

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 controller, which often comes already with commercial robots, we provide application programmers with a useful tool for specifying tasks, which involve 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.

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, as the name implies, 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, like in service robotics, it is hard 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 application programming layer for IFCs, regulating position and forces simultaneously, both on joint and Cartesian level. Tasks are formulated in terms of a linear mapping of the virtual manipulator’s joint velocity. After that, standard nullspace projection methods are used to organize different subtasks in a strict hierarchical way. By defining subtasks in appropriate subspaces, the available degrees of freedom for lower priority tasks are increased.

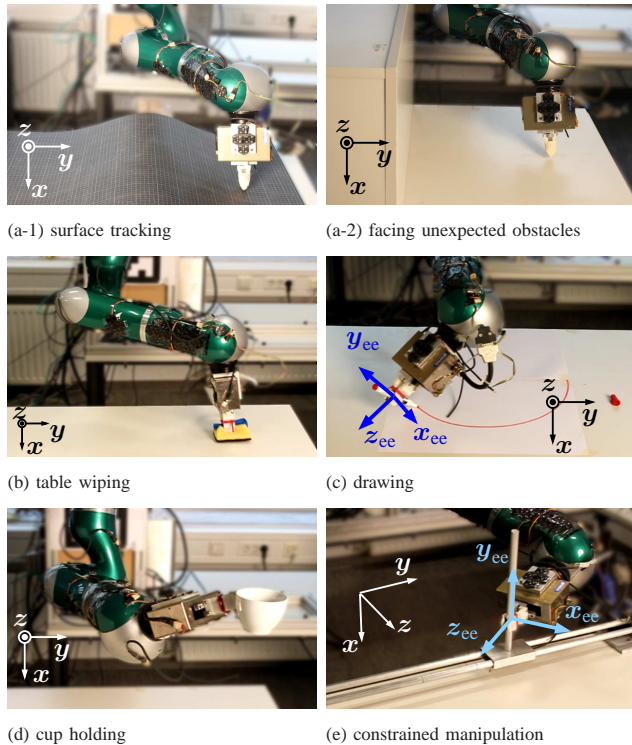


Fig. 1. Example tasks with mixed positioning and force components

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 relevant if a table gets wiped with a contact force of 5 or 10N or if an object is placed accurate to a millimeter. An important benefit of our approach is, that it builds on well-established low-level control methods, which are often already set up for commercially available robots. Hence, it can be used directly, without having to override the robots built-in controller and implementing a whole new framework. Also the simplicity and low requirements on the hardware make the approach in general applicable on a broad class of robots.

Our method combines ideas from different approaches in robot control. An underlying IFC scheme provides the required robustness and stability, which is mandatory for vaguely defined interaction tasks. Breaking the task down to multiple subtasks with different priorities is a commonly used method in task level control for redundant robots, both for positioning and force tasks [2], [3]. The general idea of separating subtasks by defining them in different subspaces is borrowed from hybrid position/force control [4].

The paper is structured in the following way. 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 and impedance controller. A brief summary and outlook 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 [5], [6], [7]. 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 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 [2] and [3], where it has been done on kinematic, respectively force level and was basically used as a tool to resolve the manipulator's redundancy. The basic concepts have been used and expanded since then in many publications e.g. [8], [9], [10], [11]. The focus lies mainly on redundancy resolution, without regarding lower dimensional subtasks. Most of the works also limit oneself 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 this way. Basically due to the introduced inaccuracy related to IFC, mentioned before.

Considering task specification, the extensive research done by the group around De Schutter in the last years has to be mentioned [12], [13]. Their work is based on the concept of the task-frame formalism [14] 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. Their framework builds on detailed modeling of the task geometry and is dedicated for high precision control in more industrial like settings.

III. THEORETICAL BACKGROUND

A. Manipulator Representation

The very basics of robotic manipulation are assumed to be known, hence only the relations directly in connection to the present work are summarized. The configuration of a manipulator with n degrees of freedom (DoF) is defined by a set of n generalized coordinates \mathbf{q} , which are for revolute

joints usually the joint angles. The Cartesian pose of the end effector, but any other frame on the manipulator as well, can be denoted with a vector $\mathbf{x} = [\mathbf{p} \ \mathbf{o}]^T$ with the three dimensional position $\mathbf{p} = [x \ y \ z]^T$ and a vector \mathbf{o} , describing the orientation of the frame. The dimension and unit of \mathbf{o} depend on the chosen orientation representation, e.g. for fixed angles $\mathbf{o} = [\phi \ \theta \ \psi]^T$ where ϕ , θ and ψ are the rotation angles around the x , y and z axis.

The base Jacobian $\mathbf{J}(\mathbf{q})$ relates the six-dimensional, generalized end effector velocity, or twist \mathbf{v} , to the joint velocities $\dot{\mathbf{q}}$ in the instantaneous kinematics

$$\mathbf{v} = \begin{pmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{pmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (1)$$

with $\dot{\mathbf{p}}$ and $\boldsymbol{\omega}$ being three-dimensional vectors, representing the translational and angular velocity of the end effector. Independent from the actual orientation representation, $\dot{\mathbf{x}}$ can always be expressed via \mathbf{v} , using appropriate coordinate transformations.

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

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{h}, \quad (2)$$

where $\mathbf{h} = (\mathbf{f} \ \mathbf{m})^T$ is called the end effector wrench. The wrench due to applied torques is computed vice versa, using the Moore-Penrose pseudoinverse of the transposed Jacobian.

$$\mathbf{h} = \mathbf{J}^{T+} \boldsymbol{\tau} \quad (3)$$

B. Indirect Force Control

Indirect force control is characterized by regulating the configuration of a virtual manipulator, represented by its generalized coordinates \mathbf{q}_v (see Fig. 2). The relation of this virtual manipulator to the physical manipulator \mathbf{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 $(\mathbf{q}_v - \mathbf{q})$ and applied commanded joint torque $\boldsymbol{\tau}_c$ via a stiffness matrix \mathbf{K} . The simplest IFC variant is stiffness control, which basically corresponds to a PD-controller with compensation of the gravitational forces:

$$\boldsymbol{\tau}_c = \mathbf{K}(\mathbf{q}_v - \mathbf{q}) - \mathbf{D}\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}), \quad (4)$$

where \mathbf{K} and \mathbf{D} are $n \times n$ diagonal matrices and $\mathbf{g}(\mathbf{q})$ are the torques for compensating gravitational effects. This controller is designed to achieve a desired static interaction, hence by determining a set-point \mathbf{q}_v , a static interaction torque

$$\bar{\boldsymbol{\tau}}_c = \mathbf{K}(\mathbf{q}_v - \mathbf{q}) \quad (5)$$

is indirectly commanded. For a smooth transition and avoiding large jumps in $\bar{\boldsymbol{\tau}}_c$, which are in general unfavorable, \mathbf{q}_v can be regulated by its time derivative $\dot{\mathbf{q}}_v$. The general stability and robustness of IFC schemes was shown in multiple publications, e.g. [15], [16].

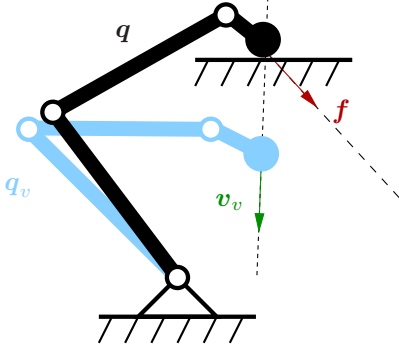


Fig. 2. Motion and interaction forces of the physical manipulator (black) are controlled indirectly by generating set points for the virtual manipulator (blue).

IV. GENERALIZED POSITION AND FORCE REGULATION IN IFC

In this section we will show how manipulator configuration and static interaction torques, respectively Cartesian poses and Interaction wrenches can be regulated in an IFC framework.

A. General Task Formulation

We define a general task variable $\sigma \in \mathbb{R}^m$ and a task error

$$\tilde{\sigma}(t) = \sigma_d(t) - \sigma(t), \quad (6)$$

with the desired task variable σ_d , which can be formulated in joint ($m = n$) or Cartesian space ($m = 6$). Dependencies on the time t will be dropped from now on for the sake of readability. Furthermore we postulate, that the time derivative of the task variable can be written in the form of

$$\dot{\sigma} = \mathbf{A}\dot{q}_v + \eta \quad (7)$$

where \mathbf{A} is a $m \times n$ matrix and η is a m -dimensional vector, combining the terms which do not depend on \dot{q}_v .

A task controller has the general form

$$\dot{q}_{v_d} = \mathbf{A}^+(\Lambda\tilde{\sigma} - \eta) \quad (8)$$

where \dot{q}_{v_d} is the desired velocity of the virtual manipulator and Λ is a positive definite $m \times m$ gain matrix. As \dot{q}_v is a virtual quantity, which can be set at will, $\dot{q}_{v_d} = \dot{q}_v$ holds. Hence, both values will be used interchangeably.

The stability of this controller can be easily proven, using the Lyapunov function

$$V = \frac{1}{2}\tilde{\sigma}^T\tilde{\sigma} \quad (9)$$

with the derivative

$$\dot{V} = \tilde{\sigma}^T\dot{\tilde{\sigma}} = -\tilde{\sigma}^T\dot{\sigma} = -\tilde{\sigma}^T(\mathbf{A}\dot{q}_v + \eta) \quad (10)$$

With the controller (8), (10) can be rewritten as

$$\dot{V} = -\tilde{\sigma}^T(\mathbf{A}\mathbf{A}^+(\Lambda\tilde{\sigma} - \eta) + \eta) = -\tilde{\sigma}^T\Lambda\tilde{\sigma}. \quad (11)$$

Being a quadratic form, \dot{V} is negative if Λ is a positive definite matrix, what is given by definition. Hence, with the controller (8) the task error converges to zero.

B. Joint Position / Cartesian Pose Regulation

Positioning controllers building on an IFC framework usually provide trajectories for q_v , without regarding the actual motion of q . How well q tracks q_v depends on the IFC implementation, especially on the stiffness \mathbf{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 q_v in comparison to 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.

On joint level, the trivial relation $\dot{q}_v = \mathbf{I}_n\dot{q}_v$ leads to the simple controller

$$\dot{q}_v = \Lambda_q\tilde{q}_v. \quad (12)$$

Using standard instantaneous inverse kinematic methods, a controller for the Cartesian pose x_v of an arbitrary frame on the manipulator can be derived. Using (1), the derivative of the task variable x_v is

$$\dot{x}_v = \mathbf{J}_v\dot{q}_v \quad (13)$$

with $\mathbf{J}_v = \mathbf{J}(q_v)$ as the short notation for the Jacobian of the virtual configuration q_v . Required coordinate transformations, depending on the used orientation representation, are assumed to be already incorporated in \mathbf{J}_v to simplify notation. The Cartesian pose controller is hence

$$\dot{q}_v = \mathbf{J}_v^+\Lambda_x\tilde{x}_v. \quad (14)$$

Actually, the term controller is misleading when regulating the virtual manipulator's position, since there is no feedback of the physical entities and q_v is set usually in an open loop way. However, complying with the general formulation (8) is helpful when combining different tasks as will be seen in section V.

C. Joint Torque / Wrench Regulation

As mentioned before, we are only aiming for regulation of the static force components and assuming the ideal relation

$$\tau = \bar{\tau}_c, \quad (15)$$

neglecting dynamic effects and imperfect IFC implementation. The time derivative of τ is hence

$$\dot{\tau} = \mathbf{K}(\dot{q}_v - \dot{q}). \quad (16)$$

Complying with (8) the according interaction torque controller has the form

$$\dot{q}_v = \mathbf{K}^{-1}(\Lambda_\tau\dot{\tau} + \mathbf{K}\dot{q}) \quad (17)$$

Since \mathbf{K} is a positive definite diagonal matrix, the Moore-Penrose inverse can be replaced by the regular inverse here.

For the static wrench controller, we use (3) to obtain the relation

$$\dot{h} = \frac{\delta h}{\delta t} = \frac{\delta \mathbf{J}^{T+}}{\delta t}\tau + \mathbf{J}^{T+}\dot{\tau}. \quad (18)$$

Using (15), (16) and (5), (18) is stated as

$$\dot{\mathbf{h}} = \frac{\delta \mathbf{J}^{T+}}{\delta t} \mathbf{K}(\mathbf{q}_v - \mathbf{q}) + \mathbf{J}^{T+} \mathbf{K}(\dot{\mathbf{q}}_v - \dot{\mathbf{q}}) \quad (19)$$

and rewritten to comply with (7)

$$\dot{\mathbf{h}} = \mathbf{J}^{T+} \mathbf{K} \dot{\mathbf{q}}_v - \mathbf{J}^{T+} \mathbf{K} \dot{\mathbf{q}} + \frac{\delta \mathbf{J}^{T+} \mathbf{K}}{\delta t} (\mathbf{q}_v - \mathbf{q}). \quad (20)$$

With $\mathbf{A}_h = \mathbf{J}^{T+} \mathbf{K}$, the resulting controller is

$$\dot{\mathbf{q}}_v = \mathbf{A}_h^+ (\mathbf{\Lambda}_h \tilde{\mathbf{h}} + \mathbf{A}_h \dot{\mathbf{q}} - \dot{\mathbf{A}}_h (\mathbf{q}_v - \mathbf{q})) \quad (21)$$

Note, that here the Jacobian of the actual configuration \mathbf{q} has to be used. Table I summarizes the four basic task types

- end effector pose
- joint position
- end effector wrench
- joint torques

V. HIERARCHICAL TASK PROGRAMMING

A. Tasks in Subspaces

Optionally, every task σ can be defined only in a certain subspace \mathbb{S}_σ of \mathbb{R}^m . This subspace is characterized by an orthonormal matrix \mathbf{S} , containing the basis vectors of the according subspace. \mathbf{S} has some similarity to the compliance selectivity matrix known from hybrid position/force control. While in hybrid position/force control an additional specification of the task frame is required, this information is already included in our \mathbf{S} . Also the subspaces of the different subtasks do not have to be orthogonal but can be defined in any way. In fact, the introduction of \mathbf{S} gives the application programmer more freedom in defining a task, as not relevant directions can be neglected and the additional degrees of freedom can be used to fulfill lower priority tasks.

A vector $\mathbf{a} \in \mathbb{R}^m$ can be projected to \mathbb{S}_σ with

$$\mathbf{a}_s = \mathbf{S}^T \mathbf{a}. \quad (22)$$

The equations from section IV can be modified for subtasks σ_s , defined in \mathbb{S}_σ . Basically only \mathbf{A} has to be projected to \mathbb{S}_σ with

$$\mathbf{A}_s = \mathbf{S}_\sigma^T \mathbf{A}. \quad (23)$$

From the mathematical point of view, this operation enlarges the task's nullspace, what makes the proposed method also interesting for nonredundant manipulators. Subscripts will be dropped in future formulations.

TABLE I
SPECIFICATIONS OF THE FOUR BASIC TASK TYPES

type	σ	\mathbf{A}	η	$\mathbf{\Lambda}$
ee pose	$\sigma_x = \mathbf{x}_v$	$\mathbf{A}_x = \mathbf{J}_v$	$\mathbf{0}$	$\mathbf{\Lambda}_x$
joint position	$\sigma_q = \mathbf{q}_v$	$\mathbf{A}_q = \mathbf{I}_n$	$\mathbf{0}$	$\mathbf{\Lambda}_q$
ee wrench	$\sigma_h = \mathbf{h}$	$\mathbf{A}_h = \mathbf{J}^{T+} \mathbf{K}$	$\dot{\mathbf{A}}_h (\mathbf{q}_v - \mathbf{q}) - \mathbf{A}_h \dot{\mathbf{q}}$	$\mathbf{\Lambda}_h$
joint torque	$\sigma_\tau = \boldsymbol{\tau}$	$\mathbf{A}_\tau = \mathbf{K}$	$-\mathbf{K} \dot{\mathbf{q}}$	$\mathbf{\Lambda}_\tau$

B. Enforcing a Task Hierarchy

Using the general task formalism from section IV, we can now define an arbitrary large set of subtasks $[\sigma_1 \dots \sigma_k]$, with k as the number of subtasks, sorted by descending priority and optionally expressed in a certain subspace as described in section V-A. Every subtask σ_i is assigned a matrix \mathbf{A}_i , stating the linear mapping

$$\boldsymbol{\lambda}_i = \mathbf{A}_i \dot{\mathbf{q}}_v, \quad (24)$$

where i is the subtask index. Equation (24) is a reformulation of (8) with $\boldsymbol{\lambda}_i = \mathbf{\Lambda}_i \tilde{\boldsymbol{\sigma}}_i - \boldsymbol{\eta}_i$ for better readability.

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

$$N(\mathbf{X}) = \ker(\mathbf{X})^T \ker(\mathbf{X}) \quad (25)$$

is defined, which projects a vector on the nullspace of \mathbf{X} .

With this, the k subtasks can be combined recursively with $\mathbf{0}_n$ as the $n \times n$ zero matrix:

$$\begin{aligned} \dot{\mathbf{q}}_{v0} &= \mathbf{0} \\ \mathbf{A}_0 &= \mathbf{0}_n \\ \dot{\mathbf{q}}_{vi} &= N([\mathbf{A}_0 \dots \mathbf{A}_{i-1}]^T) \mathbf{A}_i^+ (\boldsymbol{\lambda}_i - \mathbf{A}_i \dot{\mathbf{q}}_{vi-1}) \\ \dot{\mathbf{q}}_v &= \sum_{i=1}^k \dot{\mathbf{q}}_{vi}. \end{aligned} \quad (26)$$

Every subtask is projected in the nullspace of all the higher priority tasks by applying $N(\mathbf{X})$ on the augmented matrix $[\mathbf{A}_0 \dots \mathbf{A}_{i-1}]^T$, containing the linear maps of all the higher priority tasks. This way of task combination guarantees, that the task hierarchy is not violated, what is elaborately discussed in [17]. The term $\mathbf{A}_i \dot{\mathbf{q}}_{vi-1}$ is the compensation for the effects of the higher priority tasks on $\boldsymbol{\lambda}_i$.

It has to be mentioned, that (26) will not produce the optimal solution in terms of task execution, but the optimal solution in terms of $\dot{\mathbf{q}}_{vi}$, since the projection (25) is orthogonal in joint-space but not in the task space. The optimal task space solution would be obtained by

$$\dot{\mathbf{q}}_{vi} = (\mathbf{A}_i N([\mathbf{A}_0 \dots \mathbf{A}_{i-1}]^T))^+ (\boldsymbol{\lambda}_i - \mathbf{A}_i \dot{\mathbf{q}}_{vi-1}), \quad (27)$$

where the nullspace mapping is incorporated before the computation of the pseudoinverse. However, this method introduces additional possibilities for singularities, namely if some \mathbf{A} matrix drops rank, becoming square and $N(\bullet)$ hence returns a singular matrix. This can for example happen when handling joint limits as described later in section V-C. In fact, when applying the pseudoinverse, it is not the singularity itself but the bad conditioning near the singularity which is problematic, as the inverse ends up with large entries. This adjacency to singularities can be due to numeric noise or if the rank drop of \mathbf{A} is not instantaneous. To avoid those problems for the present work, we settle for the solution obtained with (26) and postpone proper treatment of singularities to future work, respectively refer to some recent publications on this, e.g. [18].

C. Incorporating Joint Limits

Referring to [10], we set the according row in all the linear maps $\mathbf{A}_{0\dots k}$ to $\mathbf{0}$, if a joint hits its limit. This basically removes the respective joint from the computations in (26) and treats the joint as a static connection.

Such clamping of a joint leads to an instantaneous rank drop of the $\mathbf{A}_{0\dots k}$ matrices, what gets propagated to the controller and leads to discontinuous solutions for $\dot{\mathbf{q}}_v$. 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 [19], [20].

VI. EXPERIMENTAL RESULTS

A. Implementation Details and Hardware

The experiments have been carried out on our KUKA LWR-IV lightweight arm. The manipulator was running a joint space impedance controller, which details can be found in [21]. The experimental setups are depicted in Fig. 1. The rate of the discrete controller was $r = 500Hz$ and the default stiffness was $\mathbf{K} = 400\mathbf{I}_7Nm/rad$. All the gains 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. We limit ourselves to the basic types of subtasks, whose linear maps are obtained from table I, by using the virtual and actual end effector Jacobians. A certain subtask can now be defined by specifying the task type, providing the desired task variable $\sigma_d \in \mathbb{S}_\sigma$ with according gain matrix $\mathbf{\Lambda}$ and an appropriate subspace matrix \mathbf{S} .

The presented approach was also verified in a simplified simulation. However, we leave the results out in this paper as they do not add any insights and due to limited space.

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, starting at \mathbf{x}_{init} with radius R and frequency f in the y - z -plane, while keeping the constant initial orientation \mathbf{o}_{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 or alternatively minimizing the joint torques.

TABLE II
SET OF SUBTASKS FOR SURFACE TRACKING

prio	type	σ_d	$\mathbf{\Lambda}$	\mathbf{S}
1	ee wrench	$8N$	2.5	$[1\ 0\ 0\ 0\ 0\ 0]^T$
2	ee pose	$\begin{bmatrix} y_{init} + R \cos(2f\pi t) - R \\ z_{init} + R \sin(2f\pi t) \\ o_{yinit} \\ o_{zinit} \end{bmatrix}$	$10\mathbf{I}_4$	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
3	joint position joint torque	$\mathbf{0}$ $\mathbf{0}$	$0.6\mathbf{I}_7$ \mathbf{I}_7	\mathbf{I}_7 \mathbf{I}_7

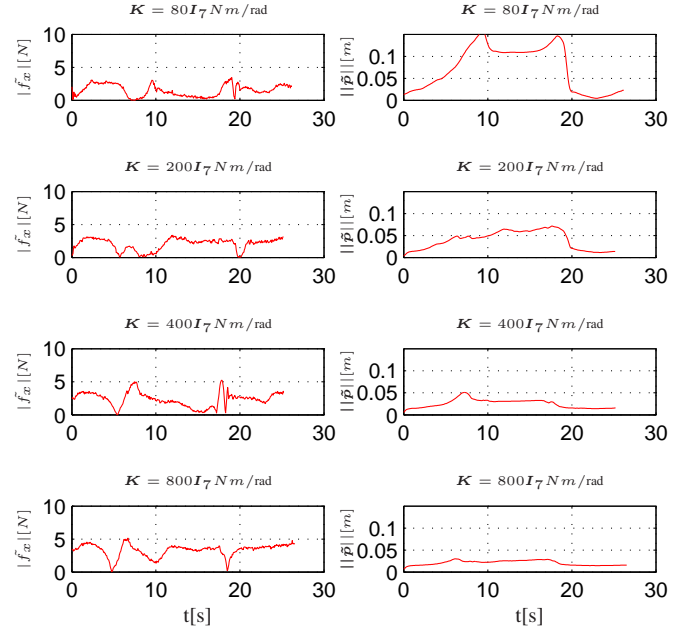


Fig. 3. Force and positioning error for varying virtual joint stiffness during a surface tracking task.

The first trial was conducted on a curved surface with unknown flexibility (see Fig. 1(a-1)). The impacts of execution speed and stiffness \mathbf{K} of the IFC are demonstrated with this example. The task errors for the force and positioning subtasks are plotted in Fig. 3 for alternating stiffness and in Fig. 4 for alternating execution speed. As stated in section IV-B, the quality of position tracking increases with higher values for \mathbf{K} and decreases for faster execution speeds. The force tracking error mainly comes from the fact, that the assumed identity (15) does not take modeling 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 (15). Incrementing the stiffness also leads to slightly worse force tracking, due to higher sensibility of the static interaction torques to $\dot{\mathbf{q}}_v$. These are clear downsides when using an IFC scheme.

The second run 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, what can be seen also in the video accompanying this paper.

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

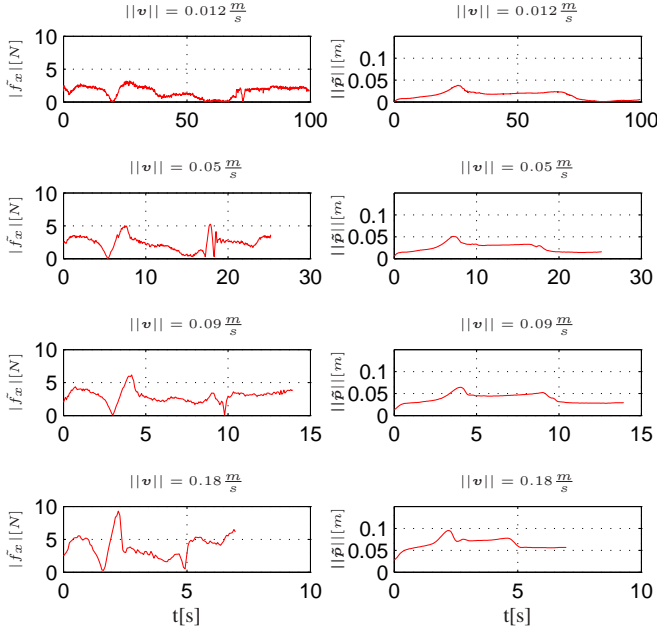


Fig. 4. Force and positioning error for varying execution speed during a surface tracking task.

radii 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 S . 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, resulting in smoother motion. The third subtask is to keep the joints away from their limits. Table IV shows the task parameters. With R_{ee} being the end effector orientation matrix,

$$T_{IE} = \begin{bmatrix} I_3 & \mathbf{0} \\ \mathbf{0} & R_{ee} \end{bmatrix}$$

transforms the rotational part of S in end effector coordinates.

Here, σ_d is defined in the global frame and then projected to \mathbb{S}_σ by premultiplication with S_2^T , denoting the transposed of the subspace matrix for the second subtask.

TABLE III
SET OF SUBTASKS FOR TABLE WIPING

prio	type	σ_d	Λ	S
1	ee wrench	$8N$	2.5	$[1 \ 0 \ 0 \ 0 \ 0 \ 0]^T$
2	ee pose	$\begin{bmatrix} y_{\text{init}} + R_1 \cos(2f_1\pi t) - R_1 \\ z_{\text{init}} + R_2 \sin(2f_2\pi t) \\ \sigma_{\text{init}} \end{bmatrix}$	$10I_5$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ I_5 \end{bmatrix}$
3	joint torque	$\mathbf{0}$	I_7	I_7

TABLE IV
SET OF SUBTASKS FOR CIRCLE DRAWING

prio	type	σ_d	Λ	S
1	ee wrench	$3N$	2.5	$[1 \ 0 \ 0 \ 0 \ 0 \ 0]^T$
2	ee pose	$S_2^T \begin{bmatrix} 0 \\ y_{\text{init}} + R \cos(2f\pi t) - R \\ z_{\text{init}} + R \sin(2f\pi t) \\ \sigma_{\text{init}} \end{bmatrix}$	$10I_4$	$T_{IE} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
3	joint position	$\mathbf{0}$	$0.6I_7$	I_7

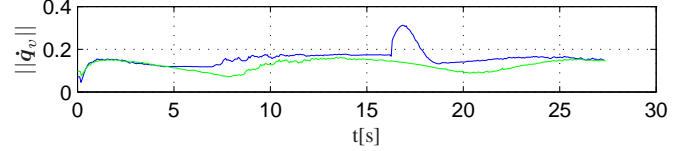


Fig. 5. Norm of joint velocities for different task specifications for circle drawing. Blue: keeping constant orientation. Green: Rotation around end effector y -axis permitted.

The positioning subtask was also specified with S_2 as in the table wiping task (III) and the trajectory is defined in a way, that joint limits are hit if the task is executed with this specification. In Fig. 5 one can see the resulting discontinuity in $\|\dot{q}_v\|$. When relaxing the task constraints, the additional degree of freedom is used to avoid joint limits, resulting in a smoother trajectory.

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 video). See table V for task description.

e) Operating Unknown Constrained Mechanisms (Fig. 1(e)): By using a simple constrained estimator [22], which gives us the three-dimensional direction vector of possible translational motion d , a simple controller for operating constrained mechanisms can be designed. A constant force is assigned along d , while the end effector pose 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.

TABLE V
SET OF SUBTASKS FOR CUP HOLDING

prio	type	σ_d	Λ	S
1	ee pose	σ_{init}	$10I_3$	$\begin{bmatrix} \mathbf{0}_3 \\ I_3 \end{bmatrix}$
2	joint torque	$\mathbf{0}$	I_7	I_7

TABLE VI
SET OF SUBTASKS FOR CONSTRAINED MANIPULATION

prio	type	σ_d	Λ	S	
1	ee wrench	$10Nd$	$10I_3$	I_3	O_3
2	joint torque	$0N$	1	$[0\ 0\ 0\ 0\ 0\ 1\ 0]^T$	
3	ee pose	$S_3^T x_{init}$	$10I_5$	T_{IE}	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
4	joint torque	0	I_7	I_7	

In addition, our method makes it easy to incorporate joint specific actions at any layer, e.g. assuming we have a weak joint (e.g. joint 6) and want to minimize the torques on this joint as good as possible without affecting the primary task, we simply insert this as a secondary joint torque task.

VII. CONCLUSION AND FUTURE WORK

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 nonredundant robots. The usage of the well-established IFC scheme 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.

In future work, we plan to deal with some details of the approach, in particular handling singularities properly, avoiding discontinuities in set point generation (e.g. due to joint limits), incorporating dynamic limits and investigating dynamic selection of the subspace matrix S to extend the framework to unilateral constraints.

ACKNOWLEDGMENT

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

REFERENCES

[1] N. Hogan, "Impedance control: An approach to manipulation," in *American Control Conference, 1984*. IEEE, 1984, pp. 304–313.
[2] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, p. 3, 1987.

[3] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, 1987.
[4] M. Raibert and J. Craig, "Hybrid position/force control of manipulators," *ASME Journal of Dynamic Systems, Measurement and Control*, pp. 126–133, 1981.
[5] M. Vukobratovic and a. Tuneski, "Contact control concepts in manipulation robotics-an overview," *IEEE Transactions on Industrial Electronics*, vol. 41, no. 1, pp. 12–24, 1994.
[6] T. Lefebvre, J. Xiao, H. Bruyninckx, and G. de Gerssem, "Active compliant motion: a survey," *Advanced Robotics*, vol. 19, no. 5, pp. 479–499, Jun. 2005.
[7] 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.
[8] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-Loop Inverse Kinematics Schemes for Constrained Redundant Manipulators with Task Space Augmentation and Task Priority Strategy," *The International Journal of Robotics Research*, vol. 10, no. 4, pp. 410–425, Aug. 1991.
[9] 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.
[10] 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.
[11] L. Sentis and O. Khatib, "Task-oriented control of humanoid robots through prioritization," in *IEEE/RSJ International Conference on Humanoid Robots*, 2004, pp. 1–16.
[12] 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.
[13] 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.
[14] 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.
[15] P. Tomei, "A simple PD controller for robots with elastic joints," *Automatic Control, IEEE Transactions on*, vol. 36, no. 10, pp. 1208–1213, 1991.
[16] N. Hogan, "On the stability of manipulators performing contact tasks," *IEEE Journal on Robotics and Automation*, vol. 4, no. 6, pp. 677–686, 1988.
[17] 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.
[18] 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.
[19] O. Kanoun, F. Lamiroux, 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.
[20] 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.
[21] 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.
[22] 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.