# Hierarchical Force and Positioning Task Specification for Indirect Force-Controlled Robots

Ewald Lutscher, Emmanuel C. Dean-León,
and Gordon Cheng, *Fellow, IEEE,*

***Abstract*—Indirect force control (*IFC*) architectures are a common approach for dealing with unknown environments. What all IFC schemes have in common, is that the relation between the set-point and the actual configuration of the robot is determined by a mechanical relationship (e.g. a mass-spring-damper system). In this work, we propose a set-point generation method for IFC schemes, enabling intuitive specification of mixed force and positioning tasks on joint and Cartesian level. In addition, the formulation of equality and inequality tasks is supported and a passivity based stability proof is formulated using the concept of virtual energy storage.**

**The resulting task programming interface is demonstrated on a 7 degree of freedom robot, running a joint space impedance controller. One sample task demonstrates the application of the developed approach and highlights the basic features.**

## I. Motivation

Compliant control involving force and positioning tasks has been investigated elaborately in the last decades. A popular approach to realize compliance is provided by indirect force controllers, where the motion and interaction forces of the physical robot are indirectly controlled by moving a virtual robot, which is coupled to the physical robot via a virtual mechanical relationship. Fig. 1 depicts the basic idea. The most popular variation of an IFC is the well known impedance control paradigm, introduced by Hogan in his seminal paper [1], but also simpler variants, like stiffness control or an ordinary PD-controller with compensation of the gravitational torques and sufficiently low proportional gain. This control scheme has also nice stability properties which are basically independent from the environmental dynamics. Furthermore, series elastic actuator (*SEA*) robots, which are equipped with physical springs in their joints, can also be considered as IFC's.

The set-point selection for the virtual robot is also referred to as virtual trajectory generation and while there is an extensive amount of work, having the purpose of improving the accuracy of the virtual mechanical relationship or proposing different extensions or variations for IFC's (for impedance control in particular), the literature covering pure virtual trajectory generation is very sparse.

The present work is meant to fill the gap between the low level control design and high level application programming by introducing an additional control layer, which we call set-point-generator (*SPG*). The purpose of the SPG is to provide set-points for the virtual manipulator according to the specified tasks, using only information on the current joint state. These tasks are defined by the application programmer via a task specification interface, where a set of hierarchically ordered force and positioning subtasks on joint and Cartesian level is determined. To our best knowledge, there is no work treating virtual set-point generation for joint level IFC's in order to achieve mixed force and positioning tasks without additional sensory feedback, like force or vision.

In [2] we proposed an approach to compose mixed force and positioning interaction tasks for a robotic manipulator under joint space indirect force control using recursive null-space mapping. In [3] we extended the formalism to support inequality tasks by applying a sequence of quadratic programs (*QP*). In the present work we offer a more general formulation with a formal stability proof and additional experimental validation.

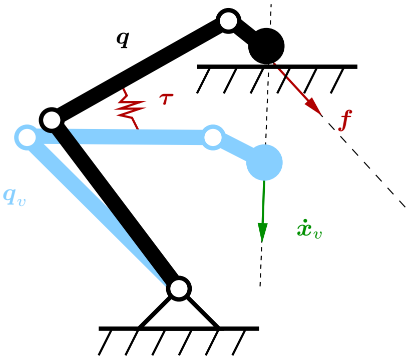

Fig. 1. Motion and interaction forces of the physical manipulator (black) are controlled indirectly by generating set-points for the virtual manipulator (blue). For joint space compliance, the direction of applied forces is in general not aligned with the Cartesian position deviation.

The remainder of the paper is structured as follows. Section II provides an overview on the fundamental ideas and approaches on which our work is based and summarizes the main contributions. In Section III, a simplified method of realizing force and positioning tasks for indirect force controlled robots is introduced, which is used in Section IV to implement a hierarchically structured set of equality and inequality tasks. A passivity based stability proof can be found in Section V. The experimental results are presented in Section VI and Section VII concludes the paper.

## II. State of the Art and Contribution

### A. Indirect Force Control

The major contributions regarding IFC have the intention to improve the performance of the controller itself or to introduce different variants and extensions ([4], [5], [6]). Several investigations have also been made targeting the construction of compliances, optimized for specific tasks [7], [8].

Considering applications, conventional trajectory planning approaches are often applied and the IFC is used to compensate for contact uncertainty and unexpected collisions, e.g. [9], [10], [11]. In these works, the occurring interaction forces are not considered explicitly. The only works explicitly dealing with virtual trajectory generation aim at pure force tracking or are part of a customized approach to a specific problem. In [12], [13], [14], [15], [16] and [17], force tracking in impedance control for industrial applications is treated, considering a Cartesian impedance controller. The focus of these works lies on setting proper virtual trajectories to adapt to the unknown environmental stiffness.

### B. Task Specification

The first concept of assembling a mixed force and positioning main task from different subtasks was derived in conjunction with the task frame formalism [18] and further developed in [19], [20], [21]. A force or positioning subtask is assigned to every direction of the specified task frame and a hybrid force/position controller is used to track the desired trajectories or set-points simultaneously. Tasks are denoted as artificial constraints. A survey on such constraint-based task specification formalisms can be looked up in [22]. As the hybrid force position control scheme is used as the underlying low level control, the task geometry has to be known in advance or estimated online as in [23], [24].

E. Lutscher, E. Dean-León and G. Cheng are with the Institute for Cognitive Systems, Technische Universität München, Germany e-mail: (see http://www.ics.ei.tum.de).

Manuscript received July 21, 2017; revised September 16, 2017.

### C. Multi-Task Programming with Inequality Constraints

Due to computational limitations, only equality tasks are considered in the classical approaches for redundancy resolution. There exist a vast number of resolving joint level inequality constraints, e.g. [25], [26], or handling specific inequality constraints, like collision/singularity avoidance, which has been treated in the past via the gradient projection method [27]. Flacco et. al. introduced an algorithm to incorporate joint angle, velocity and acceleration limits and exploit them as well as possible to achieve a Cartesian task by scaling it appropriately [28]. A unified approach is presented in [29], where general inequality tasks are treated on every priority level in a stack-of-tasks framework. In recent contributions, QP methods are used to find an optimal solution for the inverse kinematic problem with a given task hierarchy ([30], [31]). The QP formulation is also exploited on the dynamics level to resolve the hyper redundancy of humanoid robots (e.g. [32], [33]). To our best knowledge there is no application in the context of set point generation for indirect force control.

### D. Contribution

The goal of the present work is to derive a task programming scheme for a general joint level IFC setup. The intention is not to present another IFC variation or force controller, instead an additional control layer between the IFC and the task level is introduced, which generates the appropriate set-points for an IFC, depending on the specified tasks. In this work, a simple underlying IFC scheme is assumed, which emulates a spring-damper system while canceling the torques due to gravity $\boldsymbol{\tau}_g$, realizing the relation

$$\boldsymbol{\tau} = \boldsymbol{K}(\boldsymbol{q}_v - \boldsymbol{q}) - \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{\tau}_g \tag{1}$$

between the robot torques $\boldsymbol{\tau}$ and deviation of virtual and actual joint configuration $\boldsymbol{q}_v$ and $\boldsymbol{q}$. $\boldsymbol{K}$ and $\boldsymbol{D}$ are constant, diagonal, positive definite matrices parameterizing the virtual spring and damper in each joint. Nevertheless, more robust or advanced IFC schemes can be used with this approach.

This additional layer brings the following advantages:

- Force and positioning equality and inequality tasks on joint and Cartesian level are captured with one unifying formalism.
- Intuitive, yet powerful task programming without requiring detailed information of the underlying structure, while the inherent compliance of the IFC is preserved.
- By treating the IFC-robot entity as a general abstraction layer (see Fig. 2) and assuming no additional sensor information, the developed approach is basically hardware independent as long as the robot features an IFC interface, including SEA robots.
- As neither modification, nor detailed information of the underlying IFC is required, the proposed formalism is a suitable approach for dealing with so called closed architectures, where only limited access (i.e. an IFC interface) to the robot is granted.

## III. Generalized Force and Position Regulation in IFC Context

### A. Robotic Foundations

The configuration of a manipulator with $n$ degrees of freedom is uniquely defined by a set of $n$ generalized coordinates $\boldsymbol{q}$ which are for revolute joints usually the joint angles. The Cartesian pose $\boldsymbol{x}$ of any desired frame, attached to the robot can be computed from $\boldsymbol{q}$ by applying the forward kinematics map

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{o} \end{pmatrix}, \tag{2}$$

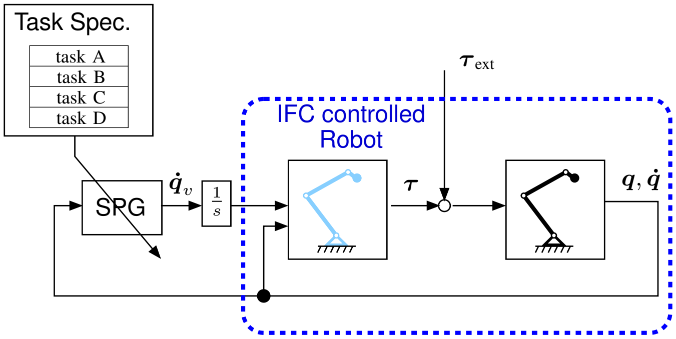

Fig. 2. Overview of the full IFC system. The IFC-robot block (blue-dashed box) is considered a black box, which returns the robot state and takes set-point commands for the virtual robot as input. These set-points are transformed to torque commands, usually depending on a virtual mechanical relationship between the virtual and the physical robot. The set-points are provided by the SPG, depending on the specified subtasks in the task specification block and the robot state. The external torques $\boldsymbol{\tau}_{\text{ext}}$ are due to contact with the environment or humans.

where $\boldsymbol{p} \in \mathbb{R}^3$ denotes the position and $\boldsymbol{o}$ the orientation of the frame. The unit and dimension of $\boldsymbol{o}$ depend on the chosen orientation representation.

To control the motion of the robot in a reactive way, the derivative of (2)

$$\dot{\boldsymbol{x}} = \begin{pmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{pmatrix} = \frac{\partial \boldsymbol{x}(\boldsymbol{q})}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{3}$$

with $\boldsymbol{\omega}$ as the angular velocity of the end effector frame and $\boldsymbol{J}$ as the manipulator's base Jacobian can be used.

Another useful property of $\boldsymbol{J}$ is the static wrench transmission which relates the three-dimensional end effector forces $\boldsymbol{f}$ and moments $\boldsymbol{m}$, to joint torques $\boldsymbol{\tau} = \boldsymbol{J}^T\boldsymbol{h}$, where $\boldsymbol{h} = (\boldsymbol{f}^T \quad \boldsymbol{m}^T)^T$ denotes the end effector wrench. If $\tau$ is due to an external wrench at the end effector, the Moore-Penrose pseudoinverse [34] of the transposed base Jacobian can be used to compute it from the torques:

$$\boldsymbol{h} = \boldsymbol{J}^{T+}\boldsymbol{\tau}. \tag{4}$$

A more general formulation, considering dynamic components in $\boldsymbol{\tau}$, was proposed by Khatib in [35]. With the mass matrix $\boldsymbol{M}$, this so called dynamically consistent pseudoinverse $(\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T)^{-1}\boldsymbol{J}\boldsymbol{M}^{-1}$, should be used instead of $\boldsymbol{J}^+$ in (4) and all related equations to improve the external wrench estimation. Still, in the present work the Moore-Penrose pseudoinverse was used in a quasistatic approximation in order to avoid using the mass matrix, requiring additional knowledge of the robotic hardware. This is done according to the assumption, that the major components of $\boldsymbol{\tau}$ are related to the external wrench, while dynamics can be neglected.

### B. Task Variables

Consider a pure IFC interface like (1). Neither position $\boldsymbol{q}$ nor the full interaction torque $\boldsymbol{\tau}$ can be controlled directly, hence we declare the following replacement task variables:

- $\boldsymbol{q}_v$ – virtual joint space position
- $\boldsymbol{x}_v = \boldsymbol{g}(\boldsymbol{q}_v)$ – virtual Cartesian pose (i.e. end effector)
- $\boldsymbol{\tau}_s = \boldsymbol{K}(\boldsymbol{q}_v - \boldsymbol{q})$ – static interaction joint torques
- $\boldsymbol{h}_s = \boldsymbol{J}^{T+}\boldsymbol{\tau}_s = \boldsymbol{J}^{T+}\boldsymbol{K}(\boldsymbol{q}_v - \boldsymbol{q})$ – static interaction wrench

These variables have a close relation to the actual variables of interest $(\boldsymbol{q}, \boldsymbol{x}, \boldsymbol{\tau}, \boldsymbol{h})$ but with a direct relation to $\boldsymbol{q}_v$.

### C. Generalized Task Control

To execute multiple tasks simultaneously we need a generalized task control scheme. If the robot is redundant with respect to the task, local inversion of the differential relation between task variable and actuating variable ($\boldsymbol{q}_v$) is the most common approach. The main advantages are the simplicity of task design, the possibility of a sensor-based execution, and the easy integration of multiple tasks with priorities [36]. This differential relation is captured by the so called task Jacobian.

*1) Deriving the Task Jacobians:* In the following we will derive the task Jacobians for the four task variables defined in the previous Section.

*Joint Position:* For the joint position $\boldsymbol{q}_v$ the task Jacobian is simply the $n \times n$ identity matrix $\boldsymbol{I}_n$:

$$\dot{\boldsymbol{q}}_v = \boldsymbol{I}_n \dot{\boldsymbol{q}}_v. \tag{5}$$

*Cartesian Pose:* For the Cartesian pose, the instantaneous kinematics relation (3) is used:

$$\dot{\boldsymbol{x}}_v = \boldsymbol{J}_v \dot{\boldsymbol{q}}_v \tag{6}$$

with $\boldsymbol{J}_v = \boldsymbol{J}(\boldsymbol{q}_v)$ denoting the Jacobian of the virtual manipulator.

*Joint Torque:* Taking the time derivative of $\boldsymbol{\tau}_s$ and reformulating gives us

$$\dot{\boldsymbol{\tau}}_s + \boldsymbol{K}\dot{\boldsymbol{q}} = \boldsymbol{K}\dot{\boldsymbol{q}}_v.$$

It can be noted that an additional dependency on the joint velocity of the physical robot $\dot{\boldsymbol{q}}$ prevents a straight forward linear formulation as in (5) and (6). Therefore, a new differential torque task variable $\boldsymbol{\beta}_\tau$ is defined, which compensates for the effects of $\dot{\boldsymbol{q}}$:

$$\boldsymbol{\beta}_\tau = \dot{\boldsymbol{\tau}}_s + \boldsymbol{K}\dot{\boldsymbol{q}}$$
$$\boldsymbol{\beta}_\tau = \boldsymbol{K}\dot{\boldsymbol{q}}_v.$$

*Wrench:* With the time derivative of (4) we get

$$\dot{\boldsymbol{h}}_s = \dot{\boldsymbol{J}}^{T+}\boldsymbol{\tau}_s + \boldsymbol{J}^{T+}\dot{\boldsymbol{\tau}}_s.$$

By assuming only moderate joint velocities, we neglect the term with the derivative of the Jacobian, resulting in

$$\dot{\boldsymbol{h}}_s = \boldsymbol{J}^{T+}\dot{\boldsymbol{\tau}}_s = \boldsymbol{J}^{T+}\boldsymbol{K}(\dot{\boldsymbol{q}}_v - \dot{\boldsymbol{q}}).$$

Again, a compensated differential wrench task variable $\boldsymbol{\beta}_f$ can be defined and a differential relation stated:

$$\boldsymbol{\beta}_f = \dot{\boldsymbol{h}}_s + \boldsymbol{J}^{T+}\boldsymbol{K}\dot{\boldsymbol{q}} \tag{7}$$
$$\boldsymbol{\beta}_f = \boldsymbol{J}^{T+}\boldsymbol{K}\dot{\boldsymbol{q}}_v \tag{8}$$

This relation is only valid for Jacobians, which are sufficiently far away from singularities, due to the inversion in (8).

To unify the four task variables, we define the general $m$-dimensional task variable $\boldsymbol{\sigma}$, so that the general compensated differential task variable $\boldsymbol{\beta}$ and general $m \times n$ task Jacobian can be defined as

$$\boldsymbol{\beta} = \dot{\boldsymbol{\sigma}} + \boldsymbol{\gamma}(\dot{\boldsymbol{q}})$$
$$\boldsymbol{\beta} = \boldsymbol{A}\dot{\boldsymbol{q}}_v,$$

where $\boldsymbol{\gamma}(\dot{\boldsymbol{q}})$ is the respective compensation for $\dot{\boldsymbol{q}}$.

*2) Trimmed Task Space:* Objectives do not have to be necessarily defined in the full task space. Often, only a particular subspace is relevant, and the released degrees of freedom can be used to achieve other tasks. The base of this subspace can be denoted by a set of orthonormal vectors, which form the columns of a matrix $\boldsymbol{S}$. With $\boldsymbol{\sigma}$ being defined in the subspace frame, the basic task Jacobians have to be transformed to the subspace coordinates via $\boldsymbol{A} := \boldsymbol{S}^T\boldsymbol{A}$.

$\boldsymbol{S}$ serves for two purposes here. First, it can be used as a selection matrix to select relevant directions of the full task space, similar to the selection matrix in hybrid force and position control [37]. Second, a coordinate transformation can be incorporated in $\boldsymbol{S}$ to formulate the task (partially) in a convenient coordinate system. It follows that $\boldsymbol{S}$ is either $6 \times m$ or $n \times m$, depending on whether $\boldsymbol{\sigma}$ is defined in Cartesian or in joint space.

*3) Generalized Controller:* The simplest way to bring the task variable $\boldsymbol{\sigma}$ continuously to a desired state $\boldsymbol{\sigma}_d$ is by using the task error $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}$ to formulate a general proportional task level controller

$$\dot{\boldsymbol{\sigma}}_{cmd} = \boldsymbol{\Lambda}\tilde{\boldsymbol{\sigma}} + \dot{\boldsymbol{\sigma}}_{ff}, \tag{9}$$

where $\dot{\boldsymbol{\sigma}}_{ff}$ denotes a feedforward term and $\boldsymbol{\Lambda}$ a diagonal gain matrix. The main purpose of $\dot{\boldsymbol{\sigma}}_{ff}$ is to allow a direct specification of the differential task variable. The desired compensated task variable is then

$$\boldsymbol{\beta}_d = \dot{\boldsymbol{\sigma}}_{cmd} + \boldsymbol{\gamma}(\dot{\boldsymbol{q}})$$

## IV. Simultaneous Position and Force Control in IFC

With the generalized task control presented in the previous section we can apply some of the most common redundancy resolution techniques to combine different tasks.

In [2] it was shown how equality tasks could be realized in an IFC context by applying hierarchical null-space mapping and in [3] the approach was extended to inequality tasks via a QP based formulation. As the stability proof presented in this paper requires limitation of $\dot{\boldsymbol{q}}_v$, the more general QP approach is briefly recapitulated.

Instead of having one desired value for the task variable $\boldsymbol{\sigma}_d$, we specify lower and upper bounds ($\boldsymbol{\sigma}_m$ and $\boldsymbol{\sigma}_M$) as a desired range for $\boldsymbol{\sigma}$, what can be defined as an inequality task or constraint $\boldsymbol{\sigma}_m \le \boldsymbol{\sigma} \le \boldsymbol{\sigma}_M$. An equality task $\boldsymbol{\sigma}_d$ can be specified by setting $\boldsymbol{\sigma}_m = \boldsymbol{\sigma}_M = \boldsymbol{\sigma}_d$.

The global limits are transformed to local limits on the differential task variable $\dot{\boldsymbol{\sigma}}_{cmd_m} \le \dot{\boldsymbol{\sigma}} \le \dot{\boldsymbol{\sigma}}_{cmd_M}$, where

$$\dot{\boldsymbol{\sigma}}_{cmd_m} = \boldsymbol{\Lambda}(\boldsymbol{\sigma}_m - \boldsymbol{\sigma}) + \dot{\boldsymbol{\sigma}}_{m_{ff}}$$
$$\dot{\boldsymbol{\sigma}}_{cmd_M} = \boldsymbol{\Lambda}(\boldsymbol{\sigma}_M - \boldsymbol{\sigma}) + \dot{\boldsymbol{\sigma}}_{M_{ff}}.$$

As in (9), a feedforward term $\dot{\boldsymbol{\sigma}}_{m_{ff}}$, respectively $\dot{\boldsymbol{\sigma}}_{M_{ff}}$ can be specified for both bounds. The according limits on $\boldsymbol{\beta}$ are

$$\boldsymbol{\beta}_m = \dot{\boldsymbol{\sigma}}_{cmd_m} + \boldsymbol{\gamma}(\dot{\boldsymbol{q}})$$
$$\boldsymbol{\beta}_M = \dot{\boldsymbol{\sigma}}_{cmd_M} + \boldsymbol{\gamma}(\dot{\boldsymbol{q}}).$$

The main advantage compared to other common approaches, like artificial potential fields [38], is that a clear inequality constraint is specified, hence no switching or priority shifting is required. Also the limit can be reached in finite time, hence the full range of possible motions is exploited [30].

The set of subtasks can be formulated as a sequence of QPs, where every problem is solved in an optimal way, without altering the quality of the solution for the previous tasks. Starting with the highest priority level, for every subtask the following QP problem is stated

$$\text{min.} \quad \frac{1}{2}\boldsymbol{s}^T\boldsymbol{s} + \frac{1}{2}\rho\dot{\boldsymbol{q}}_v^T\dot{\boldsymbol{q}}_v \tag{10}$$
$$\text{s.t.} \quad \boldsymbol{\beta}_m \le \boldsymbol{A}\dot{\boldsymbol{q}}_v - \boldsymbol{s} \le \boldsymbol{\beta}_M \tag{11}$$
$$\bar{\boldsymbol{\beta}}_m + \bar{\boldsymbol{s}}^* \le \bar{\boldsymbol{A}}\dot{\boldsymbol{q}}_v \le \bar{\boldsymbol{\beta}}_M + \bar{\boldsymbol{s}}^* \tag{12}$$
$$\dot{\boldsymbol{q}}_{v_m} \le \dot{\boldsymbol{q}}_v \le \dot{\boldsymbol{q}}_{v_M}, \tag{13}$$

where $\bar{\bullet}$ indicates the augmented vector or matrix of all the higher priority tasks. The cost function (10) contains a vector of slack variables $\boldsymbol{s}$, which is balanced against the virtual joint velocities with the regularization parameter $\rho \in \mathbb{R}^+$, which is important for the numerical stability of the process [39].

Minimizing a slack variable instead of the task error itself, makes the current task inequality (11) a soft constraint, allowing violation of the task velocity bounds $\boldsymbol{\beta}_m$ and $\boldsymbol{\beta}_M$ in case the task is unfeasible. The second inequality constraint (12) is a hard constraint, which makes sure that the higher priority task velocity bounds, accumulated in the augmented vectors $\bar{\boldsymbol{\beta}}_m$ and $\bar{\boldsymbol{\beta}}_M$, are not violated more than the previously determined minimal slack variables $\bar{\boldsymbol{s}}^*$. Inequality (13) finally constraints the virtual joint velocity to stay within certain bounds, which are shaped to obey joint position, velocity and acceleration limits ($\boldsymbol{q}_m, \boldsymbol{q}_M, \boldsymbol{v}_M$ and $\boldsymbol{a}_M$), using finite differences and the current commanded joint velocity $\dot{\boldsymbol{q}}_v$:

$$\dot{\boldsymbol{q}}_{v_m} = \max\{\tfrac{\boldsymbol{q}_m-\boldsymbol{q}_v}{\Delta T},\ -\boldsymbol{v}_M,\ (\boldsymbol{a}_M+\dot{\boldsymbol{q}}_v)\Delta T,\ -\sqrt{2\boldsymbol{a}_M(\boldsymbol{q}_v-\boldsymbol{q}_m)}\}$$

$$\dot{\boldsymbol{q}}_{v_M} = \min\{\tfrac{\boldsymbol{q}_M-\boldsymbol{q}_v}{\Delta T},\ \boldsymbol{v}_M,\ (-\boldsymbol{a}_M+\dot{\boldsymbol{q}}_v)\Delta T,\ \sqrt{2\boldsymbol{a}_M(\boldsymbol{q}_M-\boldsymbol{q}_v)}\},$$

assuming a constant sample time $\Delta T$.

It has to be noted, that the dynamically consistent pseudoinverse should be used here instead of $\boldsymbol{J}^+$ to cancel potential null-space torques, leading to uncontrolled motions of the robot. Alternatively, force tasks should be placed in the lowest priority level (without leaving any degrees of freedom for potential null-space torques). However, once the desired wrench is reached, the resulting differential commands are so small, that the uncontrolled null-space motion can be neglected.

## V. Stability Analysis

While it was shown in multiple works that the interconnection of an IFC block and a robot is stable (e.g. [40], [41]) for external generation of set-points, this is not necessarily the case if the SPG block is connected via a feedback loop to the system (see Figure 2).

As the solution of the QP problem cannot be stated in a closed form, classic stability analysis cannot be applied here. Passivity theory has proved as a useful tool to handle such conditions.

### A. Passivity and Stability

Being a sufficient stability condition, passivity is an intuitive approach to stabilize a nonlinear, partially unknown system. Instead of relying on a model, passivity theory makes assumptions on energetic properties to derive stability conditions. The system is energetically passive if the overall energy transmitted to the system $E$ is bounded by a constant $c \in \mathbb{R}$, which depends on the initial energy:

$$E \geq -c^2.$$

There exist multiple approaches to enforce passivity of a system, like the time-domain passivity control concept [42] or the energy bounding algorithm [43]. In [44] a passive set-point modulation (*PSPM*) for a Cartesian impedance controlled manipulator is proposed. As this setup is very similar to our scenario, we apply the basic concepts to formulate a stability proof.

To make the system passive, Lee et.al. [44] propose to implement a virtual energy reservoir, which stores the dissipated energy and use it to execute non-passive actions. This is realized by augmenting the overall system with a virtual energy storage and modulate the desired set-points to the IFC, so that the energy transfered to the virtual spring is limited by the amount of energy left in the reservoir.

### B. Passive Set-Point Modulation

Let us analyze the energy flow for the IFC-SPG system of an individual joint:

$$\tau = K\Delta q - D\dot{q} + \tau_g \tag{14}$$

where $\Delta q = (q_v - q)$ and $K$ and $D$ are the corresponding diagonal entries in the stiffness and damping matrices from (1). Specific joint indexing is omitted in the following for the sake of readability. It can be shown that the robot possesses open loop energetic passivity with respect to the external torques, hence only the port between the SPG-IFC block and the robot needs to be considered. To prove the passivity with respect to the input-output-pair $\{\tau, -\dot{q}\}$, it has to be shown that

$$-\int_0^T \tau\dot{q}dt \geq -c^2 \tag{15}$$

holds for every joint.

The potential energy stored in the virtual spring is

$$E_K = \frac{1}{2}K\Delta q^2$$

and the energy dissipated by the damper after time $T$ is

$$E_D(t) = \int_0^T D\dot{q}^2 dt.$$

It is assumed either that the internal IFC controller runs at a high frequency or a physical spring is installed so that (14) is continuous. The SPG on the other hand, is regarded as a discrete system, which provides set-points with the frequency $\frac{1}{\Delta T}$. Hence the IFC input $\dot{q}_v$ is equivalent to setting a joint position increment $\delta q_v$ with the sampling time $\Delta T$:

$$\delta q_v = \dot{q}_v \Delta T \tag{16}$$

By taking the difference of the potential energy before and after the position increment, the energy increase at the discrete sampling point $i$ due to the set-point setting can be computed:

$$\Delta E_K[i] = \frac{1}{2}K((\Delta q[i] + \delta q_v)^2 - \Delta q[i]^2), \tag{17}$$

where $\bullet[i]$ denotes the respective quantity at the discrete time-step $i$. The sign of $\Delta E_K[i]$ is not determined, hence the set-point increment is a potentially not passive action. Besides that, the dissipated energy during the time interval $\Delta T$ by the damper can be written as

$$\Delta E_D[i] = \int_{t_i}^{t_i+\Delta T} D\dot{q}^2 dt$$

where $t_i$ is the continuous time at the discrete sample point $i$.

With $E_K(t)$ being the energy stored in the spring at time $t$, the energy equation can be stated as

$$-\int_0^T \tau\dot{q}dt - E_D(T) + \sum_{j=1}^{i} \Delta E_K[j] = E_K(T) - E_K(0)$$

and with (15) the IFC-SPG block for one joint is passive with respect to the input-output-pair $\{\tau, -\dot{q}\}$ if

$$-\int_0^T \tau\dot{q}dt = E_K(T) - E_K(0) + E_D(T) - \sum_{j=1}^{i} \Delta E_K[j] \geq -c^2$$

We start with a simple energy reservoir, which stores the dissipated energy from the damper and provides energy to the SPG, used to load the virtual spring. The discrete storage function is

$$E_r[i] = E_r[i-1] + E_D(t_i) - \Delta E_K[i]. \tag{18}$$

Now for practical reasons the following adjustments have to be done:

1) $E_D$ cannot be computed due to missing information on the joint velocity between the sampling points and also missing future data. Therefor the minimal dissipated energy $E_{D_{\text{min}}}$ has to be used instead (see [44] for details on how to obtain $E_{D_{\text{min}}}$).
2) Excessive energy accumulation might allow aggressive/dangerous behavior of the system, while theoretically still being passive. Therefore the energy in the system is limited by a maximum capacity $E_{r_{\text{max}}}$ for $E_r$, what is called energy ceiling in [44].

3) For highly dissipative environments it is useful to transfer some energy to the system. This action also makes sure that the robot does not get stuck when all the energy in the storage is depleted. This is realized by adding the shuffling term $\Delta E_{\text{shuffle}}$ to the energy storage function (18).
4) With a depleted energy reservoir at $t = 0$ no motion would be possible in the beginning of the task, since any extension of the virtual spring would result in a violation of the passivity condition. To overcome this "take-off" problem, the storage is initialized with the energy $E_r[0] = E_{r_{\text{init}}}$.

The new energy storage function with minimum dissipated energy and shuffling term is

$$E_r[i] = E_r[i-1] + \Delta E_{D_{\text{min}}} + \Delta E_{\text{shuffle}} - \Delta E_K(t_i)$$

By requesting $E_r[i] \geq 0$, it is assured that the energy generated by the SPG is not larger than the energy dissipated in the IFC. This limitation can be directly incorporated into the existing SPG framework by adjusting the velocity limits on $\dot{q}_v$. From (16) and (17), new velocity limits can be derived:

$$\dot{q}_{v_m} = \frac{1}{\Delta T}\left(-\sqrt{\frac{2}{K}(E_r[i-1] + \Delta E_{D_{\text{min}}} + \Delta E_{\text{shuffle}}) + \Delta q^2} - \Delta q\right) \quad f. \quad \Delta q \leq 0 \tag{19}$$

$$\dot{q}_{v_M} = \frac{1}{\Delta T}\left(\sqrt{\frac{2}{K}(E_r[i-1] + \Delta E_{D_{\text{min}}} + \Delta E_{\text{shuffle}}) + \Delta q^2} - \Delta q\right) \quad f. \quad \Delta q \geq 0 \tag{20}$$

The energy flow between the SPG, IFC, robot and augmented energy reservoir is depicted in Fig. 3, where the robot is regarded as a system with $n + 1$ ports. One for each joint connected to its IFC and one for the connection of the robot to the environment, with $\boldsymbol{\tau}_{\text{ext}}$ as the external joint torques. Assuming a passive environment, only the energy flow between the robot and the controller has to be analyzed.

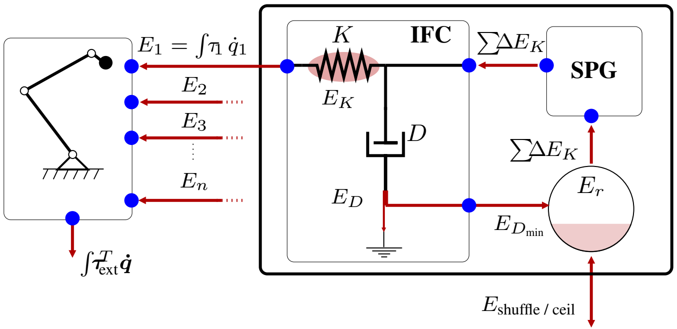
$E_1 = \int \tau_1 \, \dot{q}_1$
$E_2$
$E_3$
$E_n$
$\int \boldsymbol{\tau}_{\text{ext}}^T \dot{\boldsymbol{q}}$
$K$
IFC
$E_K$
$D$
$E_D$
$\sum \Delta E_K$
SPG
$\sum \Delta E_K$
$E_r$
$E_{D_{\text{min}}}$
$E_{\text{shuffle / ceil}}$

Fig. 3. The energy flow with the augmented energy reservoir. The energetic contribution of each joint ($E_1 \ldots E_n$) is regarded separately and is transmitted to the robot from the virtual spring $K$ in the IFC controller. The minimal dissipated energy $E_{D_{\text{min}}}$ is stored in the virtual energy reservoir $E_r$ and the SPG is only allowed to use the energy from this reservoir to load the spring, which is realized by the additional inequality constraints on the virtual joint velocity (19)-(20). By coupling the energy generation of the SPG to the virtual energy reservoir, the overall system is guaranteed to be passive, hence stable.

## VI. Experimental Results

In this paper, one exemplary task with different components, an uncertain task geometry and the possibility for unexpected collisions is presented. The accompanying video shows some other realizations, using the proposed method.

### 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, whose details can be found in [45]. This IFC controller emulates a physical mass-spring-damper system in every joint and has similar behavior as a SEA-type robot. The rate of the discrete controller was $500Hz$ and the stiffness was set to $\boldsymbol{K} = 200\boldsymbol{I}_7 Nm/\text{rad}$. The task convergence factors $\boldsymbol{\Lambda}$ and the regularization factor $\rho = 0.01$ where chosen heuristically. The C++ QP library qpOASES [46], which is an implementation of [47], was used to carry out the optimization (10) – (13).

TABLE I
SET OF SUBTASKS FOR CONSTRAINED TRAJECTORY FOLLOWING

| task | type | $\boldsymbol{\sigma}_m$ | $\boldsymbol{\sigma}_M$ | $\boldsymbol{S}$ |
|---|---|---|---|---|
| 1a,c | joint torque | $-50\text{Nm}\ \boldsymbol{I}_7$ | $50\text{Nm}\ \boldsymbol{I}_7$ | $\boldsymbol{I}_7$ |
| 1b | joint torque | $-5\text{Nm}\ \boldsymbol{I}_7$ | $5\text{Nm}\ \boldsymbol{I}_7$ | $\boldsymbol{I}_7$ |
| 2 | Cart. pose | $\boldsymbol{o}_{\text{init}}$ | $\boldsymbol{o}_{\text{init}}$ | $\begin{bmatrix} \mathbf{0}_3 \\ \boldsymbol{I}_3 \end{bmatrix}$ |
| 3a,b | wrench | 10N | 10N | $[1\ 0\ 0\ 0\ 0\ 0]^T$ |
| 3c | wrench | $\begin{bmatrix} 10 \\ -10 \\ -10 \end{bmatrix}$ N | $\begin{bmatrix} 10 \\ 10 \\ 10 \end{bmatrix}$ N | $\begin{bmatrix} \boldsymbol{I}_3 \\ \mathbf{0}_3 \end{bmatrix}$ |
| 4 | Cart. pose | $\boldsymbol{p}_d(t)$ | $\boldsymbol{p}_d(t)$ | $\begin{bmatrix} \boldsymbol{I}_3 \\ \mathbf{0}_3 \end{bmatrix}$ |
| 5 | joint position | $\mathbf{0}$rad | $\mathbf{0}$rad | $\boldsymbol{I}_7$ |

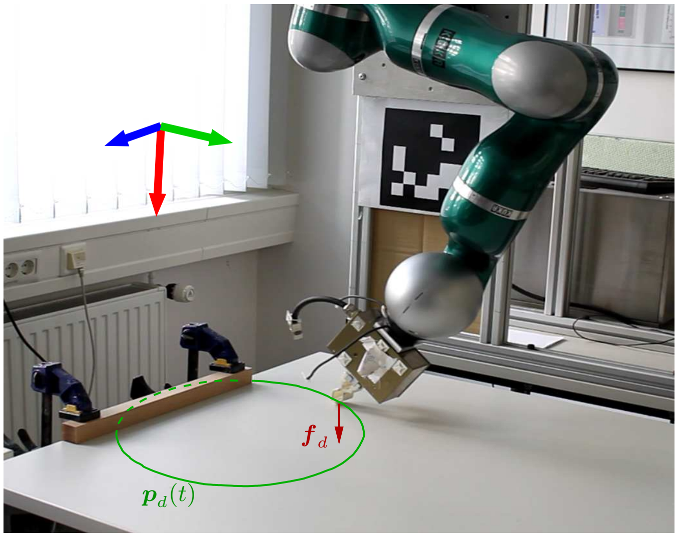

Fig. 4. A sample mixed components task with an obstacle, generating an unexpected perturbation. The main subtasks are to apply a constant interaction force and follow a circular trajectory. Different strategies to deal with unexpected perturbations are compared, relying on the IFC compliance, limiting the joint torques and limiting the end effector forces. The according data is plotted in Figure 5.

### B. Constrained Trajectory Tracking

A generic mixed components task is executed, which is following a Cartesian trajectory on a table surface while applying a constant force on it. In addition there is an obstacle blocking the way, which generates an unexpected perturbation. The setup is depicted in Figure 4 and the main subtasks are summarized in Table I.

Three different task descriptions a,b and c were applied to demonstrate some basic properties of our IFC approach and highlight the simplistic task programming interface. According to Table I tasks 2,4 and 5 are the same for all setups, which are:

- maintaining the initial orientation of the end effector $\boldsymbol{o}_{\text{init}}$ (task 2)

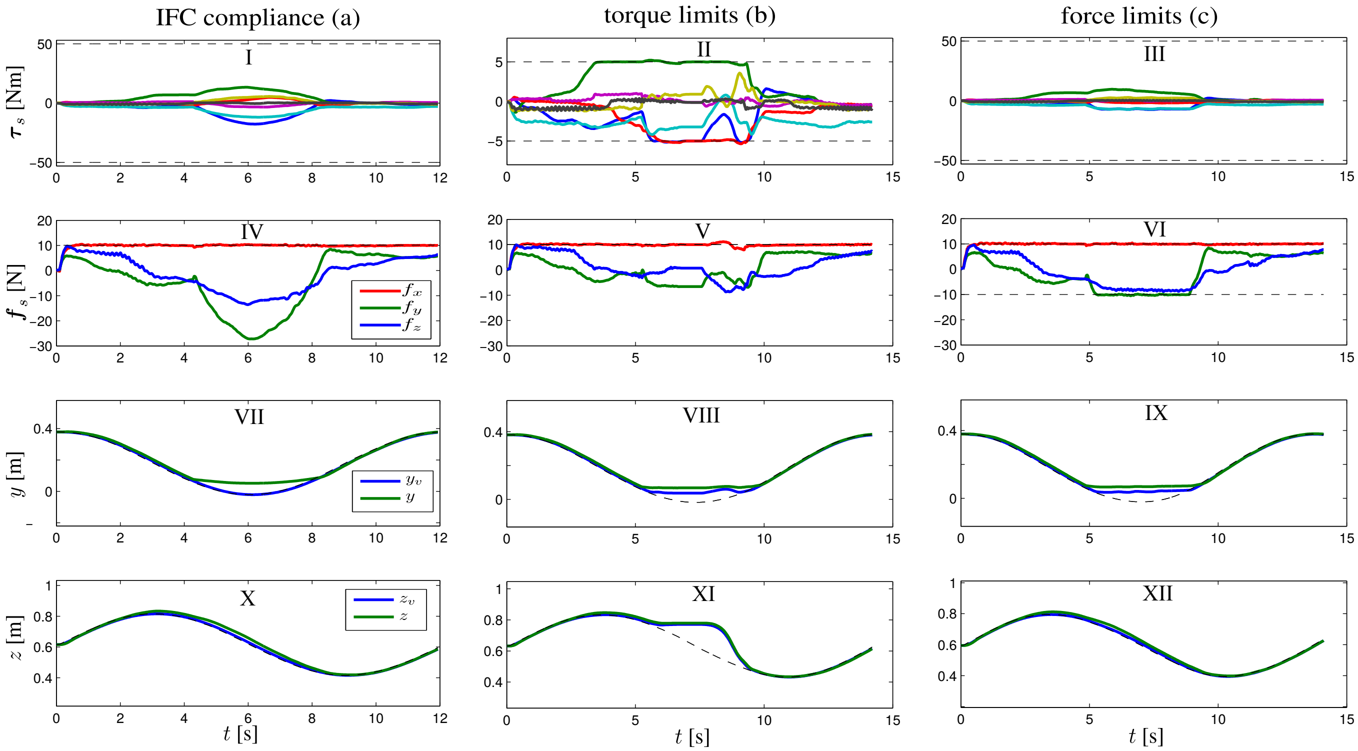

Fig. 5. Each column shows the propagation of the relevant task variables for three different task specifications from Table I. The interaction forces and torques are computed according to the static relations in Sec. III-B. For the trajectory following task, both the virtual and the physical position of the end effector are plotted. In each graph the dashed line represents the task constraints (desired value or limits).
Left: the obstacle impedes the motion of the physical end effector along the $y$-axis (VII). This is compensated by the inherent compliance of the IFC, leading to additional interaction forces in the $y$- and $z$-directions (IV). Forces along $y$ and $z$ are not constrained by the task specification. This approach corresponds basically to a classical IFC application.
Middle: the torque limits are lowered to ±5Nm in order to avoid large interaction torques (II). This prevents the virtual end effector from penetrating further into the obstacle (VIII). However, due to friction, in this case the new torque limits are so conservative that at some point no motion is possible at all while the virtual end-effector penetrates the obstacle (XI).
Right: another possibility is to explicitly limit the interaction forces instead of the torques, making it possible to tune the allowed forces in Cartesian space, to account for expected resistance, e.g. friction (VI).

- following a time dependent trajectory in the $y$-$z$ plane with the end effector (task 4)
- maximizing the distance of the joints to their limits (task 5)

First, the force task is specified as an equality constraint along the $x$-axis only (specification a). Then the torque limits are lowered to avoid large interaction torques (specification b). Finally, the force task is defined as an equality in its $x$-component and as an inequality in the $y$ and $z$ components to limit the interaction forces explicitly (specification c). The results are depicted in Figure 5.

## VII. Conclusion

We combined different methods from robot control to develop an additional layer between the task programming and the low level indirect force-controlled robot. The assumptions on the actual robotic hardware are very general, hence the formulated method is applicable on any IFC-like architecture or also SEA type manipulators, which represent a physical realization of an IFC.

The following information is required from the IFC-robot system:

- the kinematic parameters
- optionally: the joint position, velocity, acceleration and torque limits
- the IFC joint stiffness $\boldsymbol{K}$
- the physical and virtual joint positions $\boldsymbol{q}$ and $\boldsymbol{q}_v$

A task is completely specified by providing

- the task type (or task Jacobian $\boldsymbol{A}$)
- the lower and upper bounds for the task variable $\boldsymbol{\sigma}_m$ and $\boldsymbol{\sigma}_M$ (or $\boldsymbol{\sigma}_d$)
- the feedforward differential task variables $\dot{\boldsymbol{\sigma}}_{m_{ff}}$ and $\dot{\boldsymbol{\sigma}}_{M_{ff}}$
- the convergence rate $\boldsymbol{\Lambda}$
- the subspace matrix $\boldsymbol{S}$

The resulting scheme allows the specification of a hierarchical set of equality and inequality tasks, consisting of force and positioning components on joint and Cartesian level. The reduced performance in terms of accuracy is deliberately accepted in trade off a permanently compliant behavior and to allow intuitive, yet powerful task programming in highly unstructured environments. Neither modification nor detailed information of the IFC is required, which makes the approach very general in terms of robotic hardware.

## References

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

[2] E. Lutscher and G. Cheng, “A Practical Approach to Generalized Hierarchical Task Specification for Indirect Force Controlled Robots,” in *IEEE/RSJ International Conference on Intelligent Robots and System (IROS)*, 2013.

[3] ——, “Hierarchical inequality task specification for indirect force controlled robots using quadratic programming,” in *IEEE/RSJ International Conference on Intelligent Robots and System*, 2014.

[4] R. Colbaugh, H. Seraji, and K. Glass, “Direct adaptive impedance control of robot manipulators,” *Journal of Robotic Systems*, vol. 10, no. 2, pp. 217–248, 1993.

[5] Z. Lu and A. A. Goldenberg, “Robust impedance control and force regulation: theory and experiments,” *The International journal of robotics research*, vol. 14, no. 3, pp. 225–254, 1995.

[6] R. Anderson and M. W. Spong, “Hybrid impedance control of robotic manipulators,” *IEEE Journal of Robotics and Automation*, vol. 4, no. 5, pp. 549–556, 1988.

[7] M. A. Peshkin, "Programmed compliance for error corrective assembly," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 4, pp. 473–482, 1990.

[8] J. M. Schimmels and M. A. Peshkin, "Admittance matrix design for force-guided assembly," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 2, pp. 213–227, 1992.

[9] A. Jain and C. C. Kemp, "Pulling open novel doors and drawers with equilibrium point control," in *2009 9th IEEE-RAS International Conference on Humanoid Robots*. Ieee, Dec. 2009, pp. 498–505.

[10] C. Ott, B. Bäuml, C. Borst, and G. Hirzinger, "Employing cartesian impedance control for the opening of a door: A case study in mobile manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on mobile manipulators: Basic techniques, new trends & applications*, 2005.

[11] J. Sturm, A. Jain, and C. Stachniss, "Operating articulated objects based on experience," in *IEEE/RSJ International Conference on Intelligent Robots and System*, 2010, pp. 2739–2744.

[12] L. Baptista, J. Sousa, and J. da Costa, "Predictive Force Control of Robot Manipulators in Nonrigid Environments," in *Industrial Robotics: Theory, Modelling and Control*, 2006, no. December, pp. 841–875.

[13] S. Jung, "Force Tracking Impedance Control for Robot Manipulators with an Unknown Environment: Theory, Simulation, and Experiment," *The International Journal of Robotics Research*, vol. 20, no. 9, pp. 765–774, Sep. 2001.

[14] S. Jung, T. Hsia, and R. Bonitz, "Force Tracking Impedance Control of Robot Manipulators Under Unknown Environment," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 3, pp. 474–483, May 2004.

[15] T. Lasky and T. Hsia, "On force-tracking impedance control of robot manipulators," *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, no. April, pp. 274–280, 1991.

[16] H. Seraji and R. Colbaugh, "Force Tracking in Impedance Control," *The International Journal of Robotics Research*, vol. 16, no. 1, pp. 97–117, Feb. 1997.

[17] S. Singh and D. Popa, "An analysis of some fundamental problems in adaptive control of force and impedance behavior: theory and experiments," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 6, pp. 912–921, 1995.

[18] 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.

[19] J. De Schutter and H. Van Brussel, "Compliant Robot Motion I. A Formalism for Specifying Compliant Motion Tasks," *The International Journal of Robotics Research*, vol. 7, no. 4, pp. 3–17, Aug. 1988.

[20] ——, "Compliant Robot Motion II. A Control Approach Based on External Control Loops," *The International Journal of Robotics Research*, vol. 7, no. 4, pp. 18–33, Aug. 1988.

[21] H. Bruyninckx and J. De Schutter, "Specification of force-controlled actions in the "task frame formalism"-a synthesis," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 581–589, 1996.

[22] 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.

[23] J. de Schutter, "Estimating First-Order Geometric Parameters and Monitoring Contact Transitions during Force-Controlled Compliant Motion," *The International Journal of Robotics Research*, vol. 18, no. 12, pp. 1161–1184, Dec. 1999.

[24] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Polyhedral contact formation modeling and identification for autonomous compliant motion," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 26–41, Feb. 2003.

[25] D. Raunhardt and R. Boulic, "Progressive clamping," in *IEEE International Conference on Robotics and Automation*, no. April, 2007, pp. 10–14.

[26] T. Chan and R. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 286–292, 1995.

[27] 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.

[28] F. Flacco, A. D. Luca, and O. Khatib, "Motion control of redundant robots under joint constraints: Saturation in the null space," *IEEE International Conference on Robotics and Automation*, 2012.

[29] 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.

[30] 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.

[31] A. Escande, N. Mansard, and P. Wieber, "Fast Resolution of Hierarchized Inverse Kinematics with Inequality Constraints," in *IEEE International Conference on Robotics and Automation*, no. 4, 2010, pp. 3733–3738.

[32] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J. Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, 2013.

[33] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.

[34] A. Ben-Israel and T. N. Greville, *Generalized Inverses*. Springer, 2003, vol. 13.

[35] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, 1987.

[36] F. Flacco and A. D. Luca, "A reverse priority approach to multi-task control of redundant robots," in *IEEE/RSJ International Conference on Intelligent Robots and System*, 2014.

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

[38] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, pp. 90–98, 1986.

[39] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Journal of dynamic systems, measurement, and control*, vol. 108, no. 3, pp. 163–171, 1986.

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

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

[42] B. Hannaford, "Time-domain passivity control of haptic interfaces," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 1–10, 2002.

[43] J. Kim and J. Ryu, "Stable haptic interaction control using energy bounding algorithm," *IEEE/RSJ International Conference on Intelligent Robots and System*, pp. 1210–1217, 2004.

[44] D. Lee and K. Huang, "Passive-set-position-modulation framework for interactive robotic systems," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 354–369, 2010.

[45] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A unified passivity based control framework for position, torque and impedance control of flexible joint robots," *The International Journal of Robotics Research*, vol. 26, no. 1, pp. 23–39, 2007.

[46] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[47] H. Ferreau, H. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit mpc," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.