Robotica: (2020) volume 38, pp. 1807–1823. © Cambridge University Press 2019
doi:10.1017/S0263574719000973

Unified Virtual Guides Framework for Path Tracking Tasks

Leon Žlajpah* and Tadej Petrič

Department of Automation, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia
E-mail: tadej.petric@ijs.si

(Accepted May 25, 2019. First published online: July 26, 2019)

SUMMARY
In this paper, we propose a novel unified framework for virtual guides. The human–robot interaction is based on a virtual robot, which is controlled by the admittance control. The unified framework combines virtual guides, control of the dynamic behavior, and path tracking. Different virtual guides and active constraints can be realized by using dead-zones in the position part of the admittance controller. The proposed algorithm can act in a changing task space and allows selection of the tasks-space and redundant degrees-of-freedom during the task execution. The admittance control algorithm can be implemented either on a velocity or on acceleration level. The proposed framework has been validated by an experiment on a KUKA LWR robot performing the Buzz-Wire task.

KEYWORDS: Human–robot cooperation; Admittance control; Virtual guides; Redundant robots; Path following.

1. Introduction

Industrial robots are nowadays widely used for performing different tasks, by using well-known control systems that allow them to operate fast, accurate, and with a persisting strength. Equipped with adequate sensory systems they can also operate in unstructured environments to some extent due to the limitations in their perception of unpredicted situations, for example unexpected collisions. To avoid such situations, robots are usually set up behind fences where they are solely performing motions along some predefined paths without having all the details about the environment. In such setups, robots might not react properly in case of an unexpected situation. To enable robots to perform tasks outside fences in an unstructured environment with possible human presence, a novel control framework is required that can react accordingly.

By exploiting the capabilities of modern robots and combining them with the skills of a human leads to robot applications where the physical human–robot interaction (pHRI) is essential.[1,2] This cooperation does not involve only a direct or indirect physical interaction between the partners but also requires collaboration between them at the "cognitive" level. Some of the problem domains of pHRI which have been addressed in the last years are learning and imitation,[3,4] human and robot roles in this cooperation,[5,6] compliance control,[2,7] and safety.[1,8] One type of human–robot interaction is also cooperation, where we exploit the mechanical capabilities of robotic devices and combine them with perception and cognitive capabilities of humans to achieve an overall goal.[9] Here typically the role of the robot is to follow the human.

In human–robot co-manipulation, the operator typically controls the robot motion through the direct contact (cobots) or indirect contact (telemanipulators). Here, the most common method is to support and guide the human motion using *virtual guides* introduced in ref. [10] as *virtual fixtures*. Virtual guides are constraints which limit the motion of a robotic system. They have a similar function as the mechanical constraints except that they are implemented in the controller. In ref. [11],

* Corresponding author. E-mail: leon.zlajpah@ijs.si

the virtual fixtures are defined as collaborative strategies which can be used to improve or assist human to perform manipulation tasks. Fixtures can constrain the motion in different ways. *Guidance active constraints* force the operator to move toward a specific target point or to follow a predefined path,[11,12] while the *regional constraints* prevent the operator to move into the specific region and/or force the operator to move out of it.[11,13] The guides can also be modified during the task execution using refs.[12,14] or they can be learned iteratively.[15–17] In general, both type of guides are equivalent except that *guidance* constraints are attractive and *regional* constraints are repulsive. Although the guides can be related to the dynamic motion,[18] in most cases only static guides constraining the position are used. In the past different methods for calculating active constraint were given from using points or line, up to the methods considering complex surfaces or volumes.[11–14,19–21]

The physical human–robot co-manipulation relies on interaction forces between the operator and the robot. Therefore, the robot control algorithm is usually based on impedance or admittance approach.[22] Using the impedance control approach, the motion produces forces, which are then felt by the operator, whereas using the admittance control approach the forces applied to the robot by the operator contribute to the robot motion. In general, the impedance control is more suitable when the robot is in contact with the stiff environment, and the admittance control has its advantages when the environment is more compliant.[23] The distinction between the impedance and the admittance control is significant when used in pHRI framework. Namely, the impedance-based constraint generates forces to nullify the motion that violates the constraint, and the admittance-based constraint prevents the motion induced by the external forces, which would violate the constraint. For safe operation of the robot control system, the stability has to be assured, which is related to the implementation of the impedance or admittance controller. In the case of pHRI, we need to consider also the impedance of the human arm, which has characteristics of a compliant environment, when defining the impedance/admittance of the robotic system.[24–26]

In the preliminary study presented at the 27th International Conference on Robotics in Alpe-Adria Danube Region (RAAD 2018),[27] the admittance control using virtual robot was presented. The specific contributions of this paper are an extended method formulation with a more in-depth explanation, introduction of proxies for virtual guides, a more thorough overview of the related work, and an additional discussion of novel results.

In Section 2, we explain the concept of a virtual robot with implemented admittance control. In Section 3, we present our approach to virtual guides for path tracking and avoiding forbidden zones, where we also show how the task space and the redundant degrees of freedom (DOFs) may be changed regarding the state of the system. Finally, in Section 4, we illustrate the use of virtual robots and guides on KUKA LWR robot arm guiding a human to follow a path.

2. Interaction Control

Let the configuration of the manipulator be represented by joint positions $\boldsymbol{q} \in \mathbb{R}^n$. For the manipulator consisting of rigid bodies the joint space equations of motion can be written in the form[22]

$$\boldsymbol{\tau} = \mathbf{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{\tau}_F \,, \tag{1}$$

where $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of control torques, $\mathbf{H}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\boldsymbol{h} \in \mathbb{R}^n$ is the vector of Coriolis and centripetal forces, $\boldsymbol{g} \in \mathbb{R}^n$ is the vector of gravity forces, and $\boldsymbol{\tau}_F \in \mathbb{R}^n$ represents general torques due to external forces/torques $\boldsymbol{F}_i \in \mathbb{R}^6$ acting on the body of the robot manipulator

$$\boldsymbol{\tau}_F = \sum_i \mathbf{J}_i^T \boldsymbol{F}_i \,, \tag{2}$$

where $\mathbf{J}_i \in \mathbb{R}^{6\times n}$ is the Jacobian matrix for the i-th force acting point.

Let $\boldsymbol{p} = [x, y, z]^T \in \mathbb{R}^3$ represent the position, $\mathcal{Q} = \{\eta, \boldsymbol{\epsilon}\} \in \mathrm{S}^3 \subset \mathbb{R}^4$ (unit quaternion) the spatial rotation, and $\boldsymbol{v} = [\dot{\boldsymbol{p}}^T \boldsymbol{\omega}^T]^T$ the spatial velocity of the end-effector, where $\dot{\boldsymbol{p}} \in \mathbb{R}^3$ and $\boldsymbol{\omega} \in \mathbb{R}^3$ are the linear and angular spatial velocity, respectively. Then, the kinematics of the manipulator can be described with the following equations:

$$\{\boldsymbol{p}, \mathcal{Q}\} = \{\boldsymbol{p}(\boldsymbol{q}), \mathcal{Q}(\boldsymbol{q})\} \tag{3}$$

$$\boldsymbol{v} = \mathbf{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \,, \tag{4}$$

$$\dot{\boldsymbol{v}} = \mathbf{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\mathbf{J}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}, \tag{5}$$

where $\mathbf{J}$ is the geometric Jacobian matrix and $\dot{\mathbf{J}}$ is its time derivative.

https://doi.org/10.1017/S0263574719000973 Published online by Cambridge University Press

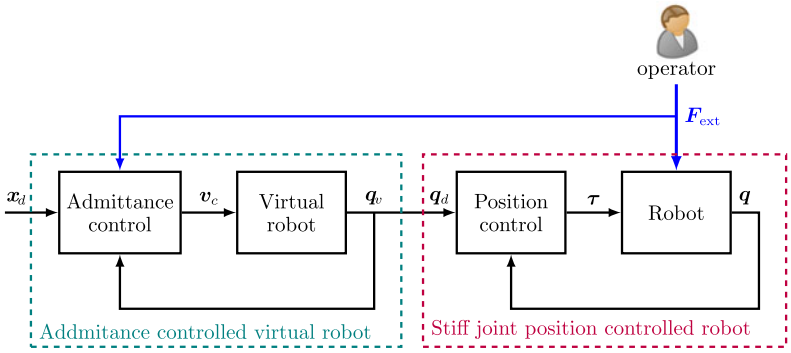

Fig. 1. Implementation of the admittance control on the virtual robot, which is generating desired motion for the stiff position controlled robot.

Modern robots, especially those used in pHRI framework, are equipped with control algorithms[28,29] which consider dynamics of the robot to decouple and linearize the robot system and to achieve the desired dynamic behavior of the robot. We assume that the low-level robot control completely decouples and linearizes the system so that the close-loop behavior can be described as

$$\ddot{\boldsymbol{e}}_q + \mathbf{K}_v \dot{\boldsymbol{e}}_q + \mathbf{K}_p \boldsymbol{e}_q = \boldsymbol{\tau}_F \,, \tag{6}$$

where $\boldsymbol{q}_d$ is the desired motion of the robot end-effector, and $\boldsymbol{e}_q$ is the position error, $\boldsymbol{e}_q = \boldsymbol{q}_d - \boldsymbol{q}$. The toques $\boldsymbol{\tau}_F$ are due to the contact forces when the robot is in contact with the environment, or due to the external forces applied to the robot.

Typically, the human–robot interaction in the industry takes place at the force level. Here a human operator can interact with a robot by applying forces to a point on the robot body in order to move and reconfigure the robot. To perform the task, the human operator can apply forces at the end-effector of the robot. Based on these forces (detected by force/torques sensors) the control algorithm can generate a motion, which moves the robot in the desired direction or it can block the motion in that direction if needed.

Let gains $\mathbf{K}_v$ and $\mathbf{K}_p$ assure the stability of the robot. We assume that the robot dynamics is faster than the expected dynamics of the human-induced motion. This means that a human does not perceive the delay between the commanded motion and actual robot motion. Next, we assume that during the human–robot co-manipulation no contact with stiff environment occurs. Hence, we can select high gains $\mathbf{K}_p$, which make the robot stiff. Consequently, motion tracking error is negligible and Eq. (6) can be simplified as

$$\boldsymbol{q} \approx \boldsymbol{q}_d \,. \tag{7}$$

Under these assumptions, we find that the admittance control approach is a suitable control framework to be used for pHRI. In general, the weakness of admittance control is when the robot is in contact with the stiff environment. However, when used for virtual guides this is not the case. Namely, if the robot end-effector is in contact with the stiff environment, then the operator does not need assistance from virtual guides as the environment is guiding or restricting the motion of the end-effector.

We have implemented the admittance control using a virtual robot and not directly on the real robot and does not rely on direct feedback from the robot. Based on the desired end-effector motion $\boldsymbol{x}_d$ and measured external forces $\boldsymbol{F}_{\text{ext}}$ due to the human interaction, admittance control generates the motion of the virtual robot $\boldsymbol{q}_v$. The motion $\boldsymbol{q}_v$ is then applied as the input $\boldsymbol{q}_d$ of the inner position control loop (see Fig. 1). If the robot is kinematically redundant, then $\boldsymbol{q}_d$ includes also the null-space motion.

Typically, the virtual robot dynamics represents a mass-damping-spring system with the desired dynamic properties. By their nature, a virtual robot can have any dynamic properties. However, it is necessary to consider that the virtual robot dynamics is bounded by the dynamic properties of the position controlled robot. Moreover, the selection of the virtual mass and damping depends on the task.[24,26,30,31] The virtual robot can be selected as an ideal system represented by a double integrator

https://doi.org/10.1017/S0263574719000973 Published online by Cambridge University Press

system, where the input is the desired acceleration and the output is the position

$$\boldsymbol{q}_v = \iint \ddot{\boldsymbol{q}}_v \, dt \, . \tag{8}$$

The dynamic behavior of the virtual robot then depends on the control used for the virtual robot motion control. As the human operates in the Cartesian space, it is reasonable to design the control in the Cartesian space and in most cases with the same dynamic properties of the system in all spatial directions.

Let $\boldsymbol{p} = [x, y, z]^T \in \mathbb{R}^3$ represent the position of the virtual robot, $\mathcal{Q} = \{\eta, \boldsymbol{\epsilon}\} \in \mathrm{S}^3 \subset \mathbb{R}^4$ (unit quaternion) the spatial rotation, and $\boldsymbol{v} = [\dot{\boldsymbol{p}}^T \boldsymbol{\omega}^T]^T$ the spatial velocity of the end-effector of the virtual robot, where $\dot{\boldsymbol{p}} \in \mathbb{R}^3$ and $\boldsymbol{\omega} \in \mathbb{R}^3$ are the linear and angular spatial velocity, respectively. Then, we can define the end-effector position and orientation error $\boldsymbol{e}_x \in \mathbb{R}^6$ as

$$\boldsymbol{e}_x = \begin{bmatrix} \boldsymbol{p}_d - \boldsymbol{p} \\ 2 \log(\mathcal{Q}_d * \mathcal{Q}^{-1}) \end{bmatrix} , \tag{9}$$

and the end-effector spatial velocity error as

$$\boldsymbol{e}_v = \boldsymbol{v}_d - \boldsymbol{v} \, , \tag{10}$$

where subscript $(.)_d$ denotes the desired value of the virtual robot variable. We propose to select the control input as

$$\boldsymbol{a}_c = \dot{\boldsymbol{v}}_d + \mathbf{M}^{-1}(\mathbf{D}\boldsymbol{e}_v + \mathbf{K}\boldsymbol{e}_x + \boldsymbol{F}_{\text{ext}}) \, , \tag{11}$$

where $\boldsymbol{F}_{\text{ext}} \in \mathbb{R}^6$ are the forces/torques applied by the human on the end-effector, $\mathbf{M} = \text{diag}(M) \in \mathbb{R}^{6\times6}$ is a positive definite matrix, $\mathbf{D} = \text{diag}(D_i) \in \mathbb{R}^{6\times6}$ positive semi-definite matrix, and $\mathbf{K} = \text{diag}(K_i) \in \mathbb{R}^{6\times6}$ positive semi-definite matrix representing the desired inertia, damping, and stiffness of the virtual robot, respectively. Finally, to obtain the desired control input $\ddot{\boldsymbol{q}}_d$ for the virtual robot, we have to solve the inverse kinematics. In the case of redundant robots, the inverse kinematic controller is given in the form

$$\ddot{\boldsymbol{q}}_v = \mathbf{J}^{\#}(\boldsymbol{a}_c - \dot{\mathbf{J}}\dot{\boldsymbol{q}}_v) + (\mathbf{I} - \mathbf{J}^{\#}\mathbf{J})\ddot{\boldsymbol{q}}_{vn} \, , \tag{12}$$

where $\mathbf{J}^{\#}$ is a generalized inverse of the Jacobian matrix $\mathbf{J}$ and $\ddot{\boldsymbol{q}}_{vn}$ are arbitrary joint accelerations (used to perform lower priority tasks). Combining (8), (11), and (12), we obtain the task space dynamics in the form

$$\mathbf{M}\dot{\boldsymbol{e}}_v + \mathbf{D}\boldsymbol{e}_v + \mathbf{K}\boldsymbol{e}_x = -\boldsymbol{F}_{\text{ext}}, \tag{13}$$

and the desired virtual robot dynamic behavior is obtained by properly selecting $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$. From Fig. 1, we can see that the only feedback loop from the robot subsystem to the virtual robot subsystem is over the human. Hence, to assure the stability of the system $\mathbf{M}$, $\mathbf{D}$ and $\mathbf{K}$ have to be selected so that the virtual robot subsystem is passive. Note that the robot subsystem is stable by assumption, and although humans are not passive per se, they are able to adapt to the environment and tend to be passive.[20] If so, the overall system is stable.

Experiments with human interacting with robots have shown that damping has greater influence on the human perception than the virtual mass of the robotic system.[30,31] Therefore, a simplified version of admittance control at the velocity level can be used. In this case, the virtual robot is defined as a single integrator, where the input is the desired velocity

$$\boldsymbol{q}_v = \int \dot{\boldsymbol{q}}_v \, dt \, , \tag{14}$$

and the admittance control is defined as

$$\boldsymbol{v}_c = \boldsymbol{v}_d + \mathbf{D}^{-1}(\mathbf{K}\boldsymbol{e}_x + \boldsymbol{F}_{\text{ext}}) \, . \tag{15}$$

On velocity level the inverse kinematics is defined as

$$\dot{\boldsymbol{q}}_d = \mathbf{J}^{\#}\boldsymbol{v}_c + (\mathbf{I} - \mathbf{J}^{\#}\mathbf{J})\dot{\boldsymbol{q}}_{vn} \, , \tag{16}$$

https://doi.org/10.1017/S0263574719000973 Published online by Cambridge University Press

where $\dot{\boldsymbol{q}}_{vn}$ are arbitrary joint velocities, which can be used to perform some lower priority tasks. Combining (14), (15), and (16), the dynamics of the virtual robot is

$$\mathbf{D}\boldsymbol{e}_v + \mathbf{K}\boldsymbol{e}_x = -\boldsymbol{F}_{\text{ext}} \,. \tag{17}$$

If we want to consider also the inertia in this framework we have to use a first-order filter for the external force

$$\hat{\boldsymbol{F}}_{\text{ext}} = \frac{\mathbf{D}}{\mathbf{M}s + \mathbf{D}} \boldsymbol{F}_{\text{ext}} \tag{18}$$

resulting from the following relation

$$\boldsymbol{F} = \mathbf{M}\ddot{x} + \mathbf{D}\dot{x} \,. \tag{19}$$

Using $\hat{\boldsymbol{F}}_{\text{ext}}$ instead of $\boldsymbol{F}_{\text{ext}}$ in (15) yields the virtual robot dynamics in the form

$$\mathbf{M}\dot{\boldsymbol{e}}_v + \left(\mathbf{D} + \frac{\mathbf{M}\,\mathbf{K}}{\mathbf{D}}\right)\boldsymbol{e}_v + \mathbf{K}\boldsymbol{e}_x = -\boldsymbol{F}_{\text{ext}} \,. \tag{20}$$

By comparing (13) and (20), we can see that using filtered external force yields a similar close loop behavior as the admittance control based on acceleration, except that the effective damping is higher.

2.1. Control in task space

Usually the motion of the end-effector is described in the Cartesian space representing the operational world space $\mathcal{O}$. However, when the task or the constraints require a special behavior in some directions which are not aligned with the world space, then we propose to formulate the control and the constraints in the relevant task space $\mathcal{T}$. This can be done by mapping the control and constraints from $\mathcal{O}$ into a suitable task space $\mathcal{T}$.[32] As the origin of $\mathcal{T}$ is not important, we apply in the admittance controller only the rotation of the task space $\mathcal{T}$. So, the admittance controller (15) becomes

$${}^t\boldsymbol{v}_c = \overline{\mathbf{R}}_t^T \boldsymbol{v}_d + \mathbf{D}^{-1}(\mathbf{K}\overline{\mathbf{R}}_t^T \boldsymbol{e}_x + \overline{\mathbf{R}}_t^T \boldsymbol{F}_{\text{ext}}) \,, \tag{21}$$

and the inverse kinematics at velocity level (16) becomes

$$\dot{\boldsymbol{q}}_d = (\overline{\mathbf{R}}_t^T \mathbf{J})^{\#}\, {}^t\boldsymbol{v}_c + (\mathbf{I} - (\overline{\mathbf{R}}_t^T \mathbf{J})^{\#}\overline{\mathbf{R}}_t^T \mathbf{J})\dot{\boldsymbol{q}}_n \,. \tag{22}$$

Here, the matrix $\overline{\mathbf{R}}_t$ is defined as

$$\overline{\mathbf{R}}_t = \begin{bmatrix} \mathbf{R}_p & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{R}_o \end{bmatrix} \,, \tag{23}$$

where $\mathbf{R}_p$ and $\mathbf{R}_o$ are rotation matrices for mapping the positions and orientations from the world frame $\mathrm{F}_{\mathcal{O}}$ to the task-space frame $\mathrm{F}_{\mathcal{T}}$, respectively.

2.2. Redundancy resolution

In general, virtual guides consider all spatial position and orientation DOFs. In our previous work,[27] we have shown that in some regions it is not required that all six spatial DOFs are controlled. Then the task space $\mathcal{T}$ is a lower-dimensional subspace of the operational space $\mathcal{O}$, $\dim(\mathcal{T}) < \dim(\mathcal{O}) = 6$, and the robot can be treated as a *functionally redundant* robot. In ref. [27], we have proposed to do the redundancy resolution by using the Jacobian which considers only DOFs in the task space $\mathcal{T}$ and using it in (12) or (16) to map the joint velocities due to the lower priority tasks into the null-space of $\mathbf{J}$. The aim of using null-space is to prevent influence of secondary tasks on the primary task, that is the motion generated by the secondary task should not interfere with the motion in the task space. However, when the dimensionality of $\mathcal{T}$ changes between the task zones, then the number of rows of $\mathbf{J}$ change discontinuously, and in general, the generated motion is not continuous and smooth.

To overcome this problem, we propose a novel strategy for the secondary tasks. The novel strategy is still exploiting the functional redundancy for secondary tasks, but instead of using a null-space

https://doi.org/10.1017/S0263574719000973 Published online by Cambridge University Press

projection we calculate virtual end-effector forces/torques, which initiate the same motion as the null-space term in (12) or (16), and this motion is not constrained as the task space motion.

The virtual forces are calculated as follows. Let $\mathbf{J} \in \mathbb{R}^{6\times n}$ be the Jacobian matrix of the robot in $\mathcal{O}$. Then, the Jacobian matrix $\mathbf{J}_\mathcal{T}$ in $\mathcal{T}$ can be obtained as

$$\mathbf{J}_\mathcal{T} = \mathbf{S}\mathbf{J}\,, \tag{24}$$

where $\mathbf{S} \in \mathbb{R}^{6\times 6}$ is a diagonal selection matrix for selecting the task space spatial directions. Next, we define virtual external forces/torques $\boldsymbol{F}_n$ based on the joint velocities due to the secondary task $\dot{\boldsymbol{q}}_n$ as

$$\boldsymbol{F}_n = \frac{1}{\alpha}\mathbf{A}\mathbf{J}(\mathbf{I} - \mathbf{J}_\mathcal{T}^{\#}\mathbf{J}_\mathcal{T})\dot{\boldsymbol{q}}_{vn}\,, \tag{25}$$

where $\mathbf{J}_\mathcal{T}^{\#}$ is a generalized inverse of $\mathbf{J}_\mathcal{T}$ and $\mathbf{A} \in \mathbb{R}^{6\times 6}$ is a diagonal scaling matrix. As virtual force $\boldsymbol{F}_n$ generates the motion of the end-effector, it is necessary to limit the maximal value of components $\boldsymbol{F}_{n,i}$. For that, we use proportional limiter implemented as a scaling factor α

$$\alpha = \max\left(1, \max_i\left(\frac{\boldsymbol{F}_{n,i}}{\boldsymbol{F}_{n,i,\max}}\right)\right)\,, \tag{26}$$

where $\boldsymbol{F}_{n,i,\max}$ are the maximal allowed values of virtual force components $\boldsymbol{F}_{n,i}$. The virtual forces $\boldsymbol{F}_n$ are then included in (11) or (15) yielding

$$\boldsymbol{a}_c = \dot{\boldsymbol{v}}_d + \mathbf{M}^{-1}(\mathbf{D}\boldsymbol{e}_v + \mathbf{K}\boldsymbol{e}_x + \boldsymbol{F}_{\text{ext}} + \boldsymbol{F}_n)\,, \tag{27}$$

or

$$\boldsymbol{v}_c = \boldsymbol{v}_d + \mathbf{D}^{-1}(\mathbf{K}\boldsymbol{e}_x + \boldsymbol{F}_{\text{ext}} + \boldsymbol{F}_n)\,. \tag{28}$$

Note that the components of $\boldsymbol{F}_n$ corresponding to directions of $\mathcal{T}$ are zero, $\mathbf{J}_\mathcal{T}^{\#}\boldsymbol{F}_n = 0$, and hence $\boldsymbol{F}_n$ is not influencing the task space motion. To exploit also the intrinsic redundancy, the null-space term in (12) or (16) is still used.

2.3. Path

Let the path $\boldsymbol{f}$ be given as a parametric curve in $SE(3)$ as

$$\{\boldsymbol{p}, \mathcal{Q}\} = \boldsymbol{f}(s) \tag{29}$$

$$\boldsymbol{v} = \mathbf{J}_s\dot{s} \tag{30}$$

$$\dot{\boldsymbol{v}} = \mathbf{J}_s\ddot{s} + \dot{\mathbf{J}}_s\dot{s}^2\,, \tag{31}$$

where $\mathbf{J}_s \in \mathbb{R}^{6\times 1}$ is the path Jacobian and represents the path direction at s. It is assumed that $\boldsymbol{f}(s)$ is continuous for all s and that s is strictly increasing. A good candidate for s is the path length. Note that path can be also defined in the task space $\mathcal{T} \subset SE(3)$ where some of the unused spatial directions are omitted. Then the dimension of $\boldsymbol{v}$ and $\mathbf{J}_s$ is lower according to the $\dim(\mathcal{T})$.

To describe a general path we use a parametric description

$$\boldsymbol{f}(s) = \frac{\sum_{i=1}^{m} \boldsymbol{w}_i\Psi_i(s)}{\sum_{i=1}^{m}\Psi_i(s)}, \tag{32}$$

where $\boldsymbol{w}_i \in \mathbb{R}^{dim(\boldsymbol{f})}$ are weights that define the path and Ψ_i are the radial basis function (RBF) kernels given as

$$\Psi_i = \exp\left(-\frac{(s-c_i)^2}{2h_i}\right)\,, \tag{33}$$

centered at c_i, where h_i are defining the widths of kernel functions. If not stated otherwise, we use c_i, $i = 1, \ldots, m$, that equally spread along the path, that is between 0 and $s_{\max}$. The number of kernel

https://doi.org/10.1017/S0263574719000973 Published online by Cambridge University Press

functions m depends on the required accuracy of the path description. Using $\boldsymbol{\Phi}(s) \in \mathbb{R}^m$ defined as a row vector with components

$$\boldsymbol{\Phi}_i(s) = \frac{\Psi_i(s)}{\sum_{i=1}^{m} \Psi_i(s)} \tag{34}$$

in (32) yields

$$\boldsymbol{f}(s) = \boldsymbol{\Phi}(s)\boldsymbol{w}\,. \tag{35}$$

Knowing the values of $\boldsymbol{f}(s)$ along the path, the corresponding weights $\boldsymbol{w}$ can be found by solving (35) in a least square sense

$$\boldsymbol{w} = \left(\hat{\boldsymbol{\Phi}}^T \hat{\boldsymbol{\Phi}}\right)^{-1} \hat{\boldsymbol{\Phi}}^T \hat{\boldsymbol{f}}\,, \tag{36}$$

where $\hat{\boldsymbol{f}} \in \mathbb{R}^{r\times \dim(f)}$ are path values and $\hat{\boldsymbol{\Phi}} \in \mathbb{R}^{r\times m}$ basis vectors for all path steps, respectively, and r is the number of path steps.

Note that using this method multiple trajectories, that is a library of the demonstrated movements that represents the same path, can be used to determine weights $\boldsymbol{w}$. For the details of parametric description of singularity-free representations of orientation, such as rotation matrices and quaternions, see ref. [33].

2.4. Motion constraints

The motion of a robot is generally constrained. The constraints are of two types: the *system constraints* imposed by the manipulator itself (due to the limits in joint torque, joint velocities, and/or joint positions) and the *task constraints* given by the task (path velocity and acceleration).[34] In the path tracking applications an operator moves the robot along a path by pushing it. As we have explained before, the forces generate the motion along the path. To prevent that maximal allowed joint or maximal task-space velocities are exceeded, the path velocity $\dot{s}$ has to be bounded.

Let us assume that the task space linear and rotational velocities are bounded

$$\|\dot{\boldsymbol{p}}\| < \dot{p}_{\max} \qquad \text{and} \qquad \|\boldsymbol{\omega}\| < \omega_{\max}\,, \tag{37}$$

where $\dot{p}_{\max}$ and $\omega_{\max}$ are the bounds. Using (30), the corresponding path velocity bounds are obtained as

$$\dot{s}^t_{\max} = \min\left(\frac{\dot{p}_{\max}}{\|\mathbf{J}_{sp}\|}, \frac{\omega_{\max}}{\|\mathbf{J}_{so}\|}\right)\,, \tag{38}$$

where $\mathbf{J}_{sp}$ and $\mathbf{J}_{so}$ are the position and orientation part of the $\mathbf{J}_s$, respectively.

In addition, $\dot{s}$ is bounded also due to bounded joint velocities. Let the joint velocity bounds be given as

$$|\dot{q}_i| \le \dot{q}_{\max,i}, \qquad i = 1, \ldots, n\,. \tag{39}$$

Considering only the task-space motion and using (4) and (30), the joint-wise bounds on $\dot{s}$ can be calculated as

$$\dot{s}^v_{\max,i} = \frac{\dot{q}_{\max,i}}{|\mathbf{J}^{\#}\mathbf{J}_s|_i}, \qquad i = 1, \ldots, n\,, \tag{40}$$

where $|\mathbf{J}^{\#}\mathbf{J}_s|_i$ denotes the absolute value of the i-th component of $\mathbf{J}^{\#}\mathbf{J}_s$, which are then combined to

$$\dot{s}^v_{\max} = \min_{i=1,\ldots,n} \dot{s}^v_{\max,i}\,. \tag{41}$$

Finally, the overall bound on $\dot{s}$ is defined as

$$|\dot{s}| \le \dot{s}_{\max} = \min\left(\dot{s}^v_{\max}, \dot{s}^t_{\max}\right)\,. \tag{42}$$

https://doi.org/10.1017/S0263574719000973 Published online by Cambridge University Press

Note that in the case of redundant robots, $\dot{s}^{y}_{\max}$ depends on the configuration of the robot and cannot be defined for a given path in advance.

3. Virtual Guides

In co-manipulation tasks, the operator is holding a tool attached to the end-effector of the robot and manipulating it to perform a task. The virtual guides assist a human operator to perform that task by regulating the motion of the operator. This means that the operator has the responsibility to generate the motion, while the robot monitors the motion and reacts if necessary regarding the preplanned trajectories or regions where the tool is. For that, the robot has enough information regarding the task path and constraint geometry, but it is not necessary that the robot has any temporal information on how to perform the task.

3.1. Unified constraint framework

By observing the admittance control algorithm (11) or (15), we can identify two sets of control design parameters and variables. The first set is associated with the dynamic behavior of the system which influences the feeling of the operator when moving the tool, and includes the virtual stiffness, damping, and inertia parameters. The other set is related to the motion and includes the desired values for the position, velocity, and acceleration, which are used to define the active constraints.

For example, let's consider a task where the robot guides an operator towards a point $\boldsymbol{p}_d$. Here, the position $\boldsymbol{p}_d$ represents the *point* constraint and a virtual guide generates a motion using $\boldsymbol{p}_d$ as the desired goal position. Note that in this situation operator is holding the tool. Therefore, it is important to properly select the gains. Typically, the gains $\mathbf{K}$, which define the force pulling the robot toward the target, have to be much lower as commonly used in position controlled robots. Also the gains $\mathbf{D}$ and $\mathbf{M}$, which define the dynamics of the motion, have to be selected properly to provide natural feeling for the operator.

The other possibility for moving towards a point would be that the robot only encourages the operator to move toward the goal position $\boldsymbol{p}_d$. In this case, the robot does not generate any motion, that is the stiffness gains $\mathbf{K}$ have to be set to 0. Here, the guiding action is achieved only by selecting proper values for gains $\mathbf{D}$ and $\mathbf{M}$. For example, by using anisotropic gains $\mathbf{D}$ the direction of motion towards the goal position can become more preferred as in the opposite direction.[18]

Another type of guides is the forbidden-region constraint, where the virtual guide does not influence the motion in a defined region but prevents the operator to leave that region. In most cases, such constraints are related to the positions or orientations and conform to the wall in a real world. Using the proposed unified framework, the forbidden-region constraints can be achieved by adding dead-zone functions for the relevant variables in Eq. (11) or (15). For example, let us design an active constraint to prevent the tool to leave the subspace $\mathcal{A}$ of the task space $\mathcal{T}$, where $y \in [y_{\min}, y_{\max}]$. First, we select the desired position in the y-direction to be somewhere between the bounds, for example in the middle of the $\mathcal{A}$

$$y_d = \frac{y_{\min} + y_{\max}}{2}\,. \tag{43}$$

Next, to "activate" the virtual guide only when the end-effector is on the border of $\mathcal{A}$, the corresponding component in the stiffness term $\boldsymbol{u}_s$ is given by

$$\boldsymbol{u}_s = \mathbf{K}\boldsymbol{e}_x, \tag{44}$$

and the control (11) or (15) is modified by applying a dead-zone to y component of $\boldsymbol{e}_x$ (see also Fig. 2)

$$\begin{aligned}(\boldsymbol{u}_s)^*_y &= \Big(\mathbf{Z}(\boldsymbol{e}_x, \mathbf{K}, \boldsymbol{e}_{\min}, \boldsymbol{e}_{\max})\Big)_y \\ &= \begin{cases} K^+_y \left(e_y - e_{y,\max}\right) & \text{for } e_y > e_{y,\max} = (y_d - y_{\min}) \\ 0 & \text{otherwise} \\ K^-_y \left(e_y - e_{y,\min}\right) & \text{for } e_y < e_{y,\min} = (y_d - y_{\max}), \end{cases}\end{aligned} \tag{45}$$

where $\mathbf{Z}(\cdot)$ represents the dead-zone function and e_y is the error in y-direction, $(\boldsymbol{e}_x)_y = \boldsymbol{p}_{y,d} - \boldsymbol{p}_y$. Consequently, the stiffness term $\boldsymbol{u}_s$ is 0 when the tool lies between $y_{\min}$ and $y_{\max}$, and when it leaves this region, a virtual spring will push the tool back into the allowed region. Note that using different

https://doi.org/10.1017/S0263574719000973 Published online by Cambridge University Press

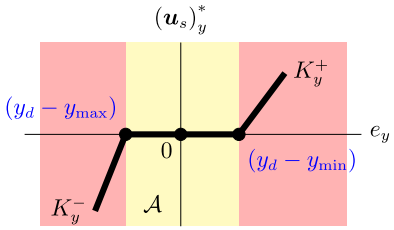

Fig. 2. Dead-zone representing forbidden-region constraint (yellow zone indicates the allowed region $\mathcal{A}$ and red zones indicate forbidden regions).

gains K_y^+ and K_y^- for each side of the dead-zone we can adjust the stiffness of the virtual borders. For hard guides high stiffness gains $\mathbf{K}$ are required, leaving the user with no or little freedom to deviate from the preferred path or to penetrate into forbidden region.

From (11) or (15), we can see that the stiffness term $\boldsymbol{u}_s$ can be treated as a virtual force proportional to the penetration depth. When the task scenario requires more than one forbidden zone or more complex zones, the stiffness terms can be governed by

$$\boldsymbol{u}_s = \sum_{i=1}^{r} \boldsymbol{u}_{si}. \tag{46}$$

When the region borders are not aligned with the task frame, then we propose to define the guides in a region dependent spaces $\mathcal{R}_i$, which are given by

$$\boldsymbol{u}_{si} = \overline{\mathbf{R}}_{\mathcal{R}_i} \mathbf{K}_i \overline{\mathbf{R}}_{\mathcal{R}_i}^T \boldsymbol{e}_{xi}\,, \tag{47}$$

where $\overline{\mathbf{R}}_{\mathcal{R}_i}$ is the rotation between $\mathcal{R}_i$ and $\mathcal{T}$. Note that the dead-zone is applied in the same manner as given by (45), but in the rotated space $\mathcal{R}_i$.

Considering (25), (45), (44), and (47) in (21) yields

$${}^t\boldsymbol{v}_c = \overline{\mathbf{R}}_t^T \boldsymbol{v}_d + \mathbf{D}^{-1} \overline{\mathbf{R}}_t^T \left(\sum_{i=1}^{r} \mathbf{Z} (\overline{\mathbf{R}}_{\mathcal{R}_i} \mathbf{K}_i \overline{\mathbf{R}}_{\mathcal{R}_i}^T \boldsymbol{e}_{xi}) + \boldsymbol{F}_{\text{ext}} + \frac{1}{\alpha} \mathbf{A}\mathbf{J} (\mathbf{I} - (\overline{\mathbf{R}}_t^T \mathbf{J})^{\#} \overline{\mathbf{R}}_t^T \mathbf{J}) \dot{\boldsymbol{q}}_n \right)\,, \tag{48}$$

which together with (22) represents the unified framework, that is combines the control of the dynamic behavior, selection of the tasks space and redundant DOFs, path tracking, and different active constraints. The framework can be summarized as follows. First, the space $\mathcal{T}$ relevant for the task has to be identified, that is the DOFs needed to perform the task. Next, the configuration of all constraints has to be determined. For most constraints this is the geometry which is defined in the space $\mathcal{T}$ or in a region-dependent space $\mathcal{R}$. Finally, the controller parameters are selected, which assure the desired dynamic behavior and generate active constraints.

The control actions (guidance, attraction, repulsion) are established by evaluating the robot configuration relatively to the constraint configuration. We have to select which states of the virtual robot are used for the constraining actions and the desired values for those states. For the regional constraints also the dead-zone parameters have to be defined. Finally, we define the gains $\mathbf{K}$, $\mathbf{D}$, and $\mathbf{M}$ to achieve the desired dynamic behavior.

3.2. Tracking tasks

The tracking tasks are tasks where the robot end-effector has to follow a predefined path. In the framework of human–robot cooperation these tasks can consist of several steps. Among many possible scenarios of robot assisting a human to track the predefined path, we have selected the one where the operator can move the tool by applying a force on the end-effector. When on the path, the robot guides the operator to follow the path, that is the operator controls the motion along the path and virtual guides prevent him from moving the robot off the path.

In this scenario, there are two situations, whether the end-effector is on the path or is not (see Fig. 3). So, a possible sequence could be that after the human operator grasps the end-effector, he/she has to move it first closer to the path, and when the target path is reached, the end-effector has to stay on the path. After the task is completed, the robot can be moved away from the path. The aim of the

https://doi.org/10.1017/S0263574719000973 Published online by Cambridge University Press

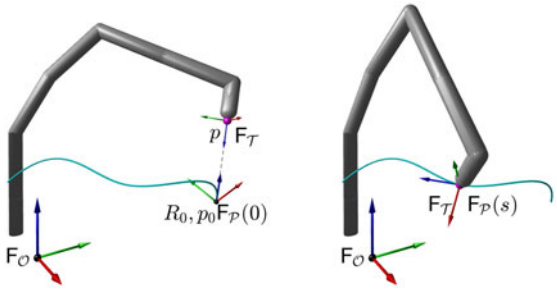

Fig. 3. Path tracking task: $F_{\mathcal{P}}(s)$ is coordinate frame attached to path, $F_{\mathcal{T}}$ is coordinate frame representing $\mathcal{T}$, $F_{\mathcal{O}}$ is world coordinate frame (a) Approaching path. (b) On path.

virtual guides is to reduce the cognitive and physical effort of the operator during the task execution. Since the requirements are different for both situations, that is when on the path or not, we design the virtual guides separately. Note that transitions between the states must be smooth.

In general, not all DOFs of the robot are needed to perform a motion in such scenario. The unused orientation DOFs can be utilized for some lower priority tasks like a pose optimization or obstacle avoidance. Let us assume that for general path tracking all spatial DOFs are needed. However, when the tool is away from the path, the orientation of the tool might not be important. Therefore, we propose to select virtual guides for the "off-path" state separately for positions and orientations.

3.2.1. Virtual guides for path tracking. The main goal of the virtual guides is to guide the operator along the path. On the path, the motion of the end-effector is fully constraint, that is the end-effector positions and orientations are as required by the path definition. The operator is only allowed to move along the path. Using the proposed unified framework, we implement the admittance control in the task space aligned with the path, and we have selected stiff behavior for off-path movements. The task space has been rotated so that one task frame axis is tangential to the path (frame $F_{\mathcal{T}}$ is aligned with frame $F_{\mathcal{P}}$) and then the controller (11) or (15) has been used with positional component of $\mathbf{K}$ in the path direction equal to 0. The other components of $\mathbf{K}$ have been selected according to the allowed deviation from the path. Also the damping and inertia related gains $\mathbf{D}$ and $\mathbf{M}$ have been selected differently for the direction aligned with the path tangent and other directions.

As a matter of fact, following the path is a one-dimensional task. The human operator can only influence the position s on the path $\boldsymbol{f}(s)$. Therefore, a one-dimensional admittance control defined in path space can be used with (29)–(31) to obtain the task space motion. Note that this is equivalent to a proxy-based approach.[11,18,35]

The dynamics of the proxy can be selected as

$$\ddot{s} = \frac{1}{m_s}\left(b_s(\dot{s}_d - \dot{s}) + k_s \mathbf{J}_s^T \boldsymbol{F}_{\text{ext}}\right), \tag{49}$$

where $\dot{s}_d$ is the desired path velocity, and m_s, b_s, and k_s are the proxy inertia, damping, and force scaling constant, respectively. The driving force for the proxy is the projection of the external force $\boldsymbol{F}_{\text{ext}}$ on the path tangent. The desired path velocity s_d is used when the path velocity profile is important and the operator is guided along the path using this velocity profile.

From the path kinematics (29)–(31) the task space motion ($\{\boldsymbol{p}, \mathcal{Q}\}$, $\boldsymbol{v}$, and $\dot{\boldsymbol{v}}$) is obtained and used in the controller of the virtual robot to get the robot motion in the joint space. When using proxy, the task space admittance control gains are selected so that stiff robot-proxy behavior is assured. Note that the motion of the proxy is constrained by the path, and therefore no other regional constraints are needed.

3.2.2. Constraints for approaching the path. Beside guiding the operator when moving along the path, the virtual guides can also assist the operator to get on the path. Since there might be some regions in the robot workspace which have to be avoided, that is due to obstacles, singular configurations, etc., the virtual guides can prevent the end-effector to enter them by selecting suitable regional constraints.

https://doi.org/10.1017/S0263574719000973 Published online by Cambridge University Press

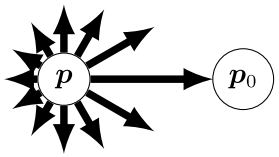

Fig. 4. Preferred motion directions ($\propto$ to arrow length) when approaching the path initial position $\boldsymbol{p}_0$ ($\boldsymbol{p}$ is the actual end-effector position)

If there are no obstacles or other forbidden regions in the workspace, the positions in the "off-path" are not constrained. This means that the operator can freely move the end-effector. Such behavior is achieved when the stiffness for the positions is set to zero, $K_i = 0$, $i = \{x, y, z\}$. To encourage the operator to move toward the initial path position $\boldsymbol{p}_0$, we propose anisotropic damping where the damping in the direction toward $\boldsymbol{p}_0$ is lower than in other directions. This can be achieved by rotating the position task space so that one task frame axis is pointing toward $\boldsymbol{p}_0$. Then, lower damping gains are selected when the end-effector is moving toward $\boldsymbol{p}_0$. For example (see also Fig. 3a), if we select $\mathbf{R}_p$ in (23) to be the rotation matrix that rotates z-axis of frame $\mathrm{F}_{\mathcal{O}}$ onto the vector $(\boldsymbol{p}_0 - \boldsymbol{p})$, where $\boldsymbol{p}$ is the actual end-effector position. Then, the damping gains of the admittance controller realized in $\mathcal{T}$ are defined as

$$B_x = B_y = b_0\,, \qquad B_z = \begin{cases} b_2 & \text{for } \dot{z} < 0 \\ b_1 & \text{for } \dot{z} > 0 \text{ (approaching } \boldsymbol{p}_0) \end{cases}, \qquad b_1 < b_0 < b_2. \tag{50}$$

Such gains give the preferred movement directions as shown in Fig. 4. Note that during the motion the task frame $\mathrm{F}_{\mathcal{T}}$ can rotate, but the controller gains are constant.

Usually the end-effector orientation is not important when "off-path" and the operator can change the orientation. To exploit the orientation DOFs for secondary tasks, we select $\mathbf{S}$ as

$$\mathbf{S} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}. \tag{51}$$

However, as the orientation is prescribed on path, we have to assure that when approaching the path initial position $\boldsymbol{p}_0$ the final orientation equals the required path orientation. For that, we apply a symmetric distance-dependent dead-zone (45) for orientation errors with borders defined as

$$(\boldsymbol{e}_x)_{i,\max} = -(\boldsymbol{e}_x)_{i,\min} = \kappa_i \|\boldsymbol{p}_d - \boldsymbol{p}\|\,, \quad i = \{\alpha, \beta, \gamma\}, \tag{52}$$

where the gains κ_i are selected so that the active orientation correcting motion initiated by virtual guides will not disturb the operator. In cases when the borders are not equal for all orientations, we have to select suitable rotation matrix $\mathbf{R}_o$. Otherwise, we can use any $\mathbf{R}_o$, for example $\mathbf{R}_o = \mathbf{I}$ or $\mathbf{R}_o = \mathbf{R}_p$.

If there is a more complex forbidden region in the robot workspace when approaching the path, it is necessary to add to the above defined control additional constraints that will prevent the end-effector positions to leave the allowed region. These constraints are realized as positional stiffness terms with dead-zones. For example, in Fig. 5 three planes represent walls of forbidden regions. Walls A and B are aligned with the frame $\mathrm{F}_{\mathcal{O}}$ and wall C is aligned with frame $\mathrm{F}_{\mathcal{R}}$. To prevent entering into the all forbidden zones, we have to define the positional stiffness term $\boldsymbol{u}_s$ as

$$\begin{aligned} \boldsymbol{u}_s = {} & \mathbf{Z}(\boldsymbol{p}_d - \boldsymbol{p}, K_{AB}, \boldsymbol{e}_{AB,\min}, \boldsymbol{e}_{AB,\max}) + \\ & \mathbf{Z}(\boldsymbol{p}_d - \boldsymbol{p}, K_C, \boldsymbol{e}_{C,\min}, \boldsymbol{e}_{C,\max}) \end{aligned} \tag{53}$$

with

$$\begin{aligned} K_{AB} &= \mathrm{diag}([K_A, 0, K_B]) \\ \boldsymbol{e}_{AB,\min} &= [-d_A, \infty, -d_B] \\ \boldsymbol{e}_{AB,\max} &= [\infty, \infty, \infty] \end{aligned}$$

https://doi.org/10.1017/S0263574719000973 Published online by Cambridge University Press

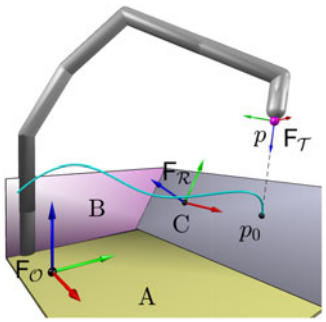

Fig. 5. Path tracking task: Approaching path where allowable workspace is bounded by three planes

$$K_C = \mathbf{R}_{\mathcal{R}} \,\mathrm{diag}([0,\, 0,\, K_C])\mathbf{R}_{\mathcal{R}}^T$$
$$\boldsymbol{e}_{C,\min} = [0,\, \infty,\, -d_C]$$
$$\boldsymbol{e}_{C,\max} = [\infty,\, \infty,\, \infty],$$

where $\mathbf{R}_{\mathcal{R}}$ is representing the rotation between $\mathrm{F}_{\mathcal{R}}$ and $\mathrm{F}_{\mathcal{O}}$, and d_A, d_B, and d_C are the distances of $\boldsymbol{p}_0$ to planes A, B, and C, respectively.

4. Experimental Evaluation

To illustrate the performance of the virtual guides based on the unified control framework we have selected the Buzz-Wire task, where a ring has to be moved along curving wire without touching it, which is a typical path following tasks that requires a certain level of cognitive and physical effort from the human for achieving the goal. To reduce the effort of the operator, we have used a KUKA LWR robot arm with 7DOF to assist the operator while performing the task.

Although the KUKA LWR arm has joint torque sensors, we have equipped it with additional force/torque sensor mounted on the end-effector, which enables more precise force/torque measurements at the end-effector. For both tasks, the tool has been mounted on the force sensor. To perform the task, the operator grasps the tool and moves it.

For the experiments, we have implemented the virtual guides using admittance control at the velocity level (21). The selection of the controller gains, dead-zones, and task space rotations is defined below for each task.

In general, path guiding constraints consider all spatial position and orientation DOFs. For the Buzz-wire task, the ring and the wire are axisymmetric, and the corresponding rotational DOF is not relevant for the task. Additionally, in some situations not all spatial directions have to be controlled, that is the orientation of the tool when far from the target path is not important. To optimize the behavior of the system, all functional redundant DOFs can be removed from the task-space and used for the self-motion, which is used for optimization of some performance index.

For demonstrations, we have used the functional redundancy together with intrinsic redundancy of the KUKA LWR robot to optimize the robot configuration. Based on the Jacobian transpose formulation, we propose to use the following simple null-space velocity control defined as

$$\dot{\boldsymbol{q}}_{vn} = \mathbf{K}_1(\boldsymbol{q}_{\mathrm{opt}} - \boldsymbol{q}_v) - \mathbf{K}_2\,\mathbf{J}^T\boldsymbol{F}_{\mathrm{body}}\,, \tag{54}$$

where $\boldsymbol{q}_{\mathrm{opt}}$ is the optimal joint configuration and $\boldsymbol{F}_{\mathrm{body}}$ are external forces acting on the body of the robot and are provided by the KUKA LWR controller using the internal joint torque sensors. $\mathbf{K_1}$ and $\mathbf{K_2}$ are suitable gain matrices. The aim of the first term is to reconfigure the robot into a preselected configuration $\boldsymbol{q}_{\mathrm{opt}}$ so that the motion along the wire is possible without reaching the joint limits. The second term allows the operator to manually reconfigure the robot. The functional redundancy resulting from unconstrained orientations has been resolved using the concept of virtual forces as defined by (27) and (28).

Note that in our experiments the operator can only influence the position of the end-effector. The orientation is always actively controlled by the robot controller. Note that the orientation is defined by the path; however, in some situations when it is not important for the task, the orientation DOFs have been exploited for the secondary task. In this case, the operator can change the orientation.

https://doi.org/10.1017/S0263574719000973 Published online by Cambridge University Press

Fig. 6. Buzz-wire task: ring is moving along the wire.

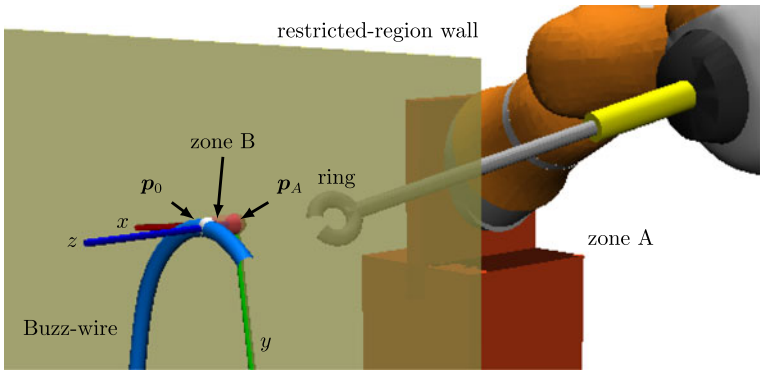

Fig. 7. Buzz-wire task: free-motion zone A, restricted region wall, and approach zone B.

The goal of the Buzz-wire task is to move the ring along the wire without touching it. The operator has to get the right balance between speed and skill in order to optimally finish the task. The ring has to be moved so that the wire is always almost in the middle of the ring, and the ring orientation has to be perpendicular to the wire (see Fig. 6). Note that the orientation of the ring regarding the wire is not important for the task, that is the ring can be freely rotated around the wire. For clearness of virtual guides definition, the wire has been designed to be in a plane. In our experiment we have modified some rules of the typical Buzz-wire game. The ring, which is originally already on the wire, has to be put in our task on the wire at the path starting point. Next, we have allowed that the ring can be removed from the wire only at this point. Note that such a task definition is solely to illustrate the capabilities of the proposed framework. The virtual guides could be easily defined to allow the operator to put the ring on the wire or to remove the ring as any location on the wire. To illustrate the use of orientation guides, we have required a fixed orientation to put the ring on the wire. As the wire path could not be described with analytic functions, the path has been encoded using the RBFs.

For the minimal operator's effort, the dynamic parameters have been selected to be: damping $D = 20$ Ns/m, stiffness $K = 2000$ Ns/m, and force filter cutoff frequency $D/M = 400\ \mathrm{s}^{-1}$.

The virtual guides have been defined depending on the distance between the ring and the path starting point on the wire. Additionally, as the ring has to be put on the wire, the orientation of the tool has to be considered in this phase of the task. We have defined that the ring could be put on the wire or removed from the wire only at the path starting point. Therefore, we have defined three zones: (A) far-away zone (distance to the path starting point $\leq$0.02 m), (B) approaching zone (distance to the path starting point $<$0.02 m), and (C) on-the-wire zone.

In zone A, the position of the tool has not been restricted and the operator could freely move the tool. Due to the robot workspace limitations, the tool could be moved only in the space between the wire and the robot. Therefore, we have defined a restricted region for the tool position. We have used the controller (45), where the task space has been selected so that the z-axis has been the normal to the plane of the wire, and we have defined a virtual wall parallel to the plane going through the position $\boldsymbol{p}_A$. Point $\boldsymbol{p}_A$ has been used as the approaching point to put the ring on the wire (see Fig. 7) and has been selected near the path initial position $\boldsymbol{p}_0$, $\boldsymbol{p}_A = \boldsymbol{p}_0 - [0.02, 0, 0]^T$m. For the zone A, the desired position has been the approaching point, $\boldsymbol{p}_d = \boldsymbol{p}_A$. The position gains in x and y directions have been set to 0 (no position control, only damping), and to prevent entering the

https://doi.org/10.1017/S0263574719000973 Published online by Cambridge University Press

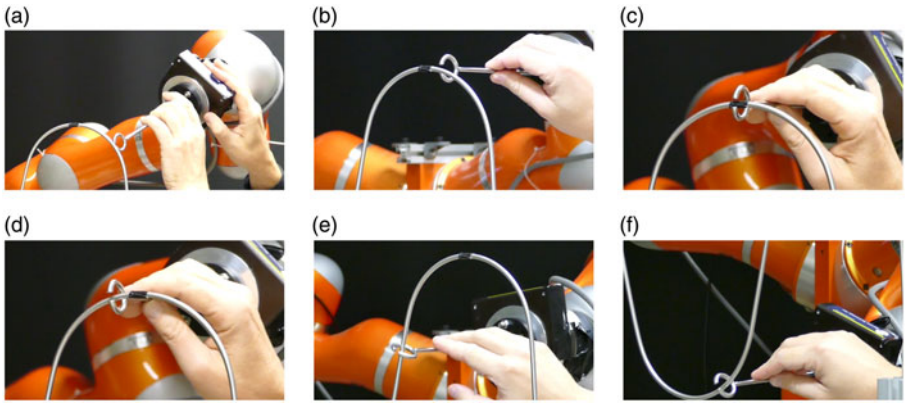

Fig. 8. Buzz-wire experiment: (a) Far away from the wire (zone A). (b) Near position $\boldsymbol{p}_A$ (zone B). (c) Close to the wire – ring is wrapped around the wire (transition from zone B to wire). (d), (e), and (f) Ring is moved along the wire.

forbidden zone, the restricted region wall has been defined by a dead-zone for position in z-direction with borders $[0, \infty]$. The orientation of the tool in zone A has been important only when approaching the point $\boldsymbol{p}_A$. So, the orientation could be arbitrary in a certain range depending on the distance to the $\boldsymbol{p}_A$. This has been achieved by selecting dead-zones for all orientation DOFs with bounds $[-d, d]$, $d = k_\varphi \|\boldsymbol{p}_d - \boldsymbol{p}\|$, where k_φ defined the range for the allowed orientations when approaching $\boldsymbol{p}_A$.

To enable putting the ring on the wire, the motion in zone B has to be restricted. Therefore, we have defined the constraint which allowed only a motion along straight line from $\boldsymbol{p}_A$ to $\boldsymbol{p}_0$ with the orientation of the tool as required in $\boldsymbol{p}_0$. This has been achieved by selecting the position and orientation gains $\mathbf{K} = 100\,\mathrm{diag}([0, 1, 1, 1, 1, 1])$. In zone B no dead-zones have been defined.

When the ring is on the wire we can use two types of virtual guides. One possibility is to rotate the task space so that x-axis is pointing in the direction of the task path and to set only the positional gain for x direction to 0 and no dead-zones have been used. In this way, the operator could move only in the direction of the path. The other possibility is to use a proxy as defined by (49). In this case, we do not use admittance control, but only the path tracking control, where the robot end-effector follows the proxy. The motion of the proxy is constrained to the path and controlled by the operator's interaction force. As the ring and the wire are axisymmetric, the orientation of the tool around wire axis is arbitrary, when the tool is on the wire. To exploit this free orientation for the secondary task, we have used the dead-zone for the orientation around x axis with bounds $[-a, a]$, $a = k_\alpha s$, where k_α defined the range for the allowed orientations. Such a dead-zone has assured the required orientation of the tool at the path starting point ($s = 0$).

The admittance control has enabled such dynamic properties of the virtual robot so that the operator could easily move the robot with low applied forces. The proposed virtual guides have supported the operator to follow the wire without touching it. Thus, his effort to follow the wire has been significantly reduced. In Fig. 8, some typical situations during task execution are shown. The details of the experiment are also shown in the first multimedia attachment.

When the operator has tried to move the ring along the wire very fast, he has reached the maximal allowed path velocity. The operator has felt the inability to move faster as an increased friction. This situation is shown in Fig. 9 where we can see how the bounds on $\dot{s}$ due to the bounded joint velocity are changing along the path. One would expect that the maximal joint velocities would not be exceeded during the motion. However, as the system is intrinsic and functional redundant and the null-space motion is not considered in the calculation of $\dot{s}_{\max}$, joint velocity bounds can be exceeded.

5. Conclusion

In this paper, we introduced the unified virtual guides framework for the physical human–robot cooperative setups. We showed that the proposed framework can act as an effective tool to assist operators to perform a task where an accurate motion or heavy load manipulation is required and/or where due to the unstructured environment human cognitive skills are needed. The proposed unified virtual guides framework is based on admittance control applied to a virtual robot, which greatly increases the usability since it can be used on any position controlled robot with mounted force/torque sensor

https://doi.org/10.1017/S0263574719000973 Published online by Cambridge University Press

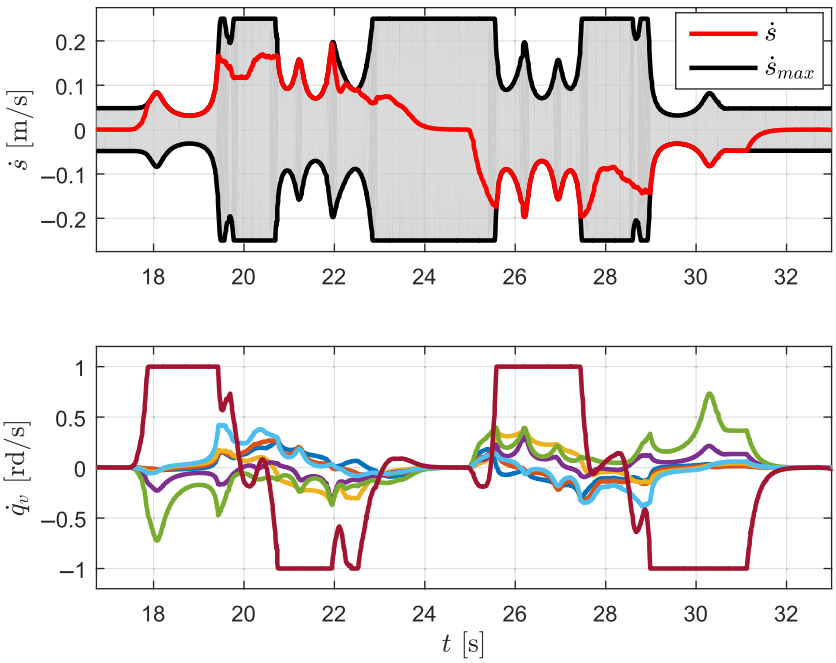

Fig. 9. Time response of a motion along the wire; velocity bounds $\dot{s}^t_{\max} = 0.25$ m/s and $\dot{q}_{\max,i} = 1$ rd/s. Top plot: path velocity $\dot{s}$ and allowable path velocity region (gray); Bottom plot: joint velocities $\dot{\boldsymbol{q}}_v$.

at the end-effector. Note that the proposed approach does not need internal force/torque sensor in the joints of the robot. The admittance control can be defined at the acceleration or velocity level. Additionally, the proposed framework allows on-line changing of the task space and selection of redundant DOFs. The proposed redundancy resolution allows to limit the extend of the self-motion and smooth motion when the number of redundant DOFs changes. As the control at the velocity level is simpler than the control at acceleration level, we favor to use the first one. Additionally, we showed how the novel methodology is used to design different types of virtual constraints and the dynamic behavior of the system when moving on the path. The proposed framework was evaluated on a KUKA LWR robot assisting a human to do different path following task. Experimental results have shown that different virtual constants with changing properties can be easily defined by assigning corresponding values for parameters and input variables of the control algorithm.

Acknowledgments

This work was supported by EU Horizon 2020 Programme grant 820767, CoLLaboratE, by Slovenian Research Agency grant P2-0076 and by GOSTOP programme C3330-16-529000 co-financed by Slovenia and EU under ERDF.

Supplementary Material

To view the supplementary material for this article, please visit https://doi.org/10.1017/S0263574719000973.

References

1. A. De Santis, B. Siciliano, A. De Luca and A. Bicchi, "An atlas of physical human–robot interaction," *Mech. Mach. Theory* **43**(3), 253–270 (2008).
2. S. Haddadin and E. Croft, "Physical Human–Robot Interaction," **In:** *Springer Handbook of Robotics* (B. Siciliano B and O. Khatib, eds.) (Springer International Publishing, Cham, 2016) pp. 1835–1874.
3. S. Ikemoto, H. B. Amor, T. Minato, B. Jung and H. Ishiguro, "Physical human–robot interaction: mutual learning and adaptation," *IEEE Robot. Auto. Mag.* **19**(4), 24–35 (2012).
4. B. Nemec, N. Likar, A. Gams and A. Ude, "Human robot cooperation with compliance adaptation along the motion trajectory," *Auto. Robot.* **42**(5), 1023–1035 (2017).
5. A. Mörtl, M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan and S. Hirche, "The role of roles: Physical cooperation between humans and robots," *Int. J. Robot. Res.* **31**(13), 1656–1674 (2012).

6. T. Petrič, R. Goljat and J. Babič, "Cooperative Human–Robot Control Based on Fitts' Law," *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Cancun, Mexico (IEEE, 2016) pp. 345–350.
7. L. Peternel, T. Petrič, E. Oztop and J. Babič, "Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach," *Auto. Robot.* **36**(1–2), 123–136 (2014).
8. A. Bicchi, M. A. Peshkin and J. E. Colgate, "Safety for Physical Human–Robot Interaction," **In:** *Springer Handbook of Robotics* (B. Siciliano B and O. Khatib, eds.) (Springer, Berlin, Heidelberg, 2008) pp. 1335–1348.
9. T. Petrič, M. Cevzar and J. Babič, "Utilizing Speed-Accuracy Trade-Off Models for Human–Robot Coadaptation During Cooperative Groove Fitting Task," *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, Birmingham, UK (2017) pp. 107–112.
10. L. Rosenberg, "Virtual Fixtures: Perceptual Tools for Telerobotic Manipulation," *Proceedings of IEEE Virtual Reality Annual International Symposium*, Seattle, WA, USA (1993) pp. 76–82.
11. S. A. Bowyer, B. L. Davies and F. Rodriguez y Baena, "Active constraints/virtual fixtures: A survey," *IEEE Trans. Robot.* **30**, 138–157 (2014).
12. A. Bettini, P. Marayong, S. Lang, A. Okamura and G. D. Hager, "Visual assisted control for manipulation using virtual fixtures," *IEEE Trans. Robot.* **20**(6), 953–966 (2004).
13. J. J. Abbott, P. Marayong and A. M. Okamura, "Haptic Virtual Fixtures for Robot-Assisted Manipulation," **In:** *Robotics Research. Springer Tracts in Advanced Robotics* (S. Thrun, R. Brooks and H. Durrant-Whyte, eds.), vol 28 (Springer, Berlin, Heidelberg, 2007) pp. 49–64.
14. G. D. Hager, "Human–Machine Cooperative Manipulation with Vision-Based Motion Constraints," **In:** *Lecture Notes in Control and Information Sciences Visual Servoing via Advanced Numerical Methods*, (G. Chesi and K. Hashimoto, eds.), vol. 401 (Springer, London, 2010) pp. 55–70.
15. S. S. Restrepo, Intuitive, Iterative and Assisted Virtual Guides Programming for Human–Robot Comanipulation, *PhD thesis* (2018).
16. G. Raiola, X. Lamy and F. Stulp, "Co-manipulation with Multiple Probabilistic Virtual Guides," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2015, Hamburg, Germany (2015) pp. 7–13.
17. D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Auto. Robot.* **31**(2–3), 115–131 (2011).
18. Z. Pezzementi, A. M. Okamura and G. D. Hager, "Dynamic Guidance with Pseudoadmittance Virtual Fixtures," *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Roma, Italy (IEEE, 2007) pp. 1761–1767.
19. D. Kragic, P. Marayong, M. Li, A. M. Okamura and G. D. Hager, "Human–Machine collaborative systems for microsurgical applications," *Int. J. Robot. Res.* **24**(9), 731–741 (2005).
20. S. S. Restrepo, G. Raiola, P. Chevalier, X. Lamy and D. Sidobre, "Iterative Virtual Guides Programming for Human–Robot Comanipulation," *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Munich, Germany (IEEE, 2017) pp. 219–226
21. G. Raiola, S. S. Restrepo, P. Chevalier, P. Rodriguez-Ayerbe, X. Lamy, S. Tliba and F. Stulp, "Co-manipulation with a library of virtual guiding fixtures," *Autonomous Robots* **42**(5), 1037–1051 (2017).
22. B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, *Robotics - Modelling, Planning and Control* (Springer-Verlag, London, 2009).
23. C. Ott, R. Mukherjee and Y. Nakamura, "Unified Impedance and Admittance Control," *2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK, USA (2010) pp. 554–561.
24. V. Duchaine, B. Mayer St-Onge, D. Gao and C. Gosselin, "Stable and intuitive control of an intelligent assist device," *IEEE Trans. Haptics* **5**(2), 148–159 (2012).
25. F. Ficuciello, L. Villani and B. Siciliano, "Variable impedance control of redundant manipulators for intuitive Human–Robot Physical Interaction," *IEEE Trans. Robot.* **31**, 850–863 (2015).
26. I. Ranatunga, F. Lewis, D. O. Popa and S. M. Tousif, "Adaptive admittance control for Human–Robot interaction using model reference design and adaptive inverse filtering," *IEEE Trans. Control Systems Technology* **25**(1), 1–10 (2017).
27. L. Žlajpah and T. Petrič, "Virtual Guides for Redundant Robots Using Admittance Control for Path Tracking Tasks," **In:** *Advances in Service and Industrial Robotics: Proceedings of the 27th International Conference on Robotics in Alpe–Adria Danube Region (RAAD 2018)* (N. A. Aspragathos, P. N. Koustoumpardis and V. C. Moulianitis, eds.) (Springer International Publishing, Cham, 2019) pp. 13–23.
28. A. Albu-Schaffer, C. Ott and G. Hirzinger, "A unified passivity-based control framework for position, torque and impedance control of flexible joint robots," *Int. J. Robot. Res.* **26**, 23–39 (2007).
29. J. Nakanishi, R. Cory, M. Mistry, S. Schaal and J. Peters, "Operational space control: A theoretical and empirical comparison," *Int. J. Robot. Res.* **27**, 737–757 (2008).
30. V. Duchaine and C. M. Gosselin, "General Model of Human–Robot Cooperation Using a Novel Velocity Based Variable Impedance Control," *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*, Tsukaba, Japan (2007) pp. 445–451.
31. A. Lecours, B. Mayer-St-Onge and C. Gosselin, "Variable Admittance Control of a Four-Degree-of-Freedom Intelligent Assist Device," *2012 IEEE International Conference on Robotics and Automation*, vol. 2, Saint Paul, MN, USA (2012) pp. 3903–3908.

32. L. Žlajpah, "Kinematic Control of Redundant Robots in Changing Task Space," **In:** *Advances in Robot Design and Intelligent Control. RAAD 2016. Advances in Intelligent Systems and Computing* (A. Rodić and T. Borangiu, eds.), vol. 540 (Springer, Cham) (2016) pp. 3–11.
33. A. Ude, B. Nemec, T. Petri and J. Morimoto, "Orientation in Cartesian Space Dynamic Movement Primitives," *2014 IEEE International Conference on Robotics and Automation (ICRA)* Hong Kong, China (2014) pp. 2997–3004.
34. L. Žlajpah, "On Time Optimal Path Control of Manipulators with Bounded Joint Velocities and Torques," *IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota (1996) pp. 1572–1577.
35. M. Mihelj and J. Podobnik, "Virtual Fixtures," **In:** *Haptics for Virtual Reality and Teleoperation* (M. Mihelj and J. Podobnik, eds.) (Springer, Netherlands, 2012) pp. 179–200.

https://doi.org/10.1017/S0263574719000973 Published online by Cambridge University Press