Integration of Planning and Execution in Force Controlled Compliant Motion

Wim Meeussen*†, Joris De Schutter*, Herman Bruyninckx*, Jing Xiao† and Ernesto Staffetti†
* Department of Mechanical Engineering
Katholieke Universiteit Leuven, Belgium
† Computer Science Department
University of North Carolina, Charlotte, USA

***Abstract*— This paper presents the *Compliant Task Generator*: a new approach for the automatic conversion of a geometric path generated by a compliant path planner to a force based task specification for a compliant robot controller. Based on the geometric model of a moving object and its environment, a compliant path planner generates a set of six-dimensional positions $x_{1...m}$ and their corresponding contact formations $CF_{1...n}$. The compliant force controller, which executes a planned path under force feedback using the *Hybrid Control Paradigm*, expects a desired force w_d, velocity t_d and position x_d at each time-step, together with their force and velocity controlled subspaces W and T. To specify these controller primitives, we add information about the desired dynamic interaction between the moving object and its environment, in the form of the desired kinetic energy E_{kin} of the moving object and the potential energy E_{pot} in the contacts with the environment, together with the inertia and stiffness matrix M and S. We fully automated the conversion process of the compliant planner output together with the added information about the dynamic interaction, to a force based task specification. This eliminates the requirement of human intervention between the planning and execution phase. The presented approach applies to all compliant motions between polyhedral objects, and is verified in a real world experiment.**

I. Introduction

Compliant motion [7] guides a manipulator or an object held by a manipulator, using the force interaction with constraining environmental objects, to help overcome uncertainties associated with assembly and manipulation tasks. M. T. Mason was one of the first to publish works on compliance and force control [13]. He introduced the *Task Frame Formalism* (TFF), an intuitive and manipulator independent tool that presents an interface to an operator for the specification of force controlled robot tasks in the *Hybrid Control Paradigm* (HCP) [14]. Bruyninckx and De Schutter made an extensive catalog of TFF models and specifications in [2]. While the TFF has potential application fields in many contact tasks, it cannot model more complex contact situations with multiple contacts, even with polyhedral objects. Recently, De Schutter presented a constraint-based task specification formalism that can cope with complex contact formations [6].

However, these approaches themselves have no "intelligence", but rely on human intervention and intuition to define a task specification compatible with the framework. For more complex tasks, involving multiple contacts and complex changes in contact formations, this can prove almost impossible. Even between two simple polyhedral objects, hundreds of contact formations and transitions between neighboring contact formations are possible. Different approaches try to automatically generate the complex sequence of contact formations in a task. The *Programming by Human Demonstration* approach [5], [16] extracts force, position and contact data about a task, while a human demonstrates the task, using a robot manipulator or a special demonstration tool. Another approach, as used in our research, uses a planning system that automatically generates a compliant path based on a geometrical model of the manipulated object and its environment. Xiao and Ji developed an off-line compliant planner [10], which is based on their systematic approach to automatically generate the contact state space between two arbitrary polyhedral objects in terms of a contact state graph [18]. Based on a geometric model of the two contacting polyhedral objects, this planner generates a complex compliant path by compliant interpolation between neighboring contact states. The resulting compliant path however, only contains geometric and topological information, and cannot be used directly by a controller using the HCP.

In this paper we present an approach, verified by real world experimental results, to automatically generate a task specification for a hybrid controller, based on the output of the compliant planner and the desired dynamic interaction between the objects in contact. This allows us to generate a complex compliant path using the geometric models of the objects in contact, and immediately execute it on a real robot manipulator using force feedback, as schematically represented in figure 3.

The rest of the paper is organized as follows. In section II we briefly review the concept of contact formation and contact state graph. The next section discusses the interfaces to the compliant planner and the hybrid controller, defined by the planner and controller primitives. Section IV describes the automatic generation of the controller primitives, using the planner primitives and information about the desired dynamics and kinematics of the system. Section V explains the internals of the hybrid force-velocity controller, that controls the manipulator according to task specified by the controller primitives. The experimental setup we used to verify our approach and the obtained results are discussed in section VI. Finally, section VII contains the conclusions and possible future extensions.

II. Review: Contact Formations and Contact State Graph

A. Contact Formations

The notion of *principal contacts* (PC's) was introduced [17] 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. As an example, figure 1 shows the six non-degenerate PC's that can be formed between two polyhedral objects. Each non-degenerate PC is associated with a *contact plane*, defined by a contacting face or the two contacting edges at an edge-edge PC. A general contact state between two objects can be characterized topologically by the set of PC's formed, called a *contact formation* (CF). A configuration compliant to a CF is called a *CF-compliant configuration*. Any motion formed by a sequence of CF-compliant configurations is called a *CF-compliant motion*.

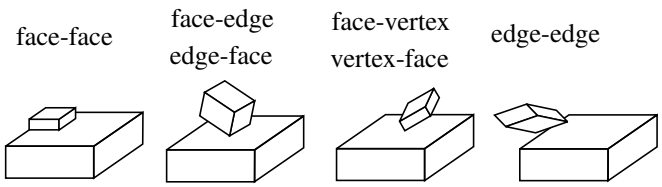

Fig. 1. Every non-degenerate contact formation between polyhedral objects can be described by a combination of the six non-degenerate Principal Contacts (PC's)

B. Contact State Graph

Xiao and Ji developed a divide-and-merge approach [18] to generate the contact state space between two polyhedral objects as a contact state graph G. In G, a node represents a contact formation, and an arc represents the adjacency relationship between two contact formations, as shown in figure 2. Specifically, the approach takes advantage of the fact that G can be divided into special subgraphs called the *goal-contact relaxation* (GCR) graphs, where each GCR graph is defined by a locally most constrained contact state, called the seed, and its less-constrained neighboring contact states. The approach was implemented with algorithms to generate a complete GCR graph automatically and to merge multiple GCR graphs automatically into a single contact state graph [18].

III. Compliant Planner and Hybrid Controller Primitives

A. Compliant Planner Primitives

The problem of compliant motion planning can be defined as follows: given a CF_1-compliant initial configuration $\boldsymbol{x}_1$, search a collision-free compliant path to a CF_n-compliant

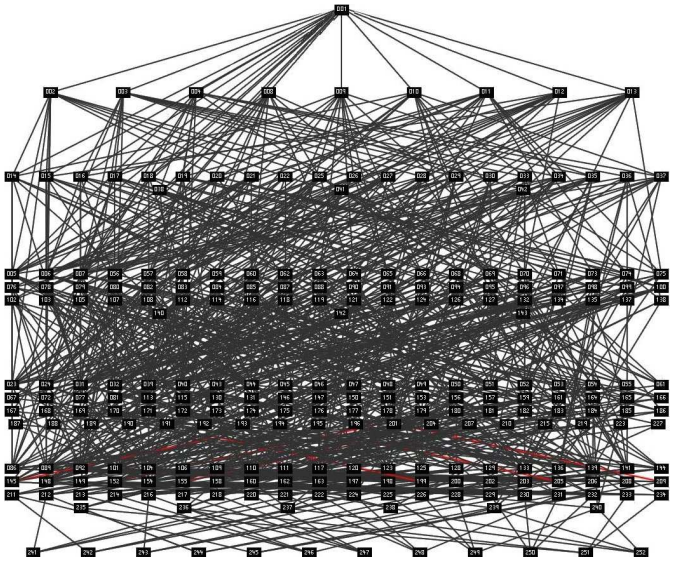

Fig. 2. The contact state graph of two polyhedral objects shows all possible contact formations (nodes) and transitions between neighboring contact formations (arcs).

end configuration $\boldsymbol{x}_m$ through a sequence of adjacent contact formations between CF_1 and CF_n. Ji and Xiao developed a two-level geometric approach to tackle this problem [10]. First a high-level graph search in the contact state graph results in a sequence of state transitions between adjacent contact formations, connecting CF_1 and CF_n. Then a low-level motion planner within each contact formation CF_j of the high level path produces a sequence of CF_j-compliant configurations, connecting the last configuration of CF_{j-1} to the first configuration of CF_{j+1}. This low-level motion planner [9] is based on extending the *Probabilistic Roadmap Paradigm* [11] by building roadmaps of randomly sampled CF_j-compliant configurations and by compliant interpolations. A roadmap contains feasible CF_j-compliant configurations that are not in other collisions unspecified by CF_j as nodes. Two nodes are connected by an arc if a feasible CF_j-compliant path can be found from straight-line compliant interpolation between the configurations of the two nodes. Next a graph search in such a roadmap can result in a feasible CF_j-compliant path in terms of a sequence of feasible CF_j-compliant configurations. Finally, a concatenation of CF_j-compliant paths for $j = 1...n$ yields a feasible compliant path from $\boldsymbol{x}_1$ of CF_1 to $\boldsymbol{x}_m$ of CF_n.

The output primitives of this compliant planner only contain geometrical and topological information and are represented by a set of configurations $\boldsymbol{x}_{1...m}$ and their corresponding contact formations $CF_{1...n}$. Each two configurations $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$ are at the same or at neighboring contact formations. The motion from the last configuration of CF_j to the first configuration of CF_{j+1} is always small and finite, allowing us to only use the information of CF_{j+1} to guide such a motion, as will be explained in section IV-B.

B. Hybrid Controller Primitives

The *Hybrid Control Paradigm* (HCP) [13], [14] is one of the two major force control paradigms, together with the *Impedance Control*, and assumes a *geometric* interaction model. In HCP terminology, an object has b Cartesian degrees of freedom which are force controlled, and $(6-b)$ Cartesian degrees of freedom which are velocity controlled, represented by a b dimensional force controlled subspace $\boldsymbol{W}$ and a $(6-b)$ dimensional velocity controlled subspace $\boldsymbol{T}$. All vectors of $\boldsymbol{W}$ and $\boldsymbol{T}$ are reciprocal [12], which means that the ideal contact forces, represented in $\boldsymbol{W}$, produce no work against the allowed instantaneous velocities, represented in $\boldsymbol{T}$.

In the specific subset of all possible compliant motion tasks which can be specified using the *Task Frame* approach [13], is always possible to find a (moving) reference frame in which the subspaces $\boldsymbol{W}$ and $\boldsymbol{T}$ are time-invariant within one contact formation. This makes an intuitive force and velocity specification along the axes of the time-invariant subspaces possible: a desired wrench and velocity along the task frame axes completely specifies the task. The Task Frame covers most simple contact formations between two objects, but for example a little more complex contact formation involving two vertex-face contacts at two different faces is not covered any more. Thus, for a general compliant motion task it is not possible to find such a time-invariant reference frame and $\boldsymbol{W}$ and $\boldsymbol{T}$ will depend on the specific configuration within each contact formation. This implies that the primitives of a general hybrid controller at time t not only include the desired wrench $\boldsymbol{w_d}$, the desired position $\boldsymbol{x_d}$ and the desired velocity $\boldsymbol{t_d}$, but also the specific reciprocal force and velocity controlled subspaces $\boldsymbol{W}$ and $\boldsymbol{T}$ at that time. The controller primitives are the task specification for the controller, and specify the same path as the planner primitives, but in a form the controller can understand. Figure 3 gives a schematic overview of the primitives used by the compliant planner and the hybrid controller. The next section explains how the planner primitives are converted into the controller primitives.

IV. Compliant Task Generator

This section describes the core of our approach, the automatic conversion of a geometric path generated by a compliant path planner, to a force based tasks specification for a compliant robot controller. The compliant planner generates a set of positions $\boldsymbol{x}_{1\ldots m}$ and their contact formations $CF_{1\ldots n}$, while the hybrid controller expects a desired force $\boldsymbol{w_d}$, velocity $\boldsymbol{t_d}$ and position $\boldsymbol{x_d}$ at each time-step, together with their force and velocity controlled subspaces $\boldsymbol{W}$ and $\boldsymbol{T}$. To automatically generate these controller primitives, based on the planner primitives, we add information about the desired dynamic interaction between the moving object and its environment, contained in only four parameters: a constant kinetic energy E_{kin} together with the mass matrix $\boldsymbol{M}$ of the moving object, and a constant potential energy E_{pot} at each elementary contact together with the local contact stiffness matrix $\boldsymbol{K}$. The constant kinetic energy is used to convert the discrete planner position setpoints $\boldsymbol{x}_{1\ldots m}$ into a smooth velocity trajectory in the velocity controlled subspace, by giving the moving object a constant kinetic energy. The constant potential energy is used to convert the planner contact formation setpoints $CF_{1\ldots n}$ into a smooth force trajectory, by giving the contacts between the moving object and its environment a constant potential energy. Note that E_{kin} only affects the velocity controlled subspace and E_{pot} only the force controlled subspace, unlike in other approaches such as the *Impedance Control Paradigm* where both E_{kin} and E_{pot} influence both forces and velocities.

A. Desired velocity, position and force

The desired twist $\boldsymbol{t_d}$, position $\boldsymbol{x_d}$ and wrench $\boldsymbol{w_d}$ are calculated using the set of configurations $\boldsymbol{x}_{1\ldots m}$ and their contact formation $CF_{1\ldots m}$, together with the kinetic and potential energy E_{kin} and E_{pot}, and the mass and stiffness matrices $\boldsymbol{M}$ and $\boldsymbol{K}$.

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 constant kinetic energy E_{kin} is defined by

$$\boldsymbol{t_d} = \sqrt{\frac{2 \cdot E_{kin}}{\boldsymbol{t_{i,i+1}^T} \cdot \boldsymbol{M} \cdot \boldsymbol{t_{i,i+1}}}} \cdot \boldsymbol{t_{i,i+1}} \tag{1}$$

in which

$$\boldsymbol{t_{i,i+1}} = \boldsymbol{x_{i+1}} - \boldsymbol{x_i} \tag{2}$$

is a constant displacement twist to move from $\boldsymbol{x_i}$ to $\boldsymbol{x_{i+1}}$. The mass matrix $\boldsymbol{M}$ represents the inertia properties of the moving object.

The desired position $\boldsymbol{x_d}$ at time $t \in [t_i, t_{i+1}[$, between $\boldsymbol{x_i}$ and $\boldsymbol{x_{i+1}}$ is defined by

$$\boldsymbol{x_d} = \boldsymbol{x_i} + \boldsymbol{t_d} \cdot (t - t_i) \tag{3}$$

The desired wrench $\boldsymbol{w_d}$ at time $t \in [t_i, t_{i+1}[$ depends on the *contact strength* of the contact formation of $\boldsymbol{x}_{i+1}$. This means that the desired wrench at a point contact (vertex-face, face-vertex or edge-edge) is smaller than at a line contact (edge-face

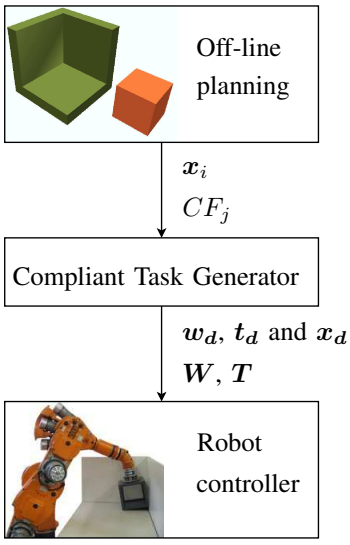

Fig. 3. Compliant task generation: the primitives of the off-line compliant path planner are converted into primitives for the hybrid robot controller.

or face-edge), and that the desired wrench at a line contact is smaller than at a plane contact (face-face). Therefore we we further decompose all PC's of the contact formation of $\boldsymbol{x}_{i+1}$ into *elementary contacts* EC_k, with $k = 1 \dots v$. EC's are associated with a contact point and a contact normal, as shown in figure 4. At each EC_k a frame f_k can be positioned, with the origin at the contact point and the x-axis along the contact normal, oriented from the environment towards the moving object. In frame f_k, a unit contact force is represented by a wrench vector

$$\boldsymbol{w_{unit}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{4}$$

Now we choose the desired wrench $\boldsymbol{w_{d_k}}$ at each EC_k so that the potential energy that is built up at each EC_k equals E_{pot}.

$$\boldsymbol{w_{d_k}} = \sqrt{\frac{2 \cdot E_{pot}}{\boldsymbol{w_{unit}^T} \cdot \boldsymbol{K^{-1}} \cdot \boldsymbol{w_{unit}}}} \cdot \boldsymbol{w_{unit}} \tag{5}$$

in which $\boldsymbol{K}$ is the local contact stiffness matrix at the elementary contact.

The total desired wrench $\boldsymbol{w_d}$ is defined by the sum of all $\boldsymbol{w_{d_{1\dots v}}}$ at each $EC_{1\dots v}$, transformed to a frame f by

$$\boldsymbol{w_d} = \sum_{k=1}^{v} \left(\begin{bmatrix} \boldsymbol{R^f_{f_k}} & \boldsymbol{0}_{3x3} \\ \left[\boldsymbol{p^f_{f_k}}\times\right] \cdot \boldsymbol{R^f_{f_k}} & \boldsymbol{R^f_{f_k}} \end{bmatrix} \cdot \boldsymbol{w_{d_k}} \right) \tag{6}$$

in which $\boldsymbol{R^f_{f_k}}$ is the rotation and $\boldsymbol{p^f_{f_k}}$ the translation from f_k to f, and $\left[\boldsymbol{p^f_{f_k}}\times\right]$ the 3×3 skew-symmetric matrix representing the cross product with $\boldsymbol{p^f_{f_k}}$

$$\left[\boldsymbol{p^f_{f_k}}\times\right] = \begin{bmatrix} 0 & -\boldsymbol{p^f_{f_k z}} & \boldsymbol{p^f_{f_k y}} \\ \boldsymbol{p^f_{f_k z}} & 0 & -\boldsymbol{p^f_{f_k x}} \\ -\boldsymbol{p^f_{f_k y}} & \boldsymbol{p^f_{f_k x}} & 0 \end{bmatrix} \tag{7}$$

B. Defining force and velocity controlled subspaces

To obtain the force controlled subspace $\boldsymbol{W}$ and the reciprocal velocity controlled subspace $\boldsymbol{T}$ at time $t \in [t_i, t_{i+1}]$, we use the topological information of the contact formation at the next time-step t_{i+1}. This is important when there is a change in contact formation between $\boldsymbol{x_i}$ and $\boldsymbol{x_{i+1}}$, which is always a change between neighboring $CF's$, CF_j and CF_{j+1}. In the case that CF_{j+1} is less constrained than CF_j, i.e, a contact is broken from CF_j to CF_{j+1}, the velocity controlled subspace of the latter will be higher dimensional than the velocity controlled subspace of the former, allowing a motion to break the contact. In the case that CF_{j+1} is more constrained than CF_j, the force controlled subspace of the latter will be higher dimensional then the force controlled subspace of the former, which allows the creation of a new contact under force feedback.

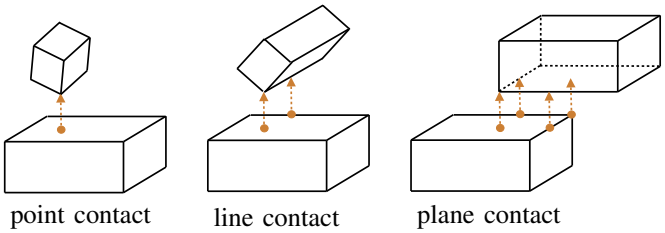

Fig. 4. A general contact formation (CF) can be decomposed into elementary contacts (EC's), which are associated with a contact point and a contact normal

To get a base for the force and velocity controlled subspaces at CF_{j+1}, we combine all wrench vectors $\boldsymbol{w}_{1\dots v}$ at CF_{j+1}, to form a non-minimal base for the $\boldsymbol{W}$. Using the *singular value decomposition* of this base, in the form of $\boldsymbol{A} = \boldsymbol{U} \cdot \boldsymbol{S} \cdot \boldsymbol{V^T}$, we get a minimal base for $\boldsymbol{W}$ and for its dual space $\boldsymbol{T}$,

$$\begin{bmatrix} \boldsymbol{w_1^T} & \dots & \boldsymbol{w_v^T} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W} & \boldsymbol{T} \end{bmatrix} \cdot \begin{bmatrix} s_1 & & 0 \\ & \ddots & \vdots \\ 0 & & s_p \\ & \dots & \end{bmatrix} \cdot \boldsymbol{V^T} \tag{8}$$

in which

$$s_1 \geq \dots \geq s_b > \epsilon > s_{b+1} \geq \dots \geq s_p \geq 0 \tag{9}$$

are the p singular values, with $p \leq 6$. The threshold $\epsilon \approx 0$ selects the singular values that are almost equal to zero, creating a b-dimensional force controlled subspace and a $(6-b)$-dimensional velocity controlled subspace. The reciprocity condition for the two subspaces

$$\boldsymbol{W^T} \cdot \boldsymbol{T} = \boldsymbol{0} \tag{10}$$

means that the wrenches of $\boldsymbol{W}$ produce no work against the twists of $\boldsymbol{T}$.

V. Implementation of the Hybrid Controller

The task specification for the controller side consists of the desired velocity $\boldsymbol{t_d}$, the desired position $\boldsymbol{x_d}$ and the desired force $\boldsymbol{w_d}$ at each time-step, in the reciprocal velocity controlled subspace $\boldsymbol{T}$ and force controlled subspace $\boldsymbol{W}$. The hybrid controller combines this input with the measured force $\boldsymbol{w_m}$ and measured position $\boldsymbol{x_m}$, into velocity setpoints $\boldsymbol{t_s}$ for a velocity controlled manipulator.

A. Position and Velocity Controller in $\boldsymbol{T}$

The desired velocity $\boldsymbol{t_d}$, the desired position $\boldsymbol{x_d}$ and the measured position $\boldsymbol{x_m}$ define the velocity for the manipulated object in the $(6-b)$-dimensional (see (9)) velocity controlled subspace $\boldsymbol{T}$, represented by the $(6-b)$-dimensional vectors u_t and u_x.

$$\boldsymbol{u_t} = \boldsymbol{T}^\dagger \cdot \boldsymbol{t_d} \tag{11}$$

$$\boldsymbol{u_x} = \boldsymbol{K^x_{FB}} \cdot \boldsymbol{T}^\dagger \cdot (\boldsymbol{x_d} - \boldsymbol{x_m}) \tag{12}$$

in which $\boldsymbol{T}^\dagger$ is the pseudo inverse of $\boldsymbol{T}$, using the mass matrix $\boldsymbol{M}$ of the manipulated object as a weight matrix

$$\boldsymbol{T}^\dagger = (\boldsymbol{T^T} \cdot \boldsymbol{M} \cdot \boldsymbol{T})^{-1} \cdot \boldsymbol{M} \tag{13}$$

which minimizes the projection error when an exact projection into the velocity controlled subspace $\boldsymbol{T}$ is not possible due to modeling uncertainties, by minimizing the kinetic energy of the mass $\boldsymbol{M}$. The $(6-b)$-dimensional diagonal matrix $\boldsymbol{K^x_{FB}}$ specifies the proportional position feedback in $\boldsymbol{T}$.

B. Force Controller in $\boldsymbol{W}$

The desired wrench $\boldsymbol{w_d}$ and the measured wrench $\boldsymbol{w_m}$ define a wrench-change for the manipulated object in the b-dimensional (see (9)) force controlled subspace $\boldsymbol{W}$, represented by the b-dimensional vector $\boldsymbol{u_w}$.

$$\boldsymbol{u_w} = \boldsymbol{K^w_{FB}} \cdot \boldsymbol{W}^{\dagger} \cdot (\boldsymbol{w_d} - \boldsymbol{w_m}) \tag{14}$$

in which $\boldsymbol{W}^{\dagger}$ is the pseudo inverse of $\boldsymbol{W}$, using the inverse stiffness matrix $\boldsymbol{K}^{-1}$, where $\boldsymbol{K}$ is the contact stiffness between the manipulator and the manipulated object, as the weight matrix

$$\boldsymbol{W}^{\dagger} = (\boldsymbol{W^T} \cdot \boldsymbol{K}^{-1} \cdot \boldsymbol{W})^{-1} \cdot \boldsymbol{K}^{-1} \tag{15}$$

which minimizes the projection error when an exact projection into the force controlled subspace $\boldsymbol{W}$ is not possible due to modeling uncertainties and friction, by minimizing the potential energy in a spring with stiffness $\boldsymbol{K}$. The b-dimensional diagonal matrix $\boldsymbol{K^w_{FB}}$ specifies the proportional force feedback in $\boldsymbol{W}$.

When changing from a less constrained CF_j to a more constrained CF_{j+1}, a new contact is added under force control because the contact formation at the next time-step us used to build the force controlled subspace. This makes the approach robust to inaccurate modeling of the objects in contact. To verify if and when the new contact has been created, we use a simple online estimator that checks when the potential energy in the difference between the desired and the measured force goes below a threshold δ

$$\frac{1}{2} \cdot \Big[(\boldsymbol{w_d} - \boldsymbol{w_m})^T \cdot \boldsymbol{K}^{-T} \cdot (\boldsymbol{w_d} - \boldsymbol{w_m}) \Big] < \delta \tag{16}$$

C. Resulting Manipulator Velocity

The $(6-m)$-dimensional vectors $\boldsymbol{u_t}$ and $\boldsymbol{u_x}$ define the resulting twist in the velocity controlled twist space, and can be directly applied by a velocity controlled manipulator. The m-dimensional vector $\boldsymbol{u_w}$ however, which defines the resulting wrench at the manipulated object, can only be applied indirectly by a velocity controlled manipulator, by an interaction between the manipulated object and its environment. Therefore, as shown in figure 6, we use the stiffness $\boldsymbol{K_s}$ of a flexible mounting part connecting the manipulated object with the manipulator to define the relation between an applied twist and the resulting wrench change.

$$\boldsymbol{t_s} = \begin{bmatrix} \boldsymbol{K}_s^{-1} \cdot \boldsymbol{W} & \boldsymbol{T} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{u_w} \\ \boldsymbol{u_t} + \boldsymbol{u_x} \end{bmatrix} \tag{17}$$

At the reference frame of the manipulated object, $\boldsymbol{K_s}$ is defined by

$$\boldsymbol{K_s} = \begin{bmatrix} \boldsymbol{R^T} & \boldsymbol{0}_{3x3} \\ -\boldsymbol{R^T} \cdot [\boldsymbol{p}\times] & \boldsymbol{R^T} \end{bmatrix} \cdot \boldsymbol{K'_s} \cdot \begin{bmatrix} \boldsymbol{R} & [\boldsymbol{p}\times] \cdot \boldsymbol{R} \\ \boldsymbol{0}_{3x3} & \boldsymbol{R} \end{bmatrix} \tag{18}$$

in which $\boldsymbol{R}$ is the rotation matrix and $\boldsymbol{p}$ the displacement vector from the flexible mounting part to the manipulated object reference frame. $[\boldsymbol{p}\times]$ is a 3×3 skew-symmetric matrix representing the cross product with $\boldsymbol{p}$ as defined in equation (7). The diagonal matrix $\boldsymbol{K'_s}$ approximates the stiffness expressed at the center of the flexible mounting part.

Finally, the twist $\boldsymbol{t_s}$ is transformed into joint velocities $\dot{\boldsymbol{q}}$ for the velocity controlled manipulator. We use a high speed control loop with Cartesian position feedback on the integrated twist $\boldsymbol{x_f}$ of the manipulated object, to eliminate drift on the manipulator position. A control scheme of this feedback loop is shown in figure 5, in which $\boldsymbol{J}$ is the manipulator Jacobian and $\boldsymbol{K^p_{FB}}$ the Cartesian position feedback constant.

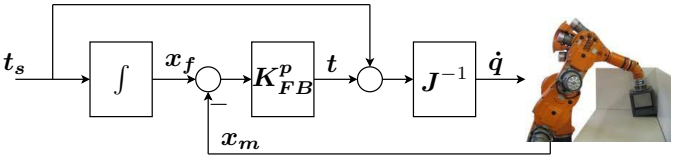

Fig. 5. The high speed Cartesian position feedback loop with velocity feedforward controls the joint velocities of the manipulator.

VI. Experimental Results

This paragraph describes the experimental setup and the obtained results of the real world experiment we used to verify our approach.

A. Experimental Setup

In our experiments we use the *Kuka 361*, a six degrees of freedom velocity controlled industrial manipulator, which is shown in figure 6. The manipulated object, a cube, is attached to the manipulator with a flexible mounting part. The cube is moved in contact with the environment, which consists of three perpendicular faces forming a corner. We use a six dimensional *JR3* wrench sensor to measure the wrench applied by the cube on the corner.

The hybrid controller and hardware communication is implemented in the hard realtime *Open Robot Control Software* (Orocos) framework [1]. The hybrid controller is implemented as a hard realtime, not interruptible task running at $100[Hz]$ with a maximum latency of $16[\mu sec]$, while the task specification is generated online, as a non-realtime, interruptible task. The Orocos framework takes care of the thread-safe communication and data flow between the two tasks.

B. Cube-Corner Experiment

First the off-line compliant path planner is used to generate a path going though a set of given configurations. The paths generated for the experiment include complex contact formations, containing point contacts, line contacts and plane contacts, as well as combinations of these contact types. Then, these paths, together with the kinetic and potential energy that are tuned for this cube-corner setup, are provided to our implementation of the approach, and immediately executed under force feedback. The task specification is generated online and provided to the controller. The result is a smooth motion of the cube in contact with the corner, along the planned path.

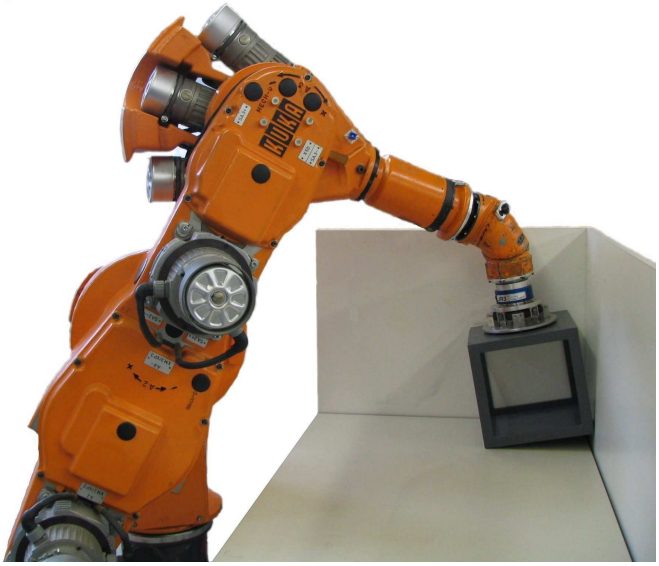

Fig. 6. Experimental setup: the Kuka 361 six degree of freedom industrial robot, manipulating a cube in contact with a corner.

VII. Conclusions

This paper presents the *Compliant Task Generator*: an approach to 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 positions $\boldsymbol{x}_{1\ldots m}$ and their corresponding contact formations $CF_{1\ldots n}$. The hybrid compliant force controller expects a desired velocity $\boldsymbol{t_d}$, position $\boldsymbol{x_d}$ and force $\boldsymbol{w_d}$ at each time-step, together with their velocity and force controlled subspaces $\boldsymbol{T}$ and $\boldsymbol{W}$. The conversion of the discrete planner primitives into a continuous path represented by the controller primitives, is processed separately for the velocity and force controlled subspaces. The former uses a constant kinetic energy E_{kin} of the manipulated object with mass $\boldsymbol{M}$, and the latter a constant potential energy E_{pot} in the elementary contacts with stiffness $\boldsymbol{K}$.

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, presented in literature. It is applicable to any compliant motion between polyhedral objects, and more general and simple then previously presented ad-hoc [3] or rule-based methods. A task-specific input of only four parameters (E_{kin}, E_{pot}, $\boldsymbol{M}$ and $\boldsymbol{K}$) is sufficient to specify the desired dynamic interaction between the moving 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 path by a manipulator, under force feedback. 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.

Our next research efforts will focus on the link in the other direction: the feedback from the compliant controller to the compliant planner. We plan to integrate online estimators, based on the Bayesian approach [4], to collect data during the force controlled execution of a planned path [15]. This can be geometrical data about the objects in contact or topological information about the current contact formation and contact formation transitions. Based on this new data, the planner can decide to re-plan part of the compliant path. The Free Software Bayesian Filtering library [8] offers a unifying framework for all Bayesian filters, that will serve our needs for the integration of online estimators such as Kalman filters, Extended Kalman filters and particle filters.

Acknowledgment

All authors gratefully acknowledge the financial support by K.U.Leuven's Concerted Research Action GOA/99/04 and the U.S. National Science Foundation under grant IIS-0328782.

References

[1] H. Bruyninckx. Open RObot COntrol Software. `http://www.orocos.org/`.

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

[3] H. Bruyninckx and J. 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.

[4] H. Bruyninckx, J. De Schutter, and Tine Lefebvre. Autonomous compliant motion: The Bayesian approach. In *Proc. IEEE/RSJ Int. Conf. Int. Robots and Systems*, pages 2310–2316, Takamatsu, Japan, 2000.

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

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

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

[8] K. Gadeyne. BFL: Bayesian Filtering Library. http://people.mech.kuleuven.ac.be/~kgadeyne/bfl.html.

[9] X. Ji and J. Xiao. Towards random sampling with contact constraints. In *Int. Conf. Robotics and Automation*, San Francisco, CA, 2000.

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

[11] 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.

[12] 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.

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

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

[15] Peter Slaets, Tine Lefebvre, Herman Bruyninckx, and Joris De Schutter. Construction of a Geometrical 3-D Model from Sensor Measurements Collected during Compliant Motion. *IEEE Trans. Rob*, 2004. Submitted.

[16] Qi Wang, J. 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.

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

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