2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

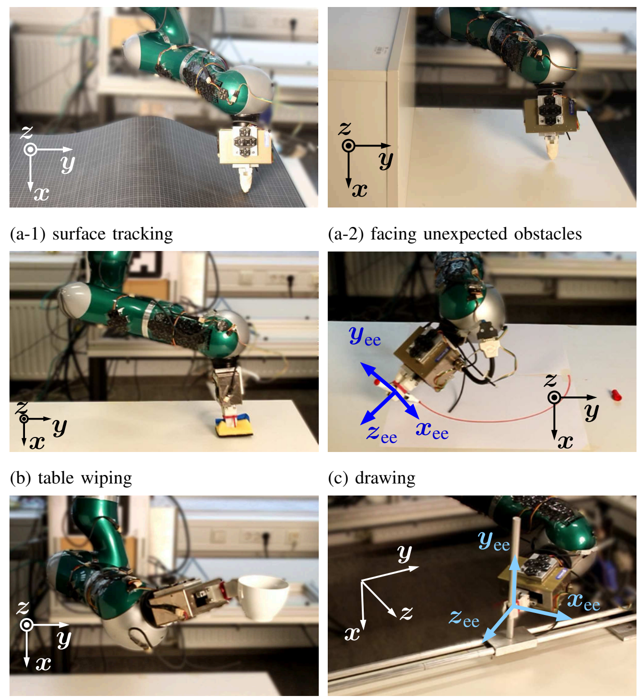
A Practical Approach to Generalized Hierarchical Task Specification for Indirect Force Controlled Robots

Ewald Lutscher and Gordon Cheng
Institute for Cognitive Systems (www.ics.ei.tum.de)
Technische Universität München

***Abstract*— The main contribution of this paper is the general formulation of force and positioning tasks on joint and Cartesian level for indirect force controlled robots and combining them in a strict hierarchical way. As a secondary contribution, we provide a simple and intuitive programming paradigm, using the developed formulation.**

By building on the well-established indirect force control scheme, which is often already provided for commercial robots, we provide application programmers with a useful tool for specifying tasks, involving positioning and force components.

Different physical interaction tasks have been implemented to show the potential of the proposed method and discuss the general advantages and drawbacks.



(a-1) surface tracking (a-2) facing unexpected obstacles

(b) table wiping (c) drawing

(d) cup holding (e) constrained manipulation

Fig. 1. Example tasks with mixed positioning and force components

I. Motivation

Compliance is a compulsory requirement for robots in unstructured environments. A standard approach to realize compliance are indirect force controllers (*IFC*), e.g. the seminal impedance control framework [1]. What all IFC schemes have in common, is the implementation of a virtual mechanical relationship between the physical and a virtual manipulator, resulting in indirect control of the interaction forces by specifying set points for the virtual manipulator. The major drawback of poor accuracy, is outplayed by the increased interaction safety and robustness to environmental uncertainties and unexpected collisions.

Traditional methods are often applied to generate set points for the IFC to regulate either a desired position or a desired force, what does not exploit the full potential of this scheme. In unstructured environments, it is difficult to clearly separate force and positioning tasks. An example is opening a spring loaded door, where a usually unknown interaction force has to be applied in order to operate the mechanism, while simultaneously regulating the pose of the end effector along an uncertain trajectory. The different subtasks are often contradicting and it requires usually some tuning from the application programmer to obtain satisfying results.

In this work we present a generalized programming layer for IFCs, regulating position and forces simultaneously, both on joint and Cartesian level. By defining subtasks in appropriate subspaces, the available degrees of freedom for lower priority tasks are increased. Our approach explicitly aims for tasks, where the demands on accuracy are relaxed, which applies for a broad palette of tasks in human environments. For example, it is not a crucial requirement that a table is wiped with a certain contact force or if an object is placed accurate to a millimeter.

The remainder of this paper is structured as follows. In section II the related work is summarized, Section III provides the basic theoretical background. In section IV the general task formulation is stated and section V shows how a task is composed of different prioritized subtasks. Section VI finally shows some basic properties of our approach and demonstrates the potential on different exemplary tasks, implemented on a manipulator running a joint space impedance controller. A brief summary is provided in section VII.

II. Related Work

Compliant control involving force and positioning tasks has been investigated elaborately in the last decades. Intensive surveys of the most popular schemes can be looked up in [2], [3]. From our point of view, indirect force control is the most promising approach when facing unstructured environments or physical interaction with humans, due to its robustness to unexpected contact events and invariance to environmental properties. This comes at the cost of decreased accuracy both in position and force tracking. However, this

This work is supported in part within the DFG excellence initiative research cluster Cognition for Technical Systems - CoTeSys (www.cotesys.org).

drawback can be deliberately accepted for many tasks in human environment.

Treatment of multiple tasks can be basically approached in two ways. The first is by assigning different weights to the usually concurring tasks. The second is strict separation of tasks via nullspace mapping. We favor the second method, since the weighting strategy requires additional tuning of the weights and subtasks are not separated in a clean way. Combining multiple tasks in a hierarchical manner using nullspace mapping dates back to [4], where it has been done for kinematic control and was used to resolve the manipulator's kinematic redundancy. The basic concepts have been used and expanded since then in many publications e.g. [5], [6], [7]. The focus lies mainly on redundancy resolution, without regarding lower dimensional subtasks. Most of the works are limited to kinematic or force control only. To our best knowledge, a combination of force and position tasks within an IFC framework has never been treated the way presented in this paper.

Considering task specification, the extensive research done by the group around De Schutter in the last years has to be mentioned [8], [9]. Their work is based on the concept of the task-frame formalism [10] and states a unifying method to incorporate external sensing and potential estimation errors within the definition of a task, by choosing appropriate object and feature frames. Like most frameworks building on the task frame formalism, the underlying controller is lacking an inherent compliance and is not considering unexpected collisions. Their approach follows another philosophy and is dedicated more for tasks where high accuracy is required, like in industrial setups.

III. Theoretical Background

A. Manipulator Representation

The configuration of a manipulator with n degrees of freedom (DoF) is defined by a set of n generalized coordinates $\boldsymbol{q}$, which are for revolute joints usually the joint angles. The Cartesian pose of the end effector can be denoted with a vector $\boldsymbol{x} = [\boldsymbol{p} \quad \boldsymbol{o}]^T$ with the three dimensional position $\boldsymbol{p} = [x\ y\ z]^T$ and a vector $\boldsymbol{o}$, describing the orientation of the frame. The dimension and unit of $\boldsymbol{o}$ depend on the chosen orientation representation, e.g. for fixed angles $\boldsymbol{o} = [\phi \quad \theta \quad \psi]^T$, where ϕ, θ and ψ are the rotation angles around the x, y and z axis.

If $\boldsymbol{o}$ is chosen that way, $\dot{\boldsymbol{x}}$ denotes the generalized end effector velocity, or twist and is related to the joint velocities $\dot{\boldsymbol{q}}$ via the base Jacobian $\boldsymbol{J}(\boldsymbol{q})$, stating the instantaneous forward kinematics

$$\dot{\boldsymbol{x}} = \begin{pmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{pmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{1}$$

with $\dot{\boldsymbol{p}}$ and $\boldsymbol{\omega}$ being three-dimensional vectors, representing the translational and angular velocity of the end effector.

Another important property of $\boldsymbol{J}$ is, that its transpose relates the three-dimensional end effector forces $\boldsymbol{f}$ and moments $\boldsymbol{m}$, to joint torques

$$\boldsymbol{\tau} = \boldsymbol{J}^T\boldsymbol{h},$$

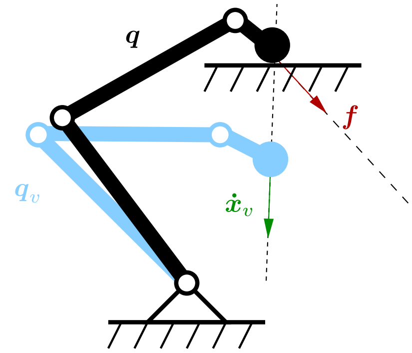


Fig. 2. Motion and interaction forces of the physical manipulator (black) are controlled indirectly by generating set points for the virtual manipulator (blue).

where $\boldsymbol{h} = (\boldsymbol{f} \quad \boldsymbol{m})^T$ is called the end effector wrench. The wrench due to applied torques can be computed vise versa, using a generalized inverse (e.g. the Moore-Penrose pseudoinverse) of the transposed base Jacobian.

$$\boldsymbol{h} = \boldsymbol{J}^{T+}\boldsymbol{\tau} \tag{2}$$

B. Indirect Force Control

Indirect force control is characterized by regulating the configuration of a virtual manipulator, represented by its generalized coordinates $\boldsymbol{q}_v$ (see Fig. 2). The relation of this virtual manipulator to the physical manipulator $\boldsymbol{q}$ is stated via a virtual mechanical relationship, established either at Cartesian or joint-level. For our work, we consider IFCs which state a virtual stiffness relationship between the joint-space position difference $(\boldsymbol{q}_v - \boldsymbol{q})$ and applied joint torque $\boldsymbol{\tau}$ via a stiffness matrix $\boldsymbol{K}$. The simplest IFC variant is stiffness control, which basically corresponds to a PD-controller with compensation of the gravitational forces:

$$\boldsymbol{\tau} = \boldsymbol{K}(\boldsymbol{q_v} - \boldsymbol{q}) - \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}), \tag{3}$$

where $\boldsymbol{K}$ and $\boldsymbol{D}$ are $n \times n$ diagonal matrices and $\boldsymbol{g}(\boldsymbol{q})$ are the torques for compensating gravitational effects. By determining a set-point $\boldsymbol{q}_v$, a static interaction torque equal to $\boldsymbol{K}(\boldsymbol{q_v} - \boldsymbol{q})$ is indirectly commanded. The general stability and robustness of IFC schemes was shown in multiple publications, e.g. [11], [12].

IV. Generalized Position and Force Regulation in IFC

A. General Task Formulation

Let $\boldsymbol{\sigma} \in \mathbb{R}^m$ be a general task variable with the desired value $\boldsymbol{\sigma}_d$,

$$\tilde{\boldsymbol{\sigma}}(t) = \boldsymbol{\sigma}_d(t) - \boldsymbol{\sigma}(t)$$

the according task error and the $m \times n$ task Jacobian $\boldsymbol{A}_\sigma = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{q}_v}$, so that

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{A}_\sigma \dot{\boldsymbol{q}}_v. \tag{4}$$

The task can be defined in Cartesian ($m = 6$) or joint space ($m = n$). Note that (4) is defined for the velocity of the virtual manipulator $\dot{\boldsymbol{q}}_v$. A classical approach for a task level control scheme is to impose

$$\dot{\boldsymbol{\sigma}}_d = \boldsymbol{\Lambda}_\sigma \tilde{\boldsymbol{\sigma}},$$

where $\boldsymbol{\Lambda}_\sigma$ is usually a diagonal, positive definite $m \times m$ gain matrix that tunes the convergence speed of the task error components to $\mathbf{0}$. By inverting (4), the general task controller is

$$\dot{\boldsymbol{q}}_{v_d} = \boldsymbol{A}_\sigma^+ \boldsymbol{\Lambda}_\sigma \tilde{\boldsymbol{\sigma}}, \tag{5}$$

where $\dot{\boldsymbol{q}}_{v_d}$ is the desired velocity of the virtual manipulator. As $\dot{\boldsymbol{q}}_v$ is a virtual quantity, $\dot{\boldsymbol{q}}_{v_d} = \dot{\boldsymbol{q}}_v$ can be assumed.

B. Joint Position / Cartesian Pose Regulation

Positioning controllers building on an IFC framework usually provide trajectories for $\boldsymbol{q}_v$, without regarding the actual motion of $\boldsymbol{q}$. How well $\boldsymbol{q}$ tracks $\boldsymbol{q}_v$ depends on the IFC implementation, especially on the stiffness $\boldsymbol{K}$. A higher stiffness results in better positioning accuracy, while a low stiffness is beneficial for contact stability. Also, fast trajectories are tracked worse in general, due to the faster dynamics of $\boldsymbol{q}_v$ in comparison to $\boldsymbol{q}$. Such type of controller is often used in applications, where the IFC is supposed to compensate for unexpected collisions and where accuracy plays a minor role. This applies basically for tasks in unstructured environments, where positioning accuracy is deliberately traded for safer physical interaction.

The trivial task Jacobian for the virtual joint position $\boldsymbol{q}_v$ is the $n \times n$ identity matrix $\boldsymbol{I}_n$ ($\dot{\boldsymbol{q}}_v = \boldsymbol{I}_n \dot{\boldsymbol{q}}_v$). Following the template (5), this leads to the simple regulator

$$\dot{\boldsymbol{q}}_v = \boldsymbol{\Lambda}_q \tilde{\boldsymbol{q}}_v.$$

Using standard instantaneous inverse kinematic methods, a controller for the Cartesian pose $\boldsymbol{x}_v$ of an arbitrary frame on the manipulator can be derived. With (1), the task Jacobian for $\boldsymbol{x}_v$ is equivalent to the base Jacobian of the virtual manipulator $\boldsymbol{J}_v = \boldsymbol{J}(\boldsymbol{q}_v)$, so that

$$\dot{\boldsymbol{x}}_v = \boldsymbol{J}_v \dot{\boldsymbol{q}}_v.$$

The Cartesian pose regulator is hence

$$\dot{\boldsymbol{q}}_v = \boldsymbol{J}_v^+ \boldsymbol{\Lambda}_x \tilde{\boldsymbol{x}}_v.$$

Note that there is no feedback of the physical entities and $\dot{\boldsymbol{q}}_v$ is set in an open loop way.

C. Joint Torque / Wrench Regulation

In an IFC scheme like (3) we have only influence on the static components of the interaction torque $\boldsymbol{\tau}$ by setting $\boldsymbol{q}_v$, hence (3) is simplified to

$$\boldsymbol{\tau} = \boldsymbol{K}(\boldsymbol{q_v} - \boldsymbol{q}). \tag{6}$$

Considering quasi static conditions of the physical manipulator $\boldsymbol{q}$, the derivative of (6) is

$$\dot{\boldsymbol{\tau}} = \boldsymbol{K}\dot{\boldsymbol{q}}_v. \tag{7}$$

Hence, the task Jacobian for static interaction torques is the stiffness matrix $\boldsymbol{K}$ and the according interaction torque controller has the form

$$\dot{\boldsymbol{q}}_v = \boldsymbol{K}^{-1} \boldsymbol{\Lambda}_\tau \tilde{\boldsymbol{\tau}} \tag{8}$$

Since $\boldsymbol{K}$ is a positive definite diagonal matrix, the Moore-Penrose inverse can be replaced by the regular inverse here. In theory one could also compensate for the effects of $\dot{\boldsymbol{q}}$ in (8) but this feedback would add a potential source of instability and in practice the effect of such a compensation is negligible.

TABLE I

SPECIFICATIONS OF THE FOUR BASIC TASK TYPES

type	$\boldsymbol{\sigma}$	$\boldsymbol{A}_\sigma$	$\boldsymbol{\Lambda}_\sigma$
cart. pose	$\boldsymbol{x}_v$	$\boldsymbol{J}_v$	$\boldsymbol{\Lambda}_x$
joint position	$\boldsymbol{q}_v$	$\boldsymbol{I}_n$	$\boldsymbol{\Lambda}_q$
wrench	$\boldsymbol{h}$	$\boldsymbol{J}^{T+}\boldsymbol{K}$	$\boldsymbol{\Lambda}_h$
joint torque	$\boldsymbol{\tau}$	$\boldsymbol{K}$	$\boldsymbol{\Lambda}_\tau$

For the static wrench controller, we use (2) to obtain the relation

$$\dot{\boldsymbol{h}} = \frac{\partial \boldsymbol{h}}{\partial t} = \frac{\partial \boldsymbol{J}^{T+}}{\partial t}\boldsymbol{\tau} + \boldsymbol{J}^{T+}\dot{\boldsymbol{\tau}}. \tag{9}$$

Applying the same simplifying assumptions as for (8) regarding the quasi static conditions on $\boldsymbol{q}$, we can remove the derivative of the base Jacobian's pseudoinverse. Using (7), (9) can now be rewritten as

$$\dot{\boldsymbol{h}} = \boldsymbol{J}^{T+}\boldsymbol{K}\dot{\boldsymbol{q}}_v$$

and the static wrench regulator can be stated as

$$\dot{\boldsymbol{q}}_v = (\boldsymbol{J}^{T+}\boldsymbol{K})^+ \boldsymbol{\Lambda}_h \tilde{\boldsymbol{h}}.$$

Note, that here the base Jacobian of the actual configuration $\boldsymbol{q}$ has to be used. Table I summarizes the four common task types introduced in this section.

V. HIERARCHICAL TASK PROGRAMMING

In this section we show how a high level task can be programmed by combining the different task types from section IV to a set of subtasks. First we show how each subtask can be expressed in a certain subspace of $\mathbb{R}^m$ to increase the redundancy of the robot with respect to that task. Standard nullspace mapping techniques can then be applied to combine the subtasks in a hierarchical way.

A. Defining Tasks in Subspaces

For a conventional 7-DoF manipulator, a six dimensional Cartesian task leaves only one DoF for a secondary objective. Therefore, multiple task frameworks are usually of less interest for such manipulators. To increase the potential number of subtasks, the task $\boldsymbol{\sigma}$ can be expressed in a certain subspace of interest $\mathbb{S}_\sigma \subseteq \mathbb{R}^m$. This subspace is characterized by a set of orthonormal vectors, which are the columns of a matrix $\boldsymbol{S}_\sigma$, having some similarity to the compliance selectivity matrix known from hybrid position/force control [13]. While in hybrid position/force control an additional specification of the task frame is required, this information is already included in $\boldsymbol{S}_\sigma$. Also, the subspaces of different subtasks do not have to be orthogonal but can be defined in any way. It is up to the application programmer to select $\boldsymbol{S}_\sigma$ in a way, that irrelevant directions are neglected, so the additional degrees of freedom can be used to fulfill lower priority tasks. The task Jacobian $\boldsymbol{A}_\sigma$, has to be replaced with

$$\hat{\boldsymbol{A}}_\sigma = \boldsymbol{S}_\sigma^T \boldsymbol{A}_\sigma,$$

which is $\boldsymbol{A}_\sigma$ expressed in $\mathbb{S}_\sigma$.

B. Enforcing a Task Hierarchy

Using the general task formalism from section IV, we can now define an arbitrary large set of subtasks $[\boldsymbol{\sigma}_1 \dots \boldsymbol{\sigma}_k]$, with k as the number of subtasks, sorted by descending priority and optionally expressed in a certain subspace as described in section V-A.

Hence, a hierarchical controller can now be derived using nullspace projection methods to enforce a strict task hierarchy. With $\mathrm{Ker}(\boldsymbol{X})$ denoting the orthonormal basis of the kernel of some linear map $\boldsymbol{X}$, an orthogonal projection operator

$$N(\boldsymbol{X}) = \mathrm{Ker}(\boldsymbol{X})^T \mathrm{Ker}(\boldsymbol{X})$$

is defined, which projects a vector in the nullspace of $\boldsymbol{X}$.

With i as the task iterator and the projector

$$\boldsymbol{N}_i = N([\hat{\boldsymbol{A}}_0 \dots \hat{\boldsymbol{A}}_{i-1}]^T), \tag{10}$$

the k subtasks can be combined in a recursive way by

$$\begin{aligned} \dot{\boldsymbol{q}}_{v_0} &= \boldsymbol{0}, \qquad \hat{\boldsymbol{A}}_0 = \boldsymbol{0}_n \\ \dot{\boldsymbol{q}}_{v_i} &= \dot{\boldsymbol{q}}_{v_{i-1}} + \boldsymbol{N}_i \hat{\boldsymbol{A}}_i^+ (\boldsymbol{\Lambda}_i \tilde{\boldsymbol{\sigma}}_i - \hat{\boldsymbol{A}}_i \dot{\boldsymbol{q}}_{v_{i-1}}), \end{aligned} \tag{11}$$

with $\boldsymbol{0}_n$ as an $n \times n$ matrix with all entries equal to 0. Every subtask is projected in the nullspace of all the higher priority tasks, where the mapping (10) guarantees, that the task hierarchy is not violated, what is elaborately discussed in [14]. The term $\hat{\boldsymbol{A}}_i \dot{\boldsymbol{q}}_{v_{i-1}}$ is the compensation for the effects of the higher priority tasks on the task level controller $\boldsymbol{\Lambda}_i \tilde{\boldsymbol{\sigma}}_i$.

It has to be mentioned, that (11) will not produce the optimal solution, since the projection is not taken into account in the pseudoinverse. The optimal solution would be obtained by

$$\dot{\boldsymbol{q}}_{vi} = \dot{\boldsymbol{q}}_{v_{i-1}} + (\hat{\boldsymbol{A}}_i \boldsymbol{N}_i)^+ (\boldsymbol{\Lambda}_i \tilde{\boldsymbol{\sigma}}_i - \hat{\boldsymbol{A}}_i \dot{\boldsymbol{q}}_{v_{i-1}}),$$

where the nullspace mapping is incorporated before the computation of the pseudoinverse. However, this method introduces additional singularities in the control [15]. To avoid those problems for the present work, we settle for the solution obtained with (11) and postpone proper treatment of singularities to future work, respectively refer to some recent publications on this, e.g. [16].

C. Incorporating Joint Limits

In general a manipulator should be prevented from physically hitting its joint limits. A simple solution is to clamp the joint if it reaches a critical distance and moves towards the limit. However, this clamping may lead to convergence to a non-optimal state if it is not accounted for in (5), see [17]. Hence it is important to capture them before computing (11). Referring to [6], we set the according row in all the task Jacobians $\hat{\boldsymbol{A}}_{0\dots k}$ to $\boldsymbol{0}$, if a joint hits its limit. This basically removes the respective joint from the computations in (11) and treats the joint as a static connection. By checking the sign of the joint velocity, the clamping can be removed if the joint is about to leave the limit.

While this is a simple way of treating joint limits, this clamping leads to an instantaneous rank drop of the $\hat{\boldsymbol{A}}_{0\dots k}$ matrices, what gets propagated to the controller and leads to discontinuous solutions for $\dot{\boldsymbol{q}}_v$. In general such discontinuities should be avoided. A detailed discussion of proper treatment of such inequality constraints would go beyond the scope of the present work. Some recent works treat this problem closely for kinematic control [18], [19].

TABLE II

SET OF SUBTASKS FOR SURFACE TRACKING

prio	type	$\boldsymbol{\sigma}_d$	$\boldsymbol{\Lambda}_\sigma$	$\boldsymbol{S}_\sigma$
1	wrench	$8N$	200	$[1\ 0\ 0\ 0\ 0\ 0]^T$
2	cart. pose	$\begin{bmatrix} y_{\text{init}} + R\cos(2f\pi t) - R \\ z_{\text{init}} + R\sin(2f\pi t) \\ \boldsymbol{o}_{\text{init}} \end{bmatrix}$	$10\boldsymbol{I}_4$	$\begin{bmatrix} 0\ 0\ 0\ 0\ 0 \\ \boldsymbol{I}_5 \end{bmatrix}$
3	joint position (joint torque)	$\boldsymbol{0}$ $(\boldsymbol{0})$	$0.6\boldsymbol{I}_7$ $(100\boldsymbol{I}_7)$	$\boldsymbol{I}_7$ $(\boldsymbol{I}_7)$

VI. EXPERIMENTAL RESULTS

A. Implementation Details and Hardware

The experiments have been carried out on our KUKA LBR-IV lightweight arm. The manipulator was running a joint space impedance controller, which details can be found in [20]. The experimental setups are depicted in Fig. 1. The rate of the discrete controller was $r = 500Hz$ and the default stiffness was $\boldsymbol{K} = 400\boldsymbol{I}_7 Nm/$rad. The task gains $\boldsymbol{\Lambda}_\sigma$ where chosen heuristically.

B. Example Applications

The following examples have been implemented to show how our approach can be used to program a variety of tasks by combining positioning and force type subtasks in joint and Cartesian space. By considering only the basic task types from section V-A, whose full task Jacobians can be found in table I, programming of complex tasks is reduced to specifying a set of subtasks, each defined by a task type, a desired task variable $\boldsymbol{\sigma}_d \in \mathbb{S}_\sigma$, the according gain matrix $\boldsymbol{\Lambda}_\sigma$ and an appropriate subspace matrix $\boldsymbol{S}_\sigma$.

a) Surface Tracking (Fig. 1(a)): This is a classical contact task, where the manipulator is supposed to exert a constant force on a surface while moving along a certain trajectory. Here a circular trajectory is tracked, starting at $\boldsymbol{p}_{\text{init}}$ with radius R and frequency f in the y-z-plane, while keeping the constant initial orientation $\boldsymbol{o}_{\text{init}}$. The task is summarized in table II. As for the third subtask, one could either choose a positioning task, keeping the joints away from their limits ($\boldsymbol{q}_{v_d} = \boldsymbol{0}$) or alternatively minimizing the joint torques ($\boldsymbol{\tau}_d = \boldsymbol{0}$).

The first trial was conducted on a curved surface with unknown flexibility (see Fig. 1(a-1)). The impacts of execution speed and stiffness $\boldsymbol{K}$ of the IFC are demonstrated with this example. The tracking errors for the force and positioning subtasks are plotted in Fig. 4 for alternating stiffness and in Fig. 3 for alternating execution speed. As stated in section IV-B, the quality of position tracking increases with higher entries in $\boldsymbol{K}$ and decreases for faster execution speeds. The force tracking error mainly comes from the fact, that the simplified relation (6) does not take IFC rendering errors into account, on which we have usually only little influence, using a built-in IFC scheme. With progressive execution speed, one also observes the influence of dynamic effects, which are also not accounted for in (6). Increasing the stiffness also leads to slightly worse force tracking, due to higher sensibility of the

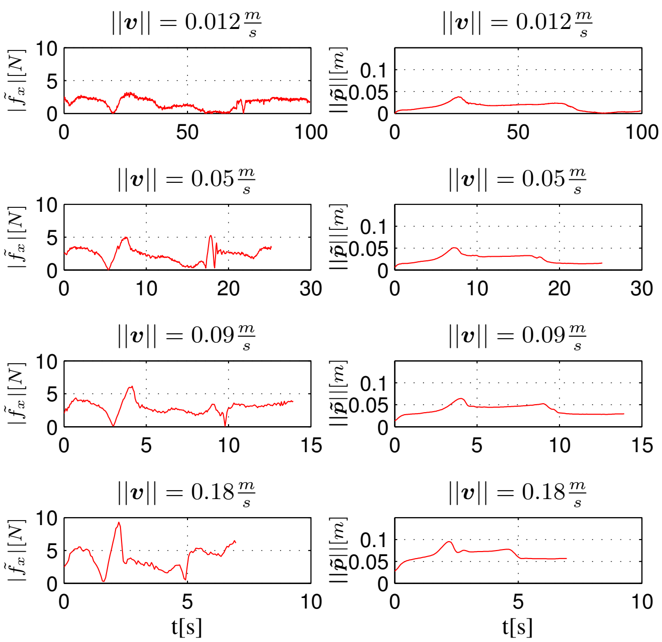


Fig. 3. Force and positioning error for varying execution speed during a surface tracking task.

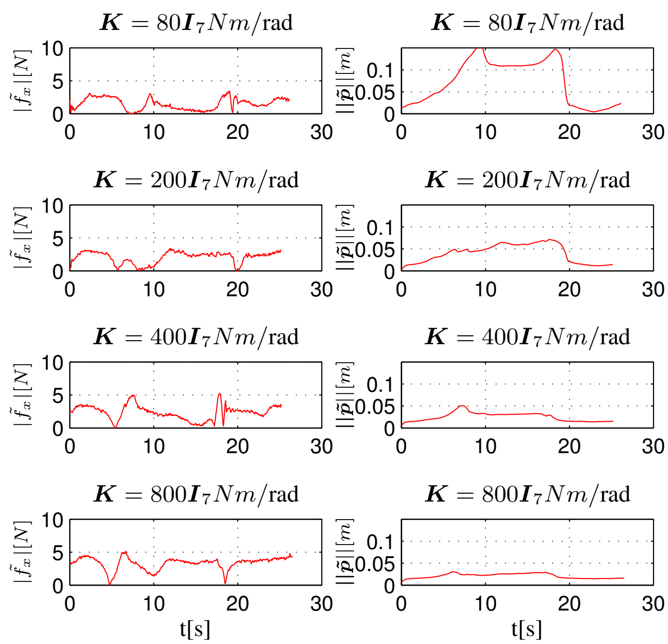


Fig. 4. Force and positioning error for varying virtual joint stiffness during a surface tracking task.

TABLE III

SET OF SUBTASKS FOR TABLE WIPING

prio	type	$\boldsymbol{\sigma}_d$	$\boldsymbol{\Lambda}_\sigma$	$\boldsymbol{S}_\sigma$
1	wrench	$8N$	200	$[1\ 0\ 0\ 0\ 0\ 0]^T$
2	cart. pose	$\begin{bmatrix} y_{\text{init}} + R_1\cos(2f_1\pi t) - R_1 \\ z_{\text{init}} + R_2\sin(2f_2\pi t) \\ \boldsymbol{o}_{\text{init}} \end{bmatrix}$	$10\boldsymbol{I}_5$	$\begin{bmatrix} 0\,0\,0\,0\,0 \\ \boldsymbol{I}_5 \end{bmatrix}$
3	joint torque	$\boldsymbol{0}$	$100\boldsymbol{I}_7$	$\boldsymbol{I}_7$

static interaction torques to $\dot{\boldsymbol{q}}_v$. These are clear downsides when using an IFC scheme.

The second experiment was conducted with an unknown obstacle blocking the path of the manipulator (see Fig. 1(a-2)). Here the advantage of IFC shows up. Due to its capabilities of handling such unexpected collisions, the manipulator remains stable and gives, for example some high-level application enough time to react on the event. Also, if the joint torque minimization subtask is set, the nullspace of the higher priority subtasks is used to compensate for collisions occurring at the "elbow"-joint.

b) Table Wiping (Fig. 1(b)): This task is similar to the surface tracking and is an example of a real-world task, where neither very accurate position, nor force tracking is required. The end effector is supposed to track a sinusoidal trajectory back and forth in the y-z-plane, determined by the amplitudes R_1 and R_2 with according frequencies f_1 and f_2. The task description is summarized in table III.

c) Circle Drawing (Fig. 1(c)): This is also a modification of the surface tracking task and demonstrates how simple it is to incorporate high-level knowledge by setting an appropriate subspace matrix $\boldsymbol{S}_\sigma$. With the pen being aligned with the end effector y-axis y_{ee}, the positioning subtask is invariant to rotations around y_{ee}. Hence the task is defined equally to the surface tracking example, despite that the rotational part is described in terms of rotations around the x_{ee} and z_{ee} end effector axes only. This relaxation of the task constraints, gives the lower priority tasks more freedom,

TABLE IV

SET OF SUBTASKS FOR CIRCLE DRAWING

prio	type	$\boldsymbol{\sigma}_d$	$\boldsymbol{\Lambda}_\sigma$	$\boldsymbol{S}_\sigma$
1	wrench	$6N$	200	$[1\ 0\ 0\ 0\ 0\ 0]^T$
2	cart. pose	$\boldsymbol{S}_2^T \begin{bmatrix} 0 \\ y_{\text{init}} + R\cos(2f\pi t) - R \\ z_{\text{init}} + R\sin(2f\pi t) \\ \boldsymbol{o}_{\text{init}} \end{bmatrix}$	$10\boldsymbol{I}_4$	$\boldsymbol{T}_{IR} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
3	joint position	$\boldsymbol{0}$	$0.6\boldsymbol{I}_7$	$\boldsymbol{I}_7$

resulting in smaller joint velocities and a smoother motion due to the successful avoidance of a joint limit, which is hit if this relaxation is not applied. The resulting discontinuity is depicted in Fig. 5. With $\boldsymbol{R}_{ee}$ being the end effector orientation matrix, $\boldsymbol{T}_{IR} = \text{diag}(\boldsymbol{I}_3, \boldsymbol{R}_{ee})$ transforms the part related to orientations of a Cartesian subspace matrix from end-effector to base coordinates. Here, $\boldsymbol{\sigma}_d$ is defined in the global frame and then expressed in $\mathbb{S}_\sigma$ by pre-multiplication with $\boldsymbol{S}_2^T$, denoting the transposed of the subspace matrix for the second subtask. The third subtask is to keep the joints away from their limits. Table IV shows the task parameters.

d) Cup Holding (Fig. 1(d)): Highest priority is given to a controller holding some fixed orientation. Lower priority tasks can now be defined in any way, e.g. to reach a certain point, react to external sensor information etc. without considering orientation anymore. We chose for example minimization of joint torques as secondary task, making it possible to push the manipulator around manually. See table

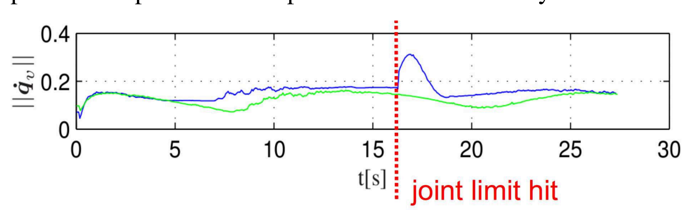


Fig. 5. Norm of joint velocities for different task specifications for circle drawing. Blue: keeping constant orientation. Green: Rotation around end effector y-axis permitted.

TABLE V

SET OF SUBTASKS FOR CUP HOLDING

prio	type	$\boldsymbol{\sigma}_d$	$\boldsymbol{\Lambda}_\sigma$	$\boldsymbol{S}_\sigma$
1	cart. pose	$\boldsymbol{o}_{\text{init}}$	$10\boldsymbol{I}_3$	$\begin{bmatrix} \boldsymbol{0}_3 \\ \boldsymbol{I}_3 \end{bmatrix}$
2	joint torque	$\boldsymbol{0}$	$100\boldsymbol{I}_7$	$\boldsymbol{I}_7$

TABLE VI

SET OF SUBTASKS FOR CONSTRAINED MANIPULATION

prio	type	$\boldsymbol{\sigma}_d$	$\boldsymbol{\Lambda}_\sigma$	$\boldsymbol{S}_\sigma$
1	wrench	$10N\boldsymbol{d}$	$10\boldsymbol{I}_3$	$\begin{bmatrix} \boldsymbol{I}_3 \\ \boldsymbol{0}_3 \end{bmatrix}$
2	cart. pose	$\boldsymbol{S}_2^T\boldsymbol{x}_{\text{init}}$	$10\boldsymbol{I}_2$	$\boldsymbol{T}_{IR}\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$
3	joint position	$\boldsymbol{0}$	$0.6\boldsymbol{I}_7$	$\boldsymbol{I}_7$

V for task description.

e) Operating Unknown Constrained Mechanisms (Fig. 1(e)): By using a simple constrained estimator [21], which gives us the three-dimensional direction vector of possible translational motion $\boldsymbol{d}$, a simple controller for operating constrained mechanisms can be designed. A constant force is assigned along $\boldsymbol{d}$, while the end effector orientation should remain unchanged as far as possible, allowing orientation around the end effector y-axis similar to the circle drawing task. With this we take advantage of previous knowledge on the gripper geometry, which allows rotation around y_{ee} when grasping a handle. Remaining degrees of freedom are used again to keep joints away from their limits. The according task specification can be looked up in table VI.

VII. CONCLUSION

We presented a generalized hierarchical task specification framework for indirect force controlled robots in uncertain environments. Only very vague knowledge on environment geometry is enough to program interaction tasks involving position and force type commands on joint and Cartesian level in a simple and intuitive way. Enforcing a strict task hierarchy reduces the need of tedious parameter tuning and the reduction of subtasks to certain subspaces makes the proposed method also applicable on non-redundant robots. The usage of the well-established IFC scheme, which is often already provided in commercial robots, makes it unnecessary to implement a new low level controller and integrate it into a running system. We showed the simple usage of our framework by implementing various interaction tasks on a 7 DoF manipulator. Even though the approach is not suitable for high precision tasks like in industrial applications, it provides a simple and intuitive interface for mixed positioning and force tracking tasks, where high accuracy is not that crucial.

REFERENCES

[1] N. Hogan, "Impedance control: An approach to manipulation," in *American Control Conference, 1984*. IEEE, 1984, pp. 304–313.

[2] T. Lefebvre, J. Xiao, H. Bruyninckx, and G. de Gersem, "Active compliant motion: a survey," *Advanced Robotics*, vol. 19, no. 5, pp. 479–499, Jun. 2005.

[3] S. Chiaverini, B. Siciliano, and L. Villani, "A survey of robot interaction control schemes with experimental comparison," *IEEE/ASME Transactions on Mechatronics*, vol. 4, no. 3, pp. 273–285, 1999.

[4] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.

[5] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, pp. 1211–1216, 1991.

[6] P. Baerlocher and R. Boulic, "An inverse kinematics architecture enforcing an arbitrary number of strict priority levels," *The Visual Computer*, vol. 20, no. 6, Jun. 2004.

[7] L. Sentis and O. Khatib, "Task-oriented control of humanoid robots through prioritization," in *IEEERAS/RSJ International Conference on Humanoid Robots*, 2004, pp. 1–16.

[8] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *The International Journal of Robotics Research*, vol. 26, no. 5, p. 433, 2007.

[9] R. Smits, T. Laet, K. Claes, H. Bruyninckx, and J. Schutter, "iTaSC: a tool for multi-sensor integration in robot manipulation," *Multisensor Fusion and Integration for Intelligent Systems*, vol. 2, pp. 235–254, 2009.

[10] M. T. Mason, "Compliance and Force Control for Computer Controlled Manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[11] P. Tomei, "A simple PD controller for robots with elastic joints," *Automatic Control, IEEE Transactions on*, vol. 36, no. 10, pp. 1208–1213, 1991.

[12] N. Hogan, "On the stability of manipulators performing contact tasks," *IEEE Journal on Robotics and Automation*, vol. 4, no. 6, pp. 677–686, 1988.

[13] M. Raibert and J. Craig, "Hybrid position/force control of manipulators," *ASME Journal of Dynamic Systems, Measurement and Control*, pp. 126–133, 1981.

[14] G. Antonelli, "Stability Analysis for Prioritized Closed-Loop Inverse Kinematic Algorithms for Redundant Robotic Systems," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 985–994, Oct. 2009.

[15] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, Jun. 1997.

[16] A. S. Deo and I. D. Walker, "Overview of damped least-squares methods for inverse kinematics of robot manipulators," *Journal of Intelligent & Robotic Systems*, vol. 14, no. 1, pp. 43–68, Sep. 1995.

[17] R. Boulic, R. Mas, and D. Tahlmann, "Interactive identification of the center of mass reachable space for an articulated manipulator," *Advanced Robotics, 1997. ICAR . . .*, 1997.

[18] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic Control of Redundant Manipulators: Generalizing the Task-Priority Framework to Inequality Task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, Aug. 2011.

[19] N. Mansard, O. Khatib, and a. Kheddar, "A Unified Approach to Integrate Unilateral Constraints in the Stack of Tasks," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 670–685, Jun. 2009.

[20] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A unified passivity based control framework for position, torque and impedance control of flexible joint robots," *Robotics Research*, pp. 5–21, 2007.

[21] E. Lutscher, M. Lawitzky, G. Cheng, and S. Hirche, "A control strategy for operating unknown constrained mechanisms," *International Conference on Robotics and Automation (ICRA), 2010 IEEE*, pp. 819–824, 2010.