also, the reciprocity condition between twist and wrench space, is based on the produced work in the interaction between twists of $\mathcal{T}$ and wrenches of $\mathcal{W}$. Since the properties of a physical system do not vary under changes of mathematical representation, reference frame, physical units or scale the method for compliant motion generation is intrinsically invariant with respect to these changes.

### *6.3.2 Invariant Twist and Wrench Projection*

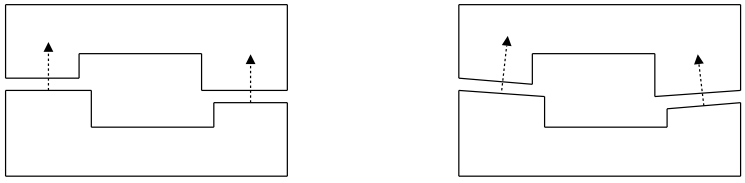
The representation of the twist and wrench subspaces is not invariant with respect to changes of reference frame, physical units or scale. Depending on these changes, the numerical $SVD$ algorithm will produce different bases which however span the same subspace (see equation (26)). All operations applied to twists and wrenches in this paper are invariant with respect to the specific representation of the subspaces. For the projection of twists and wrenches into the twist and wrench space, we use a *weighted* pseudo inverse which minimizes the kinetic or potential energy in the projection error, as shown in equations (35) and (37). These are physical properties and hence invariant. Also, the pose and wrench feedback in the twist and wrench spaces are invariant with respect to the representation of the subspaces, since the same feedback constant is used along each of the basis vectors of the subspaces, as shown in equations (32) and (36).

### *6.3.3 Numerical Issues*

It is not easy to implement linear algebra operations on a computer using a floating point representation, due to the fact that small round off errors on the representation of a subspace of a vector space can alterate its dimension. To cope with these difficulties an approach based on studing vector subspaces using the $SVD$ decomposition has been followed. Starting from a set of vectors $\boldsymbol{W}_{CF}$ that span the wrench space of two rigid bodies, the $SVD$ in (26) results in a matrix $\boldsymbol{U}$ in (27) which contains a basis for both the wrench and the twist space. A threshold $\epsilon$ for the singular values establishes linear dependences between basis vectors and allows us to understand if a column of $\boldsymbol{U}$ belongs to the kernel or to the span of $\boldsymbol{W}_{CF}$, that is, if it belongs to the wrench space or to the twist space of the rigid bodies. In the planar example in Fig. 8, the wrench space of a) is 2-dimensional while the wrench space of b) is 3-dimensional, because the contacting faces are parallel in a) but not in b). The threshold $\epsilon$ defines when the contacting faces are considered parallel or not, and hence when the transition between a 2 and a 3-dimensional wrench space occurs. It is important to point out that this threshold depends on the units and reference frame of $\boldsymbol{W}_{CF}$.

## *6.4 Control Issues*

The presented approach to convert discrete planner setpoints into a path for a robot manipulator, results in discrete changes of the desired robot joint velocity. The discrete changes are caused by (i) the linear interpolation between discrete setpoints within a CF, and (ii) the discrete change in the dimension



a b

Fig. 8. The threshold $\epsilon$ for the singular values defines if a motion degree of freedom belongs to the twist or the wrench space. In the planar example, a) has a 2-dimensional wrench space, while b) has a 3-dimensional wrench space.

of the twist and wrench space at a CF transition. Each of these causes of discontinuities requires a specific solution.

For a compliant motion within a given CF, the dimension of the twist space remains constant. The planner setpoints specify a discrete path within the twist space of the CF. Using a spline function or a smoother, a smooth continuous path can be obtained in the twist space, going through the discrete planned setpoints.

For a compliant motion between two neighboring CFs, a discrete changes in the dimension of the twist and wrench space occurs at the time of the CF transition. The applied control strategy, which is based on the twist and wrench spaces, therefore also changes in a discrete way at a CF transition. The merge approach presented in [12] allows for a smooth transition between different control strategies, by smoothly changing the weight of each of the control strategies in an optimization process.

### *6.5 Choice of Parameters*

The compliant motion generated by our approach depends on a number of arbitrary parameters.

#### *6.5.1 Planner Parameters*

For the off-line planning of the compliant path, we define a step size, using a translational and rotational component $\Delta trans$ and $\Delta rot$. A smaller step size will result in a smaller translational and rotational distance between subsequent poses $\boldsymbol{X}_i$ and $\boldsymbol{X}_{i+1}$ of the planner output. The effect of this step size is discussed in Section 6.2.

### *6.5.2 Compliant Task Generator Parameters*

For the compliant task generation, we define a desired magnitude for the twist and wrench, and two invariant norms. As explained before, the inertia matrix $\boldsymbol{M}$ and the compliance matrix $\boldsymbol{C}$ can correspond to the physical properties of the experimental setup, can be arbitrary, or can be calculated indirectly from specified magnitudes for the force, torque, translational velocity and rotational velocity components of the wrench and twist.

The compliant task generator is also based on arbitrary choices that are not "tunable" by the user. The real magnitude of the desired wrench depends on the specified magnitude for the wrench, and is also chosen proportional to the dimension of the wrench space. The direction of the desired wrench is based on the ECs of the contact formation.

### *6.5.3 Hybrid Controller Parameters*

The hybrid controller needs a proportional feedback gain $K_w^{FB}$ for the wrench feedback control loop and a proportional feedback gain $K_X^{FB}$ for the pose feedback contro llop. The gain $K_w^{FB}$ is limited by the stability of the system, and is chosen as high as possible, without making the system unstable. The gain $K_X^{FB}$ is used to eliminate small integration errors on the desired twist in the twist space. This feedback constant can be very low.

When the weighting matrices $\boldsymbol{M}_c$ and $\boldsymbol{C}_c$ are chosen equal:

$$\boldsymbol{M}_c = \boldsymbol{C}_c, \tag{42}$$

and the weighting matrix $\boldsymbol{C}_c$ represents the real dominant passive compliance between the manipulator and its environment, it can be shown [10] that the closed loops are stable and the errors go to zero, and that the twist and wrench controlled subspaces contain $(6-h)$ and $h$ completely decoupled controllers.

## 7 Experimental Results

This paragraph reports on the real world experiment to validate our approach. It first describes the experimental setup. Then the obtained results are analyzed and discussed.

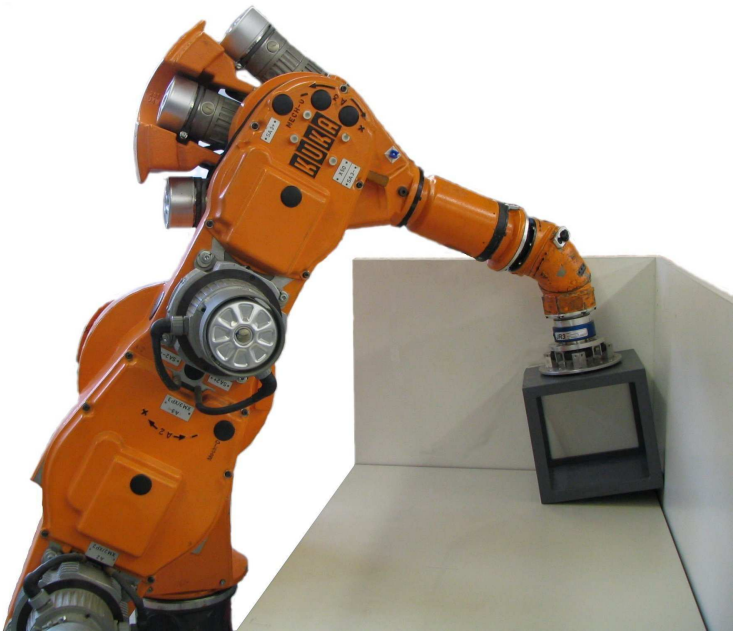


Fig. 9. Experimental setup: the Kuka 361 six degree of freedom industrial robot, manipulating a cube in contact with a corner.

### *7.1 Experimental Setup*

The real world experiment involves the *Kuka 361*, a six degrees of freedom velocity controlled industrial manipulator, shown in Fig. 9. The twist $\boldsymbol{t}_c$ sent to the manipulator is converted into joint velocities $\dot{\boldsymbol{q}}$ using the manipulator Jacobian [5]. These joint velocities are controlled in an analog hardware PID-feedback loop. We can assume that the real joint velocities correspond to the joint velocities sent to the manipulator. A six component *JR3* wrench sensor measures the contact wrench occurring between the manipulated object and the environment. The original controller of the manipulator had very limited capabilities and is bypassed to a desktop computer ($P4$ 2.8 $[Ghz]$) equipped with data acquisition cards. This control computer directly controls the manipulator and reads its sensors. The software platform on the control computer is based on the hard real-time *Open RObot COntrol Software* (Orocos) framework [5] and the *Real-Time Application Interface* (RTAI) extension to the Linux kernel. The low level controller and hardware communication is implemented as a hard realtime, non-interruptible task running at 100 $[Hz]$ with a maximum latency of 16 $[\mu s]$, while the *Compliant Task Generator* generates a task specification for the hybrid controller during the execution, and runs as a non-realtime, interruptible task.

Fig. 9 shows the manipulated object, a cube with an edge length of 25.0 $[cm]$,

[5] The problem of inverting the manipulator Jacobian when it is singular or near singular falls outside the scope of this paper.

Table 1
The sequence of CFs between the cube and its environment. For each CF the dimension of the twist and wrench space is given.

| CF | Principal Contacts | $\dim(\mathcal{W})$ | $\dim(\mathcal{T})$ |
|---|---|---|---|
| $CF_1$ | $3 \times face - face$ | 6 | 0 |
| $CF_2$ | $2 \times face - face$ | 5 | 1 |
| $CF_3$ | $2 \times edge - face$ | 4 | 2 |
| $CF_4$ | $1 \times edge - face$ | 2 | 4 |
| $CF_5$ | $1 \times face - face$ | 3 | 3 |
| $CF_6$ | $1 \times edge - face$ | 2 | 4 |
| $CF_7$ | $2 \times edge - face$ | 4 | 2 |
| $CF_8$ | $1 \times edge - face$ | 2 | 4 |
| $CF_9$ | $1 \times face - face$ | 3 | 3 |

attached to the manipulator with a flexible mounting part. The cube is moved in contact with the environment, which consists of three perpendicular faces of a corner.

### *7.2 Cube-Corner Experiment*

In the first step, a complete contact state graph is generated automatically, given the geometric models of the cube and the corner. Then, the user specifies the input for the compliant planner: the start and goal CF, and intermediate CFs that should be included in the compliant path. The CFs provided to the planner are not chosen to assemble the cube into the corner, but to include many different CFs to verify the effectiveness of our approach. The compliant planner uses the contact state graph and the provided CFs to automatically generate a compliant path that connects all CFs. The translational and rotational step size of the planner are chosen as:

$$\Delta trans = 3.0\ [cm], \tag{43}$$

$$\Delta rot = 5.0 \cdot \frac{\pi}{180}\ [rad]. \tag{44}$$

The sequence of CFs generated by the planner, together with the dimensions of the twist and wrench spaces are shown in Table 1.

While the user only specifies $CF_1$, $CF_4$, $CF_7$ and $CF_9$, the compliant planner generates all necessary intermediate CFs. The compliant path includes a sequence of CFs with relaxations of contact constraints, motions within a CFs, and creations of new contact constraints. As shown in Fig. 10, the sequence

starts at a locally most constrained $CF_1$ when the cube is in contact with all three faces of the corner. The sequence then continues clockwise as indicated by the numbering of the CFs.

The *Compliant Task Generator* automatically converts this compliant path, together with the desired magnitudes for the twist and wrench components, into a task specification for the hybrid controller. For this experiment, the desired magnitudes for the twist and wrench components, at a reference point at the center of the cube, are chosen as:

$$\bar{f} = 25\ [N] \tag{45}$$
$$\bar{\tau} = 3.5\ [Nm] \tag{46}$$
$$\bar{v} = 0.018\ [m/s] \tag{47}$$
$$\bar{\omega} = 0.025\ [rad/s] \tag{48}$$

From this specification, the norm for the twist vector

$$\boldsymbol{M} = 1\ [kg] \cdot \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & 0.72^2 \cdot [m^2] \cdot \boldsymbol{I}_{3\times3} \end{bmatrix}, \tag{49}$$

and the norm for the wrench vector

$$\boldsymbol{C} = 1\ [m/N] \cdot \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & 1/7.07^2 \cdot [1/m^2] \cdot \boldsymbol{I}_{3\times3} \end{bmatrix}, \tag{50}$$

are derived, both expressed in a reference frame at the center of the cube. The twist and wrench magnitudes for these norms, derived from the specified magnitudes, are:

$$E_{kin} = 0.324 \cdot 10^{-3}[J], \tag{51}$$
$$E'_{pot} = 625 \cdot [J]. \tag{52}$$

The compliant task generator generates the task specification for the hybrid controller online, this is during the execution. At each time step, when the hybrid controller requires a new setpoint, it is automatically generated. In this experiment, the hybrid controller uses pose and wrench feedback constants chosen as:

$$K_X^{FB} = 0.05\ [\frac{1}{s}], \tag{53}$$
$$K_w^{FB} = 3.0\ [\frac{1}{s}]. \tag{54}$$

The weighting matrices $\boldsymbol{M}_c$ and $\boldsymbol{C}_c$ are chosen as:

$$\boldsymbol{M}_c = \begin{bmatrix} 3.6\ [kg] \cdot \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & 1.9\ [kg \cdot m^2] \cdot \boldsymbol{I}_{3\times3} \end{bmatrix}, \tag{55}$$

and

$$\boldsymbol{C}_c = \begin{bmatrix} 50 \cdot 10^3\ [\frac{N}{m}] \cdot \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & 1 \cdot 10^3\ [\frac{N \cdot m}{rad}] \cdot \boldsymbol{I}_{3\times3} \end{bmatrix}. \tag{56}$$

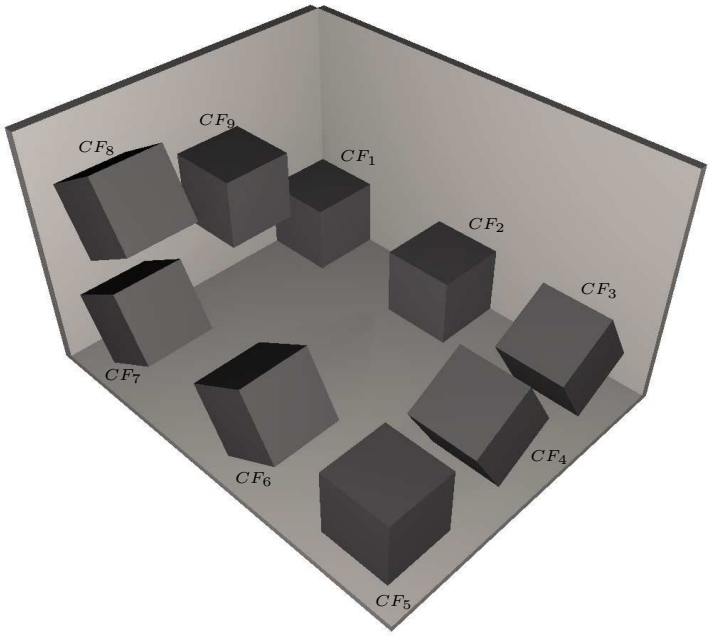



Fig. 10. The experiment executes a sequence of contact formations including different contact relaxations and the creation of new contacts under force control. The sequence starts at a locally most constrained pose $CF_1$, and continues clockwise through the CFs listed in Table 1. The figure only shows the CF sequence, not all CF-compliant poses.

### *7.3 Experimental Results*

The result of the planning, task generation and hybrid controller is a compliant motion of the cube in contact with the corner, through the planned sequence of CFs. During the experiment, wrench and twist measurements are recorded. Fig. 12 (left) shows the force and torque components of the measured contact wrench between the manipulated object and the environment, projected in the wrench space. The measured wrench is projected in the wrench space because the wrench controller only applies to the part of the wrench in the wrench space (see (36) where $\boldsymbol{W}^{\dagger_{\boldsymbol{C}_c}}$ is the projection into the wrench space), whereas

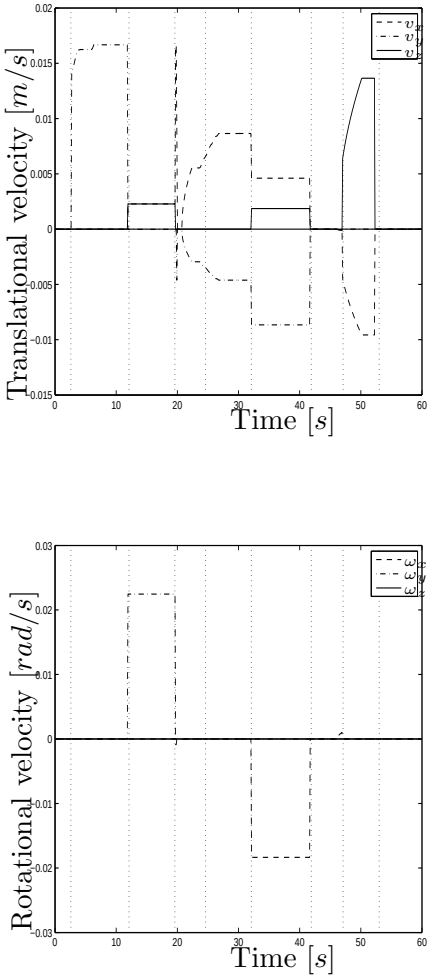



Fig. 11. Desired translational and rotational velocity of a reference point at the center of the cube, expressed in the corner frame.

the measured wrench has components in both twist and wrench space due to friction and inertia forces. The projection in the wrench space filters out the components of the measured wrench in the twist space and therefore reduces —but not completely removes— the effect of inertia and friction forces. The experimental results show that the projection of the measured wrench in the wrench space is sufficient to obtain the desired sequence of contact formations.

Fig. 12 (right) shows the force and torque components of the desired wrench. By construction the desired wrench lies within the local wrench space. The desired wrench is chosen to *achieve a motion* that adds or removes a contact constraint, or maintains a CF. The measured projected wrench is not always equal to the desired wrench. This difference is explained by the following

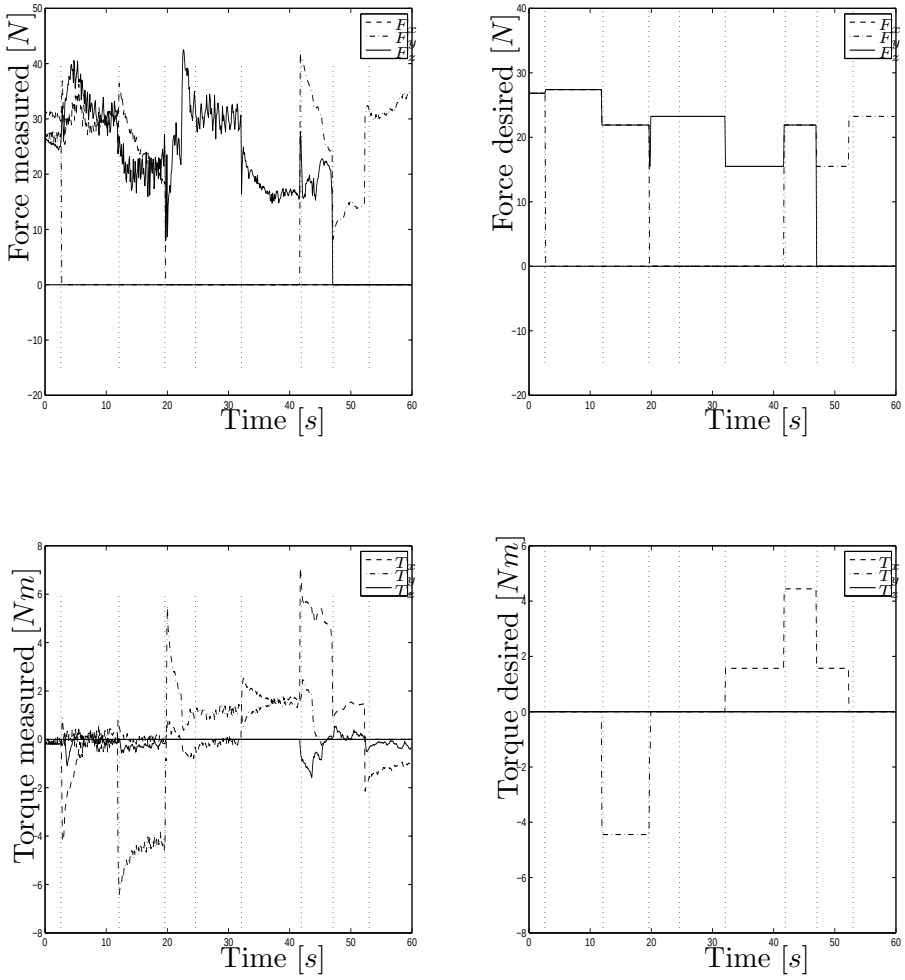



Fig. 12. The measured (left) and desired (right) force and torque reduced to a reference point at the center of the cube, expressed in the corner frame, and projected into the wrench space.

effects:

- The wrench space of a compliant motion between two discrete setpoints of the planner is approximated by the wrench space of the second setpoint. Because the approximated wrench space slightly differs from the real wrench space, it is not always possible to apply the desired contact force withing the approximated wrench space.
- When adding a new contact constraint between $CF_1$ and $CF_2$ (for example moving from an edge-face $CF_1$ to a face-face $CF_2$) the desired wrench is changed to the desired wrench in $CF_2$. However, during the whole motion from $CF_1$ to $CF_2$ the applied wrench cannot yet equal the desired wrench

because the new contact constraint is needed to apply the desired wrench.
- When removing a contact constraint between $CF_1$ and $CF_2$ (for example moving from an face-face $CF_1$ to a edge-face $CF_2$) the desired wrench is changed to the desired wrench in $CF_2$. However, as long as the compliance between the manipulated object and the manipulator is compressed, an extra wrench component is measured.

Fig. 11 shows the translational and rotational components of the desired manipulator twist, projected into the twist space. The velocity controlled manipulator uses a hardware PID velocity feedback loop, and is capable of applying this desired twist in the twist space; hence the measured and desired twist are equal and only the components of the desired twist are shown. Vertical dotted lines divide the twist and wrench plots into 9 sections, one per contact formation that occurs in the experiment. Each vertical line shows where a change of CF occurs. All measurements are expressed in a reference frame attached to the corner, as shown in Fig. 7, and reduced to a reference point at the center of the manipulated cube.

## 8 Conclusions

### *8.1 Compliant Task Generator*

This paper presents the *Compliant Task Generator*: an approach to automatically link the planning and controller efforts in active compliant motion. A compliant path planner provides a geometrical path in the form of a set of six-dimensional poses $\boldsymbol{X}_1 \dots \boldsymbol{X}_n$ and their corresponding contact formations $CF_1 \dots CF_m$. The hybrid compliant controller expects a desired twist $\boldsymbol{t}_d$, pose $\boldsymbol{X}_d$ and wrench $\boldsymbol{w}_d$ at each time step, together with their twist and wrench spaces $\mathcal{T}$ and $\mathcal{W}$. The conversion of the discrete planner primitives into a continuous path represented by the controller primitives, is processed separately for the twist and wrench space. The conversion within the twist space uses a desired magnitude and a norm for the twist, while the conversion within the wrench space uses a desired magnitude and a norm for the wrench. These magnitudes and norms can be directly specified by the user, or indirectly through the desired magnitudes of the twist and wrench components.

The *Compliant Task Generator* is, to the authors' best knowledge, the first general and automated approach that links planning and controller efforts in active compliant motion. It applies to any compliant motion between polyhedral objects with a known geometry, and is more general and simple than previously presented ad-hoc [4] or rule-based methods. Moreover, the method is invariant with respect to changes of reference frame, scale and physical units.

A task-specific input of two magnitudes and norms (or four magnitudes for the twist and wrench components) is sufficient to specify the desired dynamic interaction between the manipulated object and its environment, and allows the fully automated conversion of the planner primitives into the controller primitives. The result is the immediate execution of an off-line planned compliant path by a manipulator, under active force control. In the real world experiment, the approach proved both efficient and effective for all provided compliant paths, including complex contact formations and contact formation transitions.

## *8.2 Related and Future Work*

### *8.2.1 Programming by Human Demonstration*

Another focus of our future research is programming by human demonstration in compliant motion. The authors use the interface to the *Compliant Task Generator* for an off-line planner as well as for programming by human demonstration [32]. In programming by human demonstration, force and pose data is recorded during a demonstration of the desired task. We use estimators, based on the Bayesian approach, to simultaneously extract information about the current contact formation, contact formation changes, and the geometry of the manipulated object and its environment. The free software Bayesian Filtering Library offers a unifying framework for all Bayesian filters, that serves our needs for the integration of online estimators such as Kalman filters, Extended Kalman filters and particle filters.

### *8.2.2 Feedback to Planner*

While the wrench controller works in closed loop, the compliant planner still works in open loop. To close the loop to the compliant planner, we plan to use online estimators to recognize CFs and CF transitions during the execution phase and provide this information to the planner. In previous work by the research group of the authors, CFs where recognized based on pose, wrench and twist data [11, 17, 23, 25, 32], with ever more advanced filters, allowing more uncertainty on the geometry of the involved objects, and taking into account more possible CFs. The problem to execute motions that are more informative about the geometric parameters, allowing better estimation of the geometric parameters, was investigated in [26].

### *8.2.3 Arbitrary Choices*

The presented approach is based on some arbitrary choices, eg. building the desired wrench by positioning a wrench vector at all ECs of the contact formation, the relation between the desired wrench and the contact strength, and the choice of a hybrid controller. We plan to focus on making these choices task-specific instead of arbitrary.

## Acknowledgment

All authors gratefully acknowledge the financial support by K.U.Leuven's Concerted Research Action GOA/05/10. Part of the work was performed by the first author at the University of North Carolina at Charlotte, supported by the U.S. National Science Foundation under grant IIS-0328782.

## References

[1] Wojciech Blajer. A geometric unification of constrained system dynamics. *Multibody System Dynamics*, 1(1):3–21, 1997.

[2] Herman Bruyninckx. *Kinematic Models for Robot Compliant Motion with Identification of Uncertainties*. PhD thesis, Dept. Mech. Eng., Katholieke Univ. Leuven, Belgium, 1995.

[3] Herman Bruyninckx and Joris De Schutter. Specification of force-controlled actions in the "Task Frame Formalism": A survey. *IEEE Trans. Rob. Automation*, 12(5):581–589, 1996.

[4] Herman Bruyninckx and Joris De Schutter. Where does the Task Frame go? In Yoshiaki Shirai and Shigeo Hiroshe, editors, *Robotics Research, the 8th Intern. Symp.*, pages 55–65, Shonan, Japan, 1997. Springer-Verlag.

[5] Herman Bruyninckx, Peter Soetens, and Bob Koninckx. The real-time motion control core of the Orocos project. In *Int. Conf. Robotics and Automation*, pages 2766–2771, Taipeh, Taiwan, 2003.

[6] J. Chen and Alexander Zelinsky. Programing by demonstration: Coping with suboptimal teaching actions. *Int. J. Robotics Research*, 22(5):299–319, 2003.

[7] Stefano Chiaverini and Lorenzo Sciavicco. The parallel approach to force/position control manipulators. *IEEE Trans. Rob. Automation*, 9:289–293, 1993.

[8] Alessandro De Luca and Costanzo Manes. Modeling of robots in contact with a dynamic environment. *IEEE Trans. Rob. Automation*, 10(4):542–548, 1994.

[9] Joris De Schutter and Herman Bruyninckx. Force control of robot manipulators. In *The Control Handbook*, pages 1351–1358. CRC Press, 1996.

[10] Joris De Schutter and Herman Bruyninckx. Force control of robot manipulators. In William S. Levine, editor, *Control System Applications*, pages 177–184. CRC Press, 2000. Reprint of [9].

[11] Joris De Schutter, Herman Bruyninckx, Stefan Dutré, Jan De Geeter, Jayantha Katupitiya, Sabine Demey, and Tine Lefebvre. Estimating first order geometric parameters and monitoring contact transitions during force controlled compliant motion. *Int. J. Robotics Research*, 18(12):1161–1184, 1999.

[12] Joris De Schutter, Tinne De Laet, Johan Rutgeerts, Wilm Decré, Ruben Smits, Erwin Aertbeliën, Kapser Claes, and Herman Bruyninckx. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. 2006. conditionnaly accepted.

[13] Joris De Schutter, Johan Rutgeerts, Erwin Aertbelien, Friedl De Groote, Tinne De Laet, Tine Lefebvre, Walter Verdonck, and Herman Bruyninckx. Unified constraint-based task specification for complex sensor-based robot systems. In *Int. Conf. Robotics and Automation*, pages 3618–3623, Barcelona, Spain, 2005.

[14] Joris De Schutter and H. Van Brussel. Compliant Motion I, II. *Int. J. Robotics Research*, 7(4):3–33, Aug 1988.

[15] K. L. Doty, C. Melchiorri, and C. Bonivento. A theory of generalized inverses applied to robotics. *Int. J. Robotics Research*, 12(1):1–19, 1993.

[16] Joseph Duffy. The fallacy of modern hybrid control theory that is based on "orthogonal complements" of twist and wrench spaces. *J. Robotic Systems*, 7(2):139–144, 1990.

[17] Klaas Gadeyne, Tine Lefebvre, and Herman Bruyninckx. Bayesian hybrid model-state estimation applied to simultaneous contact formation recognition and geometrical parameter estimation. *Int. J. Robotics Research*, 24(8):615–630, 2005.

[18] Neville Hogan. Stable execution of contact tasks using impedance control. In *Int. Conf. Robotics and Automation*, pages 1047–1054, Raleigh, NC, 1987.

[19] Xuerong Ji and Jing Xiao. Planning motions compliant to complex contact states. *Int. J. Robotics Research*, 20(6):446–465, July 2001.

[20] L.E. Kavraki and J.C. Latombe. Probabilistic roadmaps for robot path planning. In Kamal Gupta and Angel P. del Pobil, editors, *Practical Motion Planning in Robotics*, pages 33–53. Wiley, 1998.

[21] Torsten Kröger, Bernd Finkemeyer, Markus Heuck, and Friedrich M. Wahl. Compliant motion programming: The task frame formalism revisited. *J. of Rob. and Mech.*, 3:1029–1034, 2004.

[22] Jean Claude Latombe. *Robot motion planning*, volume 124 of *Int. Series in Engineering and Computer Science*. Kluwer, Boston, MA, 1991.

[23] Tine Lefebvre, Herman Bruyninckx, and Joris De Schutter. Polyhedral contact formation modeling and identification for autonomous compliant motion. *IEEE Trans. Rob. Automation*, 19(1):26–41, 2003.

[24] Tine Lefebvre, Herman Bruyninckx, and Joris De Schutter. *Nonlinear Kalman Filtering for Force-Controlled Robot Tasks*. Springer Tracts in Advanced Robotics. Springer Verlag, 2005.

[25] Tine Lefebvre, Herman Bruyninckx, and Joris De Schutter. Polyhedral contact formation identification for autonomous compliant motion: Exact nonlinear Bayesian filtering. *IEEE Trans. Rob. Automation*, 21(1):124–129, 2005.

[26] Tine Lefebvre, Herman Bruyninckx, and Joris De Schutter. Task planning with active sensing for autonomous compliant motion. *Int. J. Robotics Research*, 24(1):61–82, 2005.

[27] H. Lipkin and J. Duffy. Hybrid twist and wrench control for a robotic manipulator. *Trans. ASME J. Mech. Transm. Automation Design*, 110:138–144, 1988.

[28] Josip Lončarić. Normal forms of stiffness and compliance matrices. *IEEE J. Rob. Automation*, RA-3(6):567–572, 1987.

[29] C. Manes. Recovering model consistence for force and velocity measures in robot hybrid control. In *Int. Conf. Robotics and Automation*, pages 1276–1281, Nice, France, 1992.

[30] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-11(6):418–432, 1981.

[31] Wim Meeussen. *Compliant robot motion: from path planning or human demonstration to force controlled task execution*. PhD thesis, Dept. Mech. Eng., Katholieke Univ. Leuven, Belgium, December 2006.

[32] Wim Meeussen, Johan Rutgeerts, Klaas Gadeyne, Herman Bruyninckx, and Joris De Schutter. Contact state segmentation using particle filters for programming by human demonstration in compliant motion tasks. *IEEE Trans. Rob*, 2006. in press.

[33] R. M Murray, Z. Li, and S. S. Sastry. *A mathematical introduction to robotic manipulation*. CRC Press, Boca Raton, FL, 1994.

[34] F. C. Park. Distance metrics on the rigid-body motions with applications to mechanism design. *Trans. ASME J. Mech. Design*, 117:48–54, 1995.

[35] M. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *Trans. ASME J. Dyn. Systems Meas. Control*, 102:126–133, 1981.

[36] Qi Wang, Joris De Schutter, Wim Witvrouw, and Sean Graves. Derivation of compliant motion programs based on human demonstration. In *Int. Conf. Robotics and Automation*, pages 2616–2621, Minneapolis, MN, 1996.

[37] Jing Xiao. Automatic determination of topological contacts in the presence of sensing uncertainty. In *Int. Conf. Robotics and Automation*, pages 65–70, Atlanta, GA, 1993.

[38] Jing Xiao and Xuerong Ji. A divide-and-merge approach to automatic generation of contact states and planning of contact motions. In *Int. Conf. Robotics and Automation*, pages 750–756, San Francisco, CA, 2000.

[39] Jing Xiao and Xuerong Ji. On automatic generation of high-level contact state space. *Int. J. Robotics Research*, 20(7):584–606, 2001.