TECHNISCHE UNIVERSITÄT WIEN

AUTOMATION & CONTROL INSTITUTE
INSTITUT FÜR AUTOMATISIERUNGS- & REGELUNGSTECHNIK

Intuitive Robot Programming and Physical Human-Robot Interaction

DIPLOMA THESIS

Conducted in partial fulfillment of the requirements for the degree of a

Diplom-Ingenieur (Dipl.-Ing.)

supervised by

Univ.-Prof. Dr. techn. A. Kugi
Dr. techn. C. Hartl-Nesic

submitted at the

TU Wien
Faculty of Electrical Engineering and Information Technology
Automation and Control Institute

by
Elias Pritzi
Matriculation number 01607436

Vienna, in September 2022

Complex Dynamical Systems Group
A-1040 Wien, Gußhausstr. 27–29, Internet: https://www.acin.tuwien.ac.at

Preamble

First and foremost, I would like to express my deep gratitude to my supervisor, Dr. techn. Christian Hartl-Nesic, for his continuous inspiration and guidance. The completion of this diploma thesis would not have been possible without his support, be it for the theoretical knowledge in the field of robotics as well as for the practical support during the execution of the experiments.

I would like to especially thank Univ.-Prof. Dr. techn. A. Kugi for the opportunity to conduct my diploma thesis under his supervision and for giving me access to the robot laboratory. His interesting lectures were what made me pursue this particular degree in the first place.

I am also very grateful for the continuous encouragement, motivation and support provided by my family. My family's support made my studies possible at all.

Last but not least, I would like to thank all the fellow students that assisted me during all my years of studying, whether during preparations for exams, during laboratory exercises or during various on- or off-topic discussions. In particular Felix Heidegger, Lukas Flatz and Georg Feiler are to be thanked for countless professional and personal exchanges of ideas.

Vienna, in September 2022

Abstract

Recent trends in the manufacturing industry require more flexible and customizable production sites to satisfy the demand for an increasing product diversity. To cope with this shift, manufacturing companies increasingly apply collaborative robots, since they are more versatile and are easier to adjust to new tasks compared to traditional industrial robots. Collaborative robots especially offer opportunities for small- to medium-sized enterprises, because they are designed for direct human-robot interaction and thus programmable without the need for highly sophisticated programming skills. Nevertheless, the development of intuitive interaction modes is still an active field of research to which the presented work contributes.

In this work, two novel user interaction modes for human-robot collaboration (HRC) are introduced, called *Path Snap-In* and *Path Switch*. These modes are based on path-following control and address multitask scenarios. The human operator decides online which task to perform by physically interacting with the robot.

During *Path Snap-In*, the robot can snap in and out of the tasks by pushing it towards or away from predefined paths. This is achieved by simultaneously computing multiple path-following controllers and weighting their individual control output with a distance-dependent weighting factor. The so-called *Orientation Snap* is introduced as a special case of *Path Snap-In*, in which only the orientation of the end-effector snaps in and out of predefined directions while the robot is freely movable otherwise.

During *Path Switch*, the robot is always controlled by one active path-following controller. The operator may change the active task by pushing the robot towards the path of the desired task. An underlying belief system estimates the human intent by accumulating the externally applied force. Once a desired task change is detected, an online trajectory generator generates a trajectory to transition to the new path.

The presented concepts are validated in a collaborative drilling scenario on an experimental setup with the KUKA LBR iiwa 14 R820.

Kurzzusammenfassung

Jüngste Trends in der Fertigungsindustrie erfordern flexible und anpassbare Produktionsstätten, um der Nachfrage nach einer zunehmenden Produktvielfalt nachzukommen. Um diesen Wandel zu bewältigen, setzen Fertigungsunternehmen vermehrt kollaborative Roboter ein, da diese im Vergleich zu traditionellen Industrierobotern vielseitiger sind und sich einfacher an neue Aufgaben anpassen lassen. Kollaborative Roboter bieten vor allem für kleine und mittelständische Unternehmen neue Möglichkeiten, da sie speziell für die Interaktion zwischen Mensch und Roboter konzipiert sind und daher ohne anspruchsvolle Programmierkenntnisse in Betrieb genommen werden können. Die Entwicklung von intuitiven Interaktionsmodi ist ein aktives Forschungsgebiet, zu welchem diese Arbeit einen Beitrag leistet.

In dieser Arbeit werden zwei neue Benutzerinteraktionsmodi für die Mensch-Roboter Kollaboration (HRC) vorgestellt, *Path Snap-In* und *Path Switch* genannt. Diese basieren auf der Pfadfolgeregelung und behandeln Szenarien mit mehreren Aufgaben und zugehörigen Pfaden. Die Bediener_in entscheidet während der Ausführung durch physische Interaktion mit dem Roboter, welche Aufgabe ausgeführt werden soll.

Während des *Path Snap-In* kann der Roboter auf bestimmten Pfaden einrasten oder sich davon lösen, indem er zu den Pfaden hin- oder weggedrückt wird. Dies wird durch eine simultane Berechnung von mehreren Pfadfolgereglern erreicht, dessen Stellgrößen über distanzabhängige Gewichtungsfaktoren gewichtet werden. Als spezieller Fall des *Path Snap-In* wird ein sogenannter *Orientation Snap* eingeführt, bei welchem nur die Orientierung des Endeffektors in vordefinierte Richtungen einrastet bzw. sich davon löst, während der Roboter ansonsten frei beweglich bleibt.

Während des *Path Switch* wird der Roboter immer durch einen aktiven Pfadfolgeregler geregelt. Die Bediener_in kann die aktive Aufgabe ändern, indem der Roboter in Richtung des Pfades der gewünschten Aufgabe gedrückt wird. Ein zugrundeliegendes Belief-System schätzt die menschliche Absicht durch Akkumulation der extern eingebrachten Kraft. Sobald ein gewünschter Wechsel der Aufgabe erkannt wird, generiert ein Online-Trajektoriengenerator eine Trajektorie für den Übergang zum neuen Pfad.

Die vorgestellten Konzepte werden in einer kollaborativen Bohranwendung an einem experimentellen Aufbau mit dem KUKA LBR iiwa 14 R820 validiert.

Contents

1 Introduction

In the past, the field of robotics mainly focussed on the development of traditional industrial robots, which are now widely used in manufacturing facilities for high-volume production. Although these robots require fence guardings and additional safety measures, the average robot density in the global manufacturing industry reached 126 robots per 10,000 employees in 2020 [1]. Recent trends in the manufacturing industry demand more customizability and shorter lead time to produce a larger variety of products. Especially small- and medium-sized companies follow this demand, and thus, increasingly adopt collaborative robots into their manufacturing processes [1, 2]. Collaborative robots, often abbreviated as *cobots*, are designed for direct interaction between a human operator and the robot [3]. With the employment of cobots, the work load can be divided between the human and the robot and no safety fence guards are needed. One of the main difficulties in the adoption of human-robot collaboration (HRC) is the development of intuitive interaction modes. When sophistacted interaction modes are implemented, providing inputs to the robot as well as estimating the robots intention (to interact with the robot, the human operator must be able to anticipate its motion) must be straightforward [4]. The communication with the robot should not distract the human operator from the task at hand.

1.1 Industrial Processes with Cobots

Currently, most cobots in the manufacturing industry deal with simple tasks like handling or assembling objects [4]. The use of cobots in these tasks reduces the human work load and, additionally, allows the manipulation of objects which are not suitable for a human, e. g., due to safety or hygienic reasons [5, 6]. While the mentioned tasks may also be performed by traditional industrial robots, the integration of cobots results in shorter product lifecycles, reduced time-to-market and increased flexibility [7]. However, in material handling and assembling tasks, the robot is used as a tool and not as a collaborative workmate. In these applications, the full potential of collaborative robots is not utilized. To differentiate the different levels of human-robot interaction (HRI), various classifications are found in the literature. Most of them distinguish at least three categories [8]:

- *Human-Robot Coexistence* describes the sharing of a common workspace between the human and the robot without a common task. The main goal is avoiding collisions, while both parties focus on their individual tasks.
- *Human-Robot Cooperation* describes the joint work of the human and the robot to fulfill a common task, while both parties deal with individual subtasks. In this

scenario, the robot may, e. g., fetch parts from the storage and provide them to the human, who subsequently assembles these parts to a final product.

- *Human-Robot Collaboration (HRC)* describes the performance of a complex task by the human and the robot together. For example, the joint operation of a two-person cross-cut saw falls in this category [9].

To make use of the full potential of collaborative robots, a lot of research is put into the category *Human-Robot Collaboration*. In a collaborative application, the preciseness and strength of the robot is combined with the cognitive awareness of the human operator, which enables the execution of complex tasks. One of the most common applications of industrial robots is welding. Difficulties in performing welding processes with robots arise in real working conditions, e. g., due to distortions induced by heat [10]. A skilled human welder can adapt to these difficulties, but the robot generally can not. In a human-robot collaborative welding application, the human operator may adapt the robot behavior to the real working conditions and thus improve the task execution. This applies in a similar way to other robotic operations, such as spray painting [11], milling [12], cutting, polishing, deburring or drilling [13].

These applications are suitable for HRC and share one specific property. Each task defines a geometric path along which the robot or the robotic tool should interact with the workpiece. The movement along this geometric path is not tied to a predefined timing information, since this timing information depends on real working conditions. This problem is often considered as *path following control* [14]. In path following control, the movement of the robot endeffector along the path and the movement transversal to the path are controlled separately. The main objective is to move and stay on the path, while the secondary objective is to perform a desired motion along the path. With the application of transversal feedback linearization, a rigid-body robotic system transforms into a linear system in the path coordinates [15]. With this transformation, common control strategies like force control, impedance control or admittance control are applicable in these path coordinates.

Industrial processes are often described by a desired motion of a robotic tool along an application-specific path. The motion of the tool is expressed w.r.t. a virtual reference point specific to a robotic tool. This virtual reference point is called tool center point (TCP) and may be, e. g., the tip of a milling head. The TCP of a robotic tool is described as a translation and orientation relative to the robot flange.

In applications, where human and robot work collaboratively, the robot should behave compliantly in certain situations. Various well-known control strategies let the robot react compliantly to an external contact. Using an impedance controller, the robot reacts to a deviation from the desired position with a restoring force. With this control concept, the robotic system acts as a virtual spring-mass-damper system with specifiable virtual impedance parameters [16]. Another method to implement a desired impedance is by using an admittance controller. An admittance controller reacts to an external force with a change in the desired position. The admittance-controlled system is position controlled and reacts to an external force, whereas an impedance-controlled system is torque controlled and reacts to a deviation in the position. Depending on the system at hand, either an impedance or admittance controller may be used. Note that for the

implementation of an impedance or admittance controller, the external force must be measurable. Otherwise, the virtual mass is not specifiable and only the virtual spring and the virtual damper parameters can be chosen. This resulting controller is called compliance controller.

1.2 Human-Robot Interaction in Collaborative Robotics

The interaction between the human and the robot is a very active field of research, since both, commanding tasks to the robot and estimating the intention of the robot, must be straightforward in a sophisticated interaction interface. Various interaction modes, such as visual, vocal, gestural or haptical among others, are studied [17]. Some interaction modes like vocal and gestural commanding appear natural to humans, but they are only usable to activate high-level functionality. They have a limited amount of differentiable commands and it is almost impossible to program a milling task by voice or gesture. Thus, without a large degree of robot autonomy, vocal and gestural commanding are of limited benefit in industrial processes. A more promising approach is haptic interaction. With haptic interaction, a more detailed task description can be programmed, while this interaction is still intuitive to the human. This communication type is often referred to as physical Human-Robot Interaction (pHRI) [18, 19]. It is commonly used in the programming concepts for collaborative robots.

The most common methods to program a robotic task are *lead-through programming*, *offline programming*, *walk-through programming* and *programming by demonstration* [4]. In *lead-through programming*, the robot is manually moved through the desired motion with the use of a teach pendant. It is a simple, but time-consuming programming method and is not suitable for complex tasks. With *offline programming*, the programming is remotely done in a specific software without physical access to the actual robot. It simulates the behavior of the complete robotic working environment and the program is deployed to the physical robot only after validation in the simulation. It is suitable for complex tasks and less time-consuming than lead-through programming, but it requires the knowledge of a specific robot programming language and the corresponding software tools. To make the programming more accessible to non-experts, the *walk-through programming* was introduced. During the walk-through programming, the operator is allowed to physically move and guide the robot through the desired process. The robot records the performed motion and is able to exactly reproduce it. In a more generalized way, the method of *programming by demonstration* records multiple demonstrations of a human guiding the robot through the desired motion and tries to learn and abstract the task. By taking multiple demonstrations, the robot learns, e. g., which parts of the motion require higher precision and which parts less. The difficulty lies in finding a way to abstract multiple task demonstrations.

In a collaborative scenario, the human should be able to incorporate its cognitive understanding of the environment. To do so, the operator must be able to adapt the behavior of the robot during the online execution of a task. This can either be the adaptation of a single ongoing task or the selection and switch between mutiple ones. In

the following, some research papers dealing with online single- or multi-task adaptation are briefly discussed.

The work [20] focusses on multi-task adaptation in a human-robot cooperative setup. The authors introduce four tasks and a belief system that decides which task should be the active one. The human can move the robot endeffector and by imitating the desired task, the belief system updates its values, recognizes the human intent and the robot changes its active task. The tasks are encoded as Dynamical Movement Primitives (DMPs) [21], which is a statistical approach to encode tasks and movements. It is the foundation for many other papers dealing with programming by demonstration. In [22], Nemec et al. use speed-scaled DMPs to teach tasks and adapt them during the execution. This work incorporates a two-stage learning process, where the path adaptation depends on the speed of the taught motion (low speed means high precision and vice versa) and the spatial variance of the repetitions. By considering speed and spatial difference, the path w.r.t. time, space and desired impedance parameters is adjusted.

In [23], Ansari and Karayiannidis discuss a task-based role adaptation scheme based on local geometric motion primitives. A rigid body is jointly moved by a human and a robot. The robot supports the human in the two primitive tasks of translation and rotation. Again, a belief system is used to choose the active task.

Another approach to adjust the robot behavior based on external human action is given in [24]. In this work, the future desired trajectory is adjusted based on the force applied by the human. This is helpful to avoid an obstacle on the intially planned trajectory that is detected by the human during the execution. A single push by the human operator is enough to deform the future desired trajectory. After a defined time, the robot returns to the initially planned trajectory and the new trajectory merges in the old one.

1.3 Aim of this Work

While the publications mentioned above investigate the online adaptation of robotic tasks, they rarely discuss HRC in a multi-task scenario, i. e. when multiple predefined tasks exist. If they do so, their tasks are mostly specified as DMPs or specializations thereof. These statistical formulations are useful if a task often changes slightly, or the motion for the task has to be adapted from execution to execution. However, if the tasks are not often adapted and the main goal is to switch between multiple predefined tasks, basing the approach on the concept of path-following control offers more advantages. Control concepts based on path-following control provide a systematic approach to HRC, where the robotic motion is restricted to a geometric path in the workspace. Based on this specified geometric path, the control strategy focusses on two objectives: controlling the robot in the direction of the path and controlling the robot transversal to the path. To do so, a direct formulation of the path and its derivatives is needed. DMPs do not provide that. In contrast to DMPs, the representation as paths has no time parametrization attached to it. This eases the switching between multiple tasks, since the explicit formulation in the representation as path allows to find an optimal position along the path independent of time, while the DMPs depend on the time at the moment of the switch.

Thus, the aim of this work is to design two novel modes of human-robot interaction,

based on the control concept of path-following control. They should include the definition of *multiple* collaborative tasks and enable the human operator to select and switch between them. The first mode, called *Path Snap-In*, should let the operator freely move the robot in its working environment and activate a certain task as soon as the robot comes close to a task-specific pose. In the second mode, referred to as *Path Switch*, the robot should always perform one active task, but it should transition into a new task when it is pushed by the human in a certain task-specific way.

1.4 Overview of this Thesis

The remainder of this work is structured as follows. In Chapter 2, the mathematical model of the 7-DOF collaborative robot used in the experimental section of the presented work is introduced. The control concept is based on this mathematical model.

To describe a general manufacturing task, the position and orientation are given as parametrized paths. The base of the HRI modes presented in this work is provided by path-following control. Its control concept and the used notation are introduced in Chapter 3.

The main contribution of this work is the introduction of two novel human-robot interaction modes. They are presented in Chapter 4, where the first interaction mode Path Snap-In is discussed in Section 4.1 and the second mode Path Switch is introduced in Section 4.2. For the Path Snap-In, a method for activating and deactivating the assisting control depending on the position and orientation of the tool center point is given. For the Path Switch, a method to estimate the human intention is found. It accumulates the exerted force by the human operator and decides which task should be the active one. To transition between two tasks, an online trajectory generator is designed.

In Chapter 5, the concepts are validated in an examplary drilling scenario. The chapter contains the discussion of three different experiments. The first experiment, presented in Section 5.1, includes the teach-in of the robotic drilling task. In the second experiment, presented in Section 5.2, the collaborative drilling task is performed with the Path Snap-In mode and in the third experiment, presented in Section 5.3, the collaborative drilling task exploits the Path Switch mode.

The work is concluded with a summary of the work and an outlook on possible extensions in Chapter 6.

2 Mathematical Model

The first step in controlling the robot is the derivation of a mathematical model. First, the kinematics of the robot are discussed and second, the system dynamics are presented. In this work, the KUKA LBR iiwa 14 R820 is used. The following derivation is tailored to this type of robot. For a more in-depth discussion, see, e. g., [25].

2.1 Kinematics

Describing a pose in the three-dimensional space requires six coordinates, i. e. three for the position and three for the orientation. In this work, homogeneous transformations are used to describe the relation between two coordinate frames. Using the notation $\mathbf{d}_{\mathcal{X}}^{\mathcal{Y}} \in \mathbb{R}^3$ for the position vector and $\mathbf{R}_{\mathcal{X}}^{\mathcal{Y}} \in \mathrm{SO}(3)$ for the rotation matrix of the frame $\mathcal{Y}$ with respect to $\mathcal{X}$, expressed in $\mathcal{X}$, a homogeneous transformation $\mathbf{H}_{\mathcal{X}}^{\mathcal{Y}} \in \mathrm{SE}(3)$ is given by

$$\mathbf{H}_{\mathcal{X}}^{\mathcal{Y}} = \begin{bmatrix} \mathbf{R}_{\mathcal{X}}^{\mathcal{Y}} & \mathbf{d}_{\mathcal{X}}^{\mathcal{Y}} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}, \tag{2.1}$$

where $\mathbf{0}_{m\times n}$ denotes a zero matrix or vector with m rows and n columns. In the following, if m and n are omitted, $\mathbf{0}$ denotes a zero matrix of matching dimension. Successive transformations are written as sequential multiplication of homogeneous transformations. A translation of distance d in the direction of the local axis l is denoted by $\mathbf{H}_{\mathrm{Tl,d}}$ and a rotation of α around the local axis l is denoted by $\mathbf{H}_{\mathrm{Rl},\alpha}$. In the following, the coordinate systems of the robotic joints are denoted by $\mathcal{L}_i$, see Figure 2.1. The homogeneous transformation matrices $\mathbf{H}_{\mathcal{L}_{i-1}}^{\mathcal{L}_i}$ between two successive coordinate systems are described by

$$\mathbf{H}_{\mathcal{L}_{i-1}}^{\mathcal{L}_i} = \mathbf{H}_{\mathrm{Ty,d}_{i,\mathrm{y}}} \mathbf{H}_{\mathrm{Tz,d}_{i,\mathrm{z}}} \mathbf{H}_{\mathrm{Rx},\alpha_i} \mathbf{H}_{\mathrm{Rz,q}_i}, \tag{2.2}$$

with the fixed joint parameters $d_{i,\mathrm{y}}, d_{i,\mathrm{z}}$ and α_i and the variable joint position q_i. Concatenating the robotic links yields the forward kinematics of the tool center point (TCP) frame $\mathcal{T}$ w.r.t. the base frame $\mathcal{B}$, reading as

$$\mathbf{H}_{\mathcal{B}}^{\mathcal{T}}(\mathbf{q}) = \mathbf{H}_{\mathcal{B}}^{\mathcal{L}_1}(q_1)\mathbf{H}_{\mathcal{L}_1}^{\mathcal{L}_2}(q_2)\cdots\mathbf{H}_{\mathcal{L}_6}^{\mathcal{L}_7}(q_7)\mathbf{H}_{\mathcal{L}_7}^{\mathcal{E}}\mathbf{H}_{\mathcal{E}}^{\mathcal{T}} = \begin{bmatrix} \mathbf{R}_{\mathcal{B}}^{\mathcal{T}}(\mathbf{q}) & \mathbf{d}_{\mathcal{B}}^{\mathcal{T}}(\mathbf{q}) \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}, \tag{2.3}$$

with the homogeneous transformation from the base frame $\mathcal{B}$ to the frame of the first robotic joint $\mathcal{L}_1$ (corresponding to $\mathbf{H}_{\mathcal{B}}^{\mathcal{L}_1} = \mathbf{H}_{\mathcal{L}_0}^{\mathcal{L}_1}$) and the transformation from the last joint frame $\mathcal{L}_7$ to the endeffector frame $\mathcal{E}$ (corresponding to a translational transformation of $\mathbf{H}_{\mathcal{L}_7}^{\mathcal{E}} = \mathbf{H}_{\mathrm{Tz,d}_{8,\mathrm{z}}}$), see Figure 2.1. In (2.3), the constant homogeneous transformation from the endeffector frame $\mathcal{E}$ to the TCP frame $\mathcal{T}$ is denoted by $\mathbf{H}_{\mathcal{E}}^{\mathcal{T}}$. The kinematic joint

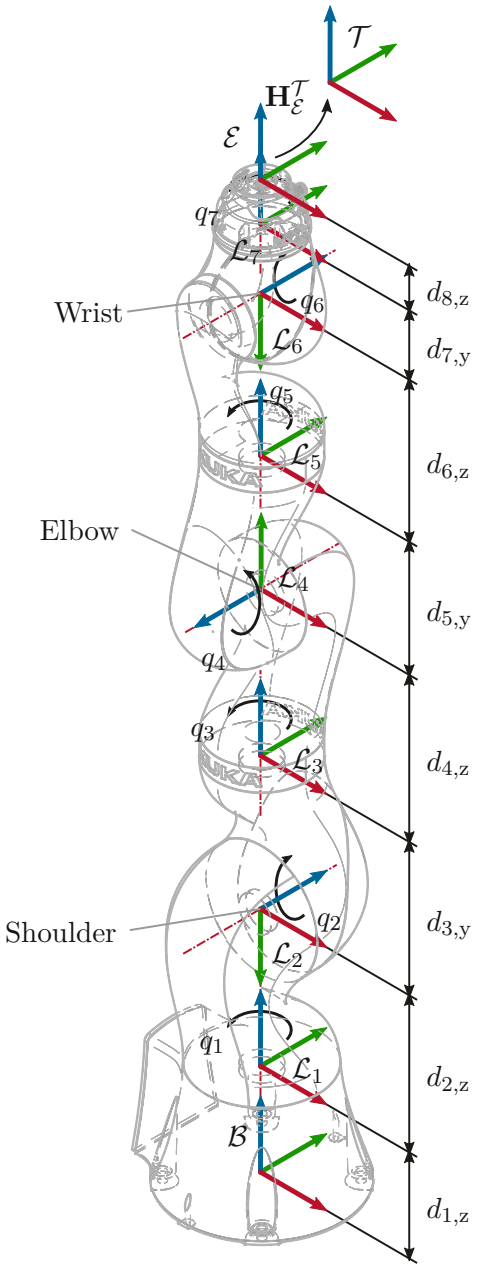


Figure 2.1: Schematic drawing of the robot KUKA LBR iiwa 14 R820 with the used coordinate frames [26].

parameters $d_{i,y}$, $d_{i,z}$, $\alpha_i, i \in \{1, ..., 7\}$ and $d_{8,z}$ for the KUKA LBR iiwa 14 R820 are found in the Appendix A.2.1.

Based on the forward kinematics (2.3), the pose of the TCP is considered as the output

of the system and is defined as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_\mathrm{t} \\ \mathbf{y}_\mathrm{r} \end{bmatrix} = \begin{bmatrix} \mathbf{d}_\mathcal{B}^\mathcal{T}(\mathbf{q}) \\ \boldsymbol{\phi}\left(\mathbf{R}_\mathcal{B}^\mathcal{T}(\mathbf{q})\right) \end{bmatrix} = \begin{bmatrix} \mathbf{h}_\mathrm{t}(\mathbf{q}) \\ \mathbf{h}_\mathrm{r}(\mathbf{q}) \end{bmatrix} = \mathbf{h}(\mathbf{q}) \in \mathbb{R}^6 \ , \tag{2.4}$$

with the position $\mathbf{y}_\mathrm{t} \in \mathbb{R}^3$ and the orientation $\mathbf{y}_\mathrm{r} \in \mathbb{R}^3$ of the TCP frame with respect to the base frame $\mathcal{B}$. The function $\boldsymbol{\phi}\left(\mathbf{R}_\mathcal{B}^\mathcal{T}\right)$ computes a minimal representation of the rotation matrix $\mathbf{R}_\mathcal{B}^\mathcal{T}$.

To relate the joint velocities to the translational and rotational velocities of the TCP, the analytic Jacobian $\mathbf{J}_\mathrm{a}(\mathbf{q}) \in \mathbb{R}^{6\times7}$ is utilized. Differentiating (2.4) results in

$$\dot{\mathbf{y}} = \begin{bmatrix} \dot{\mathbf{y}}_\mathrm{t} \\ \dot{\mathbf{y}}_\mathrm{r} \end{bmatrix} = \frac{\partial \mathbf{h}}{\partial \mathbf{q}}(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_\mathrm{t}(\mathbf{q}) \\ \mathbf{J}_\mathrm{r}(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}_\mathrm{a}(\mathbf{q})\dot{\mathbf{q}} \ . \tag{2.5}$$

The geometric Jacobian $\mathbf{J}_\mathrm{g}(\mathbf{q}) \in \mathbb{R}^{6\times7}$ directly maps to the angular velocity $\boldsymbol{\omega}_\mathcal{B}^\mathcal{T}$ of the TCP frame $\mathcal{T}$ w.r.t. the base frame $\mathcal{B}$ and is computed as

$$\begin{bmatrix} \dot{\mathbf{y}}_\mathrm{t} \\ \boldsymbol{\omega}_\mathcal{B}^\mathcal{T} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_\mathrm{t}(\mathbf{q}) \\ \frac{\partial}{\partial \dot{\mathbf{q}}} \boldsymbol{\omega}_\mathcal{B}^\mathcal{T} \end{bmatrix} \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_\mathrm{t}(\mathbf{q}) \\ \mathbf{J}_\omega(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}_\mathrm{g}(\mathbf{q})\dot{\mathbf{q}} \ , \tag{2.6}$$

The angular velocity $\boldsymbol{\omega}_\mathcal{B}^\mathcal{T}$ is calculated from the rotation matrix $\mathbf{R}_\mathcal{B}^\mathcal{T}(\mathbf{q})$ using

$$\left[\boldsymbol{\omega}_\mathcal{B}^\mathcal{T}\right]_\times = \dot{\mathbf{R}}_\mathcal{B}^\mathcal{T}(\mathbf{q})\left(\mathbf{R}_\mathcal{B}^\mathcal{T}(\mathbf{q})\right)^\mathrm{T} \ , \tag{2.7}$$

with the skew-symmetric operator $[\cdot]_\times : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$ defined as

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}_\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} . \tag{2.8}$$

In the following, if not mentioned otherwise, every rotation is described w.r.t. the base frame $\mathcal{B}$ and the lower index $(.)_\mathcal{B}$ is omitted to improve readability, i. e. $\boldsymbol{\omega}^\mathcal{T} = \boldsymbol{\omega}_\mathcal{B}^\mathcal{T}$.

2.2 Dynamics

The system dynamics describe the reaction of the robot to an applied joint torque $\boldsymbol{\tau}$. It can be shown that the dynamic model of a rigid-body system with 7 degrees of freedom, i. e. $\mathbf{q} \in \mathbb{R}^7$, is formulated as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \underbrace{\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})}_{\mathbf{n}(\mathbf{q},\dot{\mathbf{q}})} = \boldsymbol{\tau} + \boldsymbol{\tau}_\mathrm{e} \ , \tag{2.9}$$

with the mass matrix $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{7\times7}$, the Coriolis matrix $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{7\times7}$, the gravitational vector $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^7$, the applied motor torque $\boldsymbol{\tau} \in \mathbb{R}^7$ and the external torque $\boldsymbol{\tau}_\mathrm{e} \in \mathbb{R}^7$, see, e. g., [25]. The external torque $\boldsymbol{\tau}_\mathrm{e}$ is assumed to be measurable. Although the robot has elastic joints, an underlying singular perturbation controller leads to the quasi-stationary system dynamics with the same structure as (2.9). Hence, the same control strategies as for rigid-body systems can be applied, see, e. g., [27, 28]. Thus, in the following, the rigid-body model (2.9) is used for the controller design.

3 Path Following Control

The goal of robotic applications is for the robot to perform a specific motion. Depending on the task, two motion control problems are distinguished: *trajectory tracking control* (TTC) and *path following control* (PFC).

In TTC, the desired position of the TCP, as well as its derivatives, are known at every time step. The goal of the TTC is to bring the position, velocity and acceleration of the TCP to the desired values.

In contrast, PFC has no a priori time parametrization. Instead, it is based on a geometric curve that the TCP should follow. The desired geometric curve for the TCP is represented as a curve in task space with a path parameter. The primary goal of the controller is to stabilize the TCP on the path. The secondary goal is to provide a desired evolution along the path. PFC is a more general approach than TTC and is the focus of this work.

In this chapter, the control theory of the used PFC is introduced. First, the mathematical definition of a path in task space is given. A parallel transport frame is used to construct path-dependent coordinates. Second, the exact input-output linearization method is applied and third, the controller in the path-dependent frame is designed. An impedance controller is used with different parameters in transversal and tangential directions. In the last step, a nullspace controller is introduced to stabilize the nullspace of the kinematically redundant system. Most concepts of this chapter are based on the works [29, 30], where a more detailed explanation is found. The complete control structure is summarized in Figure 3.1.

3.1 Path Definition

In this work, the explicit parametrization of a geometric path is used. With the mapping

$$\boldsymbol{\sigma}(\theta) : \Theta \to \mathbb{R}^p \ , \tag{3.1}$$

the path parameter θ, the allowed interval Θ and the output dimension p, the path Σ is defined as

$$\Sigma = \{\mathbf{y} \in \mathbb{R}^p \,|\, \mathbf{y} = \boldsymbol{\sigma}(\theta), \theta \in \Theta\} \ . \tag{3.2}$$

A path Σ can be open or closed, depending on Θ. In the following, the path is assumed to be regular, i. e.

$$\boldsymbol{\sigma}'(\bar{\theta}) = \frac{\partial \boldsymbol{\sigma}}{\partial \theta}(\bar{\theta}) \neq \mathbf{0} \quad \forall \bar{\theta} \in \Theta \ , \tag{3.3}$$

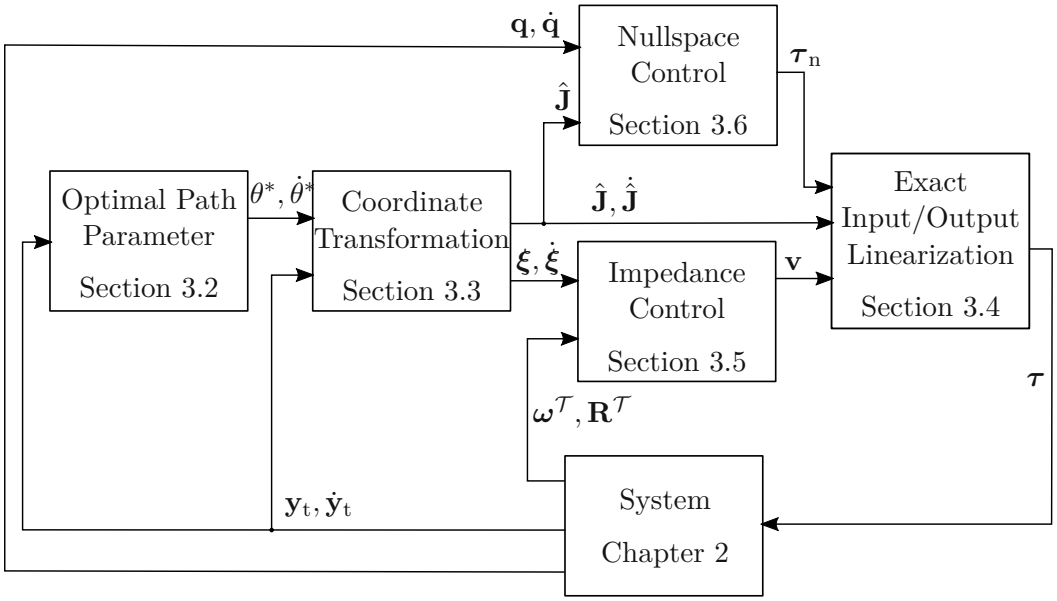


Figure 3.1: Structure of the PFC control scheme.

and the individual components of $\boldsymbol{\sigma}(\bar{\theta}), \bar{\theta} \in \Theta$, are assumed to be differentiable up to the necessary degree.

For the representation of a pose in the three dimensional Euclidean space, three parameters for the translational part and at least three parameters for the rotational part are needed. Therefore, the curve $\boldsymbol{\sigma}(\theta)$ incorporates a translational part $\boldsymbol{\sigma}_\mathrm{t}(\theta)$ and a rotational part $\boldsymbol{\sigma}_\mathrm{r}(\theta)$ in the form

$$\boldsymbol{\sigma}(\theta) = \begin{bmatrix} \boldsymbol{\sigma}_\mathrm{t}(\theta) \\ \boldsymbol{\sigma}_\mathrm{r}(\theta) \end{bmatrix} . \tag{3.4}$$

The translational part $\boldsymbol{\sigma}_\mathrm{t}(\theta)$ is used to find an optimal position along the path, which will be specified in more detail later in this section, with the corresponding optimal path parameter θ^*. Additionally, it is assumed that on any given point along the path, a desired orientation $\boldsymbol{\sigma}_\mathrm{r}(\theta)$ is given. This desired orientation of the tool frame $\mathcal{T}$ w.r.t. the base frame $\mathcal{B}$ can also be represented as a rotation matrix $\mathbf{R}_\mathcal{B}^{\mathcal{T},\mathrm{d}}(\boldsymbol{\sigma}_\mathrm{r}(\theta))$ or as a unit quaternion $\mathbf{Q}_\mathcal{B}^{\mathcal{T},\mathrm{d}}(\boldsymbol{\sigma}_\mathrm{r}(\theta))$. To simplify the notation, the local desired frame on the path is denoted by $\mathcal{D}$ and the relations $\mathbf{R}^\mathcal{D}(\theta) = \mathbf{R}_\mathcal{B}^{\mathcal{T},\mathrm{d}}(\boldsymbol{\sigma}_\mathrm{r}(\theta))$ and $\mathbf{Q}^\mathcal{D}(\theta) = \mathbf{Q}_\mathcal{B}^{\mathcal{T},\mathrm{d}}(\boldsymbol{\sigma}_\mathrm{r}(\theta))$ are introduced. This relations are utilized in the discussions of Chapter 4.

3.2 Path-Dependent Frame

The control scheme should behave differently in tangential and transversal direction of the path. In order to define coordinates in these directions, a new frame is constructed on every point along the path. A commonly used frame is the Frenet-Serret frame. However, since it is undefined at points with zero curvature, which is needed for many applications,

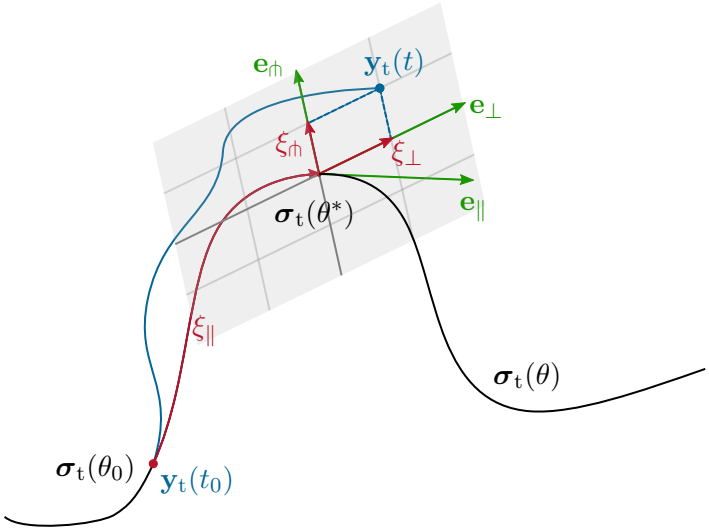


Figure 3.2: Path-based frame and coordinates.

the parallel transport frame is employed instead (see Figure 3.2). The frame consists of the path-dependent tangent vector $\mathbf{e}_\parallel(\theta)$, the normal vector $\mathbf{e}_\perp(\theta)$ and the binormal vector $\mathbf{e}_{\pitchfork}(\theta)$. The unit tangent vector $\mathbf{e}_\parallel(\theta)$ is given by

$$\mathbf{e}_\parallel(\theta) = \frac{\boldsymbol{\sigma}_\mathrm{t}'(\theta)}{\|\boldsymbol{\sigma}_\mathrm{t}'(\theta)\|_2} \ . \tag{3.5}$$

This vector points in the direction of the path tangent and always exists due to the assumed regularity of the curve (3.3).

The normal and the binormal vector, i. e. $\mathbf{e}_\perp(\theta)$ and $\mathbf{e}_{\pitchfork}(\theta)$, respectively, are orthogonal to the tangent vector $\mathbf{e}_\parallel(\theta)$ and span the transversal plane on every point along the path. The normal vector $\mathbf{e}_\perp(\theta)$ is calculated by solving the differential algebraic system of equations

$$\mathbf{e}_\perp'(\theta) = -(\mathbf{e}_\parallel'(\theta))^\mathrm{T}\mathbf{e}_\perp(\theta)\mathbf{e}_\parallel(\theta), \quad \mathbf{e}_\perp(\theta) = \mathbf{e}_{\perp,0} \tag{3.6a}$$

$$0 = 1 - \mathbf{e}_\perp^\mathrm{T}(\theta)\mathbf{e}_\perp(\theta) \tag{3.6b}$$

$$0 = \mathbf{e}_\parallel^\mathrm{T}(\theta)\mathbf{e}_\perp(\theta) \ , \tag{3.6c}$$

see [29, 31]. To complete the parallel transport frame, the binormal vector $\mathbf{e}_{\pitchfork}(\theta)$ is obtained using

$$\mathbf{e}_{\pitchfork}(\theta) = \mathbf{e}_\parallel(\theta) \times \mathbf{e}_\perp(\theta) \ . \tag{3.7}$$

With the equations (3.5) – (3.7), the parallel transport frame is completely defined for any path parameter $\theta \in \Theta$ along the path $\boldsymbol{\sigma}(\theta)$. This parallel transport frame is denoted by $\mathcal{P}$. The optimal path parameter θ^* is chosen such that the distance between the TCP position $\mathbf{y}_\mathrm{t}$ and the corresponding point on the path $\boldsymbol{\sigma}_\mathrm{t}(\theta^*)$ is minimized. By solving the minimization problem

$$\theta^* = \underset{\theta\in\Theta}{\arg\min} \, \|\mathbf{y}_\mathrm{t} - \boldsymbol{\sigma}_\mathrm{t}(\theta)\|_2^2 \ , \tag{3.8}$$

the optimal path parameter θ^* is found. In order to be a solution of the minimization problem, the necessary first-order condition

$$\frac{\partial}{\partial\theta}\Big(\|\mathbf{y}_\mathrm{t}-\boldsymbol{\sigma}_\mathrm{t}(\theta)\|_2^2\Big)\Big|_{\theta=\theta^*} = -2(\mathbf{y}_\mathrm{t}-\boldsymbol{\sigma}_\mathrm{t}(\theta^*))^\mathrm{T}\boldsymbol{\sigma}_\mathrm{t}'(\theta^*) = 0 \tag{3.9}$$

must hold. Equation (3.9) reveals that the deviation vector $\mathbf{y}_\mathrm{t}-\boldsymbol{\sigma}_\mathrm{t}(\theta^*)$ is orthogonal to the tangent vector $\mathbf{e}_\parallel(\theta^*)$ and, hence, lies in the transversal plane, cf. (3.5). Deriving the sufficient second-order condition yields

$$\frac{\partial^2}{\partial\theta^2}\Big(\|\mathbf{y}_\mathrm{t}-\boldsymbol{\sigma}_\mathrm{t}(\theta)\|_2^2\Big)\Big|_{\theta=\theta^*} > 0 \tag{3.10}$$

$$2\|\boldsymbol{\sigma}_\mathrm{t}'(\theta^*)\|_2^2 - 2(\mathbf{y}_\mathrm{t}-\boldsymbol{\sigma}_t(\theta^*))^\mathrm{T}\boldsymbol{\sigma}_\mathrm{t}''(\theta^*) > 0 \tag{3.11}$$

$$\frac{(\mathbf{y}_\mathrm{t}-\boldsymbol{\sigma}_\mathrm{t}(\theta^*))^\mathrm{T}\boldsymbol{\sigma}_\mathrm{t}''(\theta^*)}{\|\boldsymbol{\sigma}_\mathrm{t}'(\theta^*)\|_2^2} < 1\ . \tag{3.12}$$

With the auxiliary variable

$$\alpha(\mathbf{y}_\mathrm{t}) = \frac{(\mathbf{y}_\mathrm{t}-\boldsymbol{\sigma}_\mathrm{t}(\theta^*))^\mathrm{T}\boldsymbol{\sigma}_\mathrm{t}''(\theta^*)}{\|\boldsymbol{\sigma}_\mathrm{t}'(\theta^*)\|_2^2}\ , \tag{3.13}$$

at the optimal solution θ^* of (3.8), the relation $\alpha(\mathbf{y}_\mathrm{t}) < 1$ must hold.

In Section 3.3 and further, an expression for the time derivative of the optimal path parameter $\dot\theta^*$ is needed. It is calculated from the necessary first-order optimality condition (3.9). Derivating (3.9) w.r.t. time yields

$$(\dot{\mathbf{y}}_\mathrm{t}-\boldsymbol{\sigma}_\mathrm{t}'(\theta^*)\dot\theta^*)^\mathrm{T}\boldsymbol{\sigma}_\mathrm{t}'(\theta^*) + (\mathbf{y}_\mathrm{t}-\boldsymbol{\sigma}_\mathrm{t}(\theta^*))^\mathrm{T}\boldsymbol{\sigma}_\mathrm{t}''(\theta^*)\dot\theta^* = 0\ . \tag{3.14}$$

Using (3.13), equation (3.14) is rewritten as

$$\begin{aligned}\dot{\mathbf{y}}_\mathrm{t}^\mathrm{T}\boldsymbol{\sigma}_\mathrm{t}'(\theta^*) &= \Big(\|\boldsymbol{\sigma}_\mathrm{t}'(\theta^*)\|_2^2 - (\mathbf{y}_\mathrm{t}-\boldsymbol{\sigma}_\mathrm{t}(\theta^*))^\mathrm{T}\boldsymbol{\sigma}_\mathrm{t}''(\theta^*)\Big)\dot\theta^* \\ &= (1-\alpha(\mathbf{y}_\mathrm{t}))\|\boldsymbol{\sigma}_\mathrm{t}(\theta^*)\|_2^2\dot\theta^*\ .\end{aligned} \tag{3.15}$$

Finally, introducing the auxiliary variable

$$\beta(\mathbf{y}_\mathrm{t}) = (1-\alpha(\mathbf{y}_\mathrm{t}))^{-1} \tag{3.16}$$

and rearranging (3.15), the time derivative of the optimal path parameter θ^* is obtained as

$$\dot\theta^* = \frac{1}{1-\alpha(\mathbf{y}_\mathrm{t})}\frac{\boldsymbol{\sigma}_\mathrm{t}'(\theta^*)^\mathrm{T}}{\|\boldsymbol{\sigma}_\mathrm{t}'(\theta^*)\|_2^2}\dot{\mathbf{y}}_\mathrm{t} = \frac{\beta(\mathbf{y}_\mathrm{t})\mathbf{e}_\parallel^\mathrm{T}(\theta^*)}{\|\boldsymbol{\sigma}_\mathrm{t}'(\theta^*)\|_2}\dot{\mathbf{y}}_\mathrm{t}\ . \tag{3.17}$$

3.3 Coordinate Transformation

Based on the path-dependent parallel transport frame $(\mathbf{e}_\parallel, \mathbf{e}_\perp, \mathbf{e}_\pitchfork)$, new coordinates $\boldsymbol{\xi}^\mathrm{T} = \begin{bmatrix}\xi_\parallel & \xi_\perp & \xi_\pitchfork\end{bmatrix}$ are introduced in this section, see Figure 3.2. The canonical form to

parametrize a curve is based on its arc-length. Hence, the tangential coordinate $\xi_\parallel$ is chosen as the arc-length of the path at the optimal path parameter θ^* as

$$\xi_\parallel = \int_{\theta_0}^{\theta^*} \|\boldsymbol{\sigma}_\mathrm{t}'(\tau)\|_2 \mathrm{d}\tau \ , \tag{3.18}$$

with the initial path parameter $\theta_0 \in \mathbb{R}$ at the beginning of operation. The transversal coordinates $\xi_\perp$ and $\xi_\pitchfork$ are chosen as the projections of the deviation vector $(\mathbf{y}_\mathrm{t} - \boldsymbol{\sigma}_\mathrm{t}(\theta^*))$ onto the normal and binormal unit vectors, $\mathbf{e}_\perp(\theta)$ and $\mathbf{e}_\pitchfork(\theta)$, respectively, given by

$$\xi_\perp = \mathbf{e}_\perp^\mathrm{T}(\theta^*)(\mathbf{y}_\mathrm{t} - \boldsymbol{\sigma}_\mathrm{t}(\theta^*)) \tag{3.19}$$

$$\xi_\pitchfork = \mathbf{e}_\pitchfork^\mathrm{T}(\theta^*)(\mathbf{y}_\mathrm{t} - \boldsymbol{\sigma}_\mathrm{t}(\theta^*)) \ . \tag{3.20}$$

Note that the new coordinates $\boldsymbol{\xi}$ only depend on the system output $\mathbf{y}_\mathrm{t}$, which in turn only depends on the generalized coordinates $\mathbf{q}$, see (2.4).

The time derivatives of the tangential and the tranversal coordinates are calculated in the following. For the tangential coordinate $\xi_\parallel$, the derivative results in

$$\dot{\xi}_\parallel = \|\boldsymbol{\sigma}_\mathrm{t}'(\theta^*)\|_2 \dot{\theta}^* \ . \tag{3.21}$$

Using the calculated time derivative of the optimal path parameter $\dot{\theta}^*$, by putting (3.17) back into (3.21), gives the time derivative of the tangential coordinate $\dot{\xi}_\parallel$

$$\dot{\xi}_\parallel = \beta(\mathbf{y}_\mathrm{t}) \mathbf{e}_\parallel^\mathrm{T}(\theta^*) \dot{\mathbf{y}}_\mathrm{t} \ . \tag{3.22}$$

The time derivative of the normal coordinate $\xi_\perp$ yields

$$\dot{\xi}_\perp = \dot{\mathbf{e}}_\perp^\mathrm{T}(\theta^*)(\mathbf{y}_\mathrm{t} - \boldsymbol{\sigma}_\mathrm{t}(\theta^*)) + \mathbf{e}_\perp^\mathrm{T}(\theta^*)(\dot{\mathbf{y}}_\mathrm{t} - \boldsymbol{\sigma}_\mathrm{t}'(\theta^*)\dot{\theta}^*) \ , \tag{3.23}$$

which simplifies to

$$\dot{\xi}_\perp = \mathbf{e}_\perp^\mathrm{T}(\theta^*) \dot{\mathbf{y}}_\mathrm{t} \tag{3.24}$$

by using $\mathbf{e}_\perp^\mathrm{T}(\theta^*)\boldsymbol{\sigma}_\mathrm{t}'(\theta^*) = 0$ and the property $\dot{\mathbf{e}}_\perp(\theta^*) \parallel \boldsymbol{\sigma}_\mathrm{t}'(\theta^*)$, see (3.6a) and (3.9). Repeating the above for the binormal coordinate yields

$$\dot{\xi}_\pitchfork = \mathbf{e}_\pitchfork^\mathrm{T}(\theta^*) \dot{\mathbf{y}}_\mathrm{t} \ . \tag{3.25}$$

Putting (3.18), (3.19) and (3.20) together, the path-based coordinates $\boldsymbol{\xi}$ are defined as

$$\boldsymbol{\xi} = \begin{bmatrix} \xi_\parallel \\ \xi_\perp \\ \xi_\pitchfork \end{bmatrix} = \begin{bmatrix} \int_{\theta_0}^{\theta^*} \|\boldsymbol{\sigma}_\mathrm{t}'(\tau)\|_2 \mathrm{d}\tau \\ \mathbf{e}_\perp^\mathrm{T}(\theta^*)(\mathbf{y}_\mathrm{t} - \boldsymbol{\sigma}_\mathrm{t}(\theta^*)) \\ \mathbf{e}_\pitchfork^\mathrm{T}(\theta^*)(\mathbf{y}_\mathrm{t} - \boldsymbol{\sigma}_\mathrm{t}(\theta^*)) \end{bmatrix} \ . \tag{3.26}$$

Its derivative w.r.t. time is compactly rewritten as

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix} \dot{\xi}_\parallel \\ \dot{\xi}_\perp \\ \dot{\xi}_\pitchfork \end{bmatrix} = \begin{bmatrix} \beta(\mathbf{y}_\mathrm{t}) \mathbf{e}_\parallel^\mathrm{T}(\theta^*) \\ \mathbf{e}_\perp^\mathrm{T}(\theta^*) \\ \mathbf{e}_\pitchfork^\mathrm{T}(\theta^*) \end{bmatrix} \dot{\mathbf{y}}_\mathrm{t} = \underbrace{\begin{bmatrix} \beta(\mathbf{y}_\mathrm{t}) \mathbf{e}_\parallel^\mathrm{T}(\theta^*) \\ \mathbf{e}_\perp^\mathrm{T}(\theta^*) \\ \mathbf{e}_\pitchfork^\mathrm{T}(\theta^*) \end{bmatrix} \mathbf{J}_\mathrm{t}(\mathbf{q})}_{\mathbf{J}_\xi(\mathbf{q})} \dot{\mathbf{q}} = \mathbf{J}_\xi(\mathbf{q}) \dot{\mathbf{q}} \tag{3.27}$$

with the transformed translational Jacobian $\mathbf{J}_\xi(\mathbf{q})$.

3.4 Exact Input-Output Feedback Linearization

In order to provide the system with a virtual input that is directly linked to the transformed coordinates $\boldsymbol{\xi}$, an exact input-output feedback linearization is applied. This method transforms the system (2.9) into a system with linear input-output dynamics from a virtual input $\mathbf{v} \in \mathbb{R}^6$ to the acceleration of the transformed coordinates $\ddot{\boldsymbol{\xi}}$ and the angular acceleration $\dot{\boldsymbol{\omega}}^{\mathcal{T}}$. Combining the transformed dynamics (3.27) and the definition of $\mathbf{J}_\omega(\mathbf{q})$ from (2.6), the transformed Jacobian $\hat{\mathbf{J}}(\mathbf{q})$ is introduced as

$$\begin{bmatrix} \dot{\boldsymbol{\xi}} \\ \boldsymbol{\omega}^{\mathcal{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_\xi(\mathbf{q}) \\ \mathbf{J}_\omega(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \hat{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} \ . \tag{3.28}$$

The derivative of (3.28) results in

$$\begin{bmatrix} \ddot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{\omega}}^{\mathcal{T}} \end{bmatrix} = \hat{\mathbf{J}}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\hat{\mathbf{J}}}(\mathbf{q})\dot{\mathbf{q}} \tag{3.29}$$

and is rearranged to obtain the joint acceleration $\ddot{\mathbf{q}}$ as

$$\ddot{\mathbf{q}} = \hat{\mathbf{J}}^{\dagger}(\mathbf{q}) \left(\begin{bmatrix} \ddot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{\omega}}^{\mathcal{T}} \end{bmatrix} - \dot{\hat{\mathbf{J}}}(\mathbf{q})\dot{\mathbf{q}} \right) , \tag{3.30}$$

with the right pseudoinverse $\hat{\mathbf{J}}^{\dagger}(\mathbf{q}) = \hat{\mathbf{J}}^{\mathrm{T}}(\mathbf{q}) \left(\hat{\mathbf{J}}(\mathbf{q}) \hat{\mathbf{J}}^{\mathrm{T}}(\mathbf{q}) \right)^{-1}$. Putting (3.30) back into the system dynamics equation (2.9) yields the system dynamics w.r.t. the transformed coordinates

$$\mathbf{M}(\mathbf{q}) \underbrace{\hat{\mathbf{J}}^{\dagger}(\mathbf{q}) \left(\begin{bmatrix} \ddot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{\omega}}^{\mathcal{T}} \end{bmatrix} - \dot{\hat{\mathbf{J}}}(\mathbf{q})\dot{\mathbf{q}} \right)}_{\ddot{\mathbf{q}} \text{ from (3.30)}} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} + \boldsymbol{\tau}_e \ . \tag{3.31}$$

By choosing the system input $\boldsymbol{\tau}$ as

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q}) \hat{\mathbf{J}}^{\dagger}(\mathbf{q}) \left(\mathbf{v} - \dot{\hat{\mathbf{J}}}(\mathbf{q})\dot{\mathbf{q}} \right) + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) - \boldsymbol{\tau}_{\mathrm{e}} + \boldsymbol{\tau}_{\mathrm{n}} \ , \tag{3.32}$$

the system dynamics (3.31) simplifies to

$$\begin{bmatrix} \ddot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{\omega}}^{\mathcal{T}} \end{bmatrix} = \mathbf{v} + \hat{\mathbf{J}}(\mathbf{q}) \mathbf{M}^{-1}(\mathbf{q}) \boldsymbol{\tau}_{\mathrm{n}} \ , \tag{3.33}$$

with the additional nullspace control input $\boldsymbol{\tau}_{\mathrm{n}} \in \mathbb{R}^7$. The nullspace controller will be introduced in Section 3.6 and is chosen such that it has no influence on the coordinates $\ddot{\boldsymbol{\xi}}$ and $\dot{\boldsymbol{\omega}}^{\mathcal{T}}$, i. e. its control action lies in the nullspace of $\hat{\mathbf{J}}(\mathbf{q})$. Hence, (3.33) simplifies to

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_\xi \\ \mathbf{v}_\omega \end{bmatrix} = \begin{bmatrix} \ddot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{\omega}}^{\mathcal{T}} \end{bmatrix} , \tag{3.34}$$

resulting in a linear input-output characteristics from the virtual inputs $\mathbf{v}_\xi$ for translation and $\mathbf{v}_\omega$ for rotation to the accelerations in the transformed coordinates $\ddot{\boldsymbol{\xi}}$ and $\dot{\boldsymbol{\omega}}^{\mathcal{T}}$, respectively.

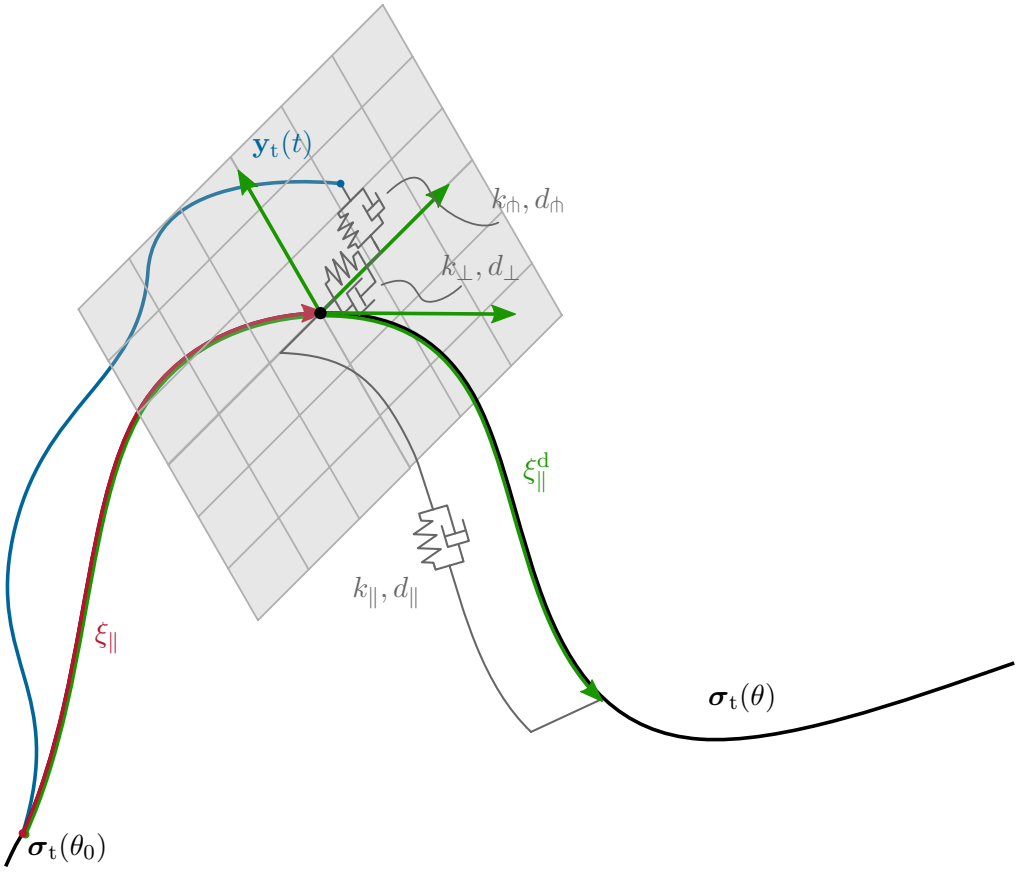


Figure 3.3: Path-based impedance.

3.5 Path-Based Impedance Control

The next step of the derivation of PFC is to specify the virtual input $\mathbf{v}$. This work focusses on the interaction between humans and robots. When a human is involved in the task, the robot should behave in a compliant way. A common way of introducing defined compliance in the robot behavior is by using an impedance controller [25]. This allows the compliance of the robot to be set by parameters and be adapted according to the task at hand. For HRI, the desired behavior of the TCP is a spring-mass-damper system with tunable impedance parameters in the coordinates $\mathbf{e}_\parallel, \mathbf{e}_\perp$ and $\mathbf{e}_\pitchfork$ of the path-based frame introduced in Section 3.1, see Figure 3.3.

Thus, the desired impedance dynamics is given by

$$\underbrace{\mathbf{M}^\mathrm{d}\begin{bmatrix}\ddot{\boldsymbol{\xi}}-\ddot{\boldsymbol{\xi}}^\mathrm{d}\\ \dot{\boldsymbol{\omega}}^\mathcal{T}-\dot{\boldsymbol{\omega}}^\mathcal{D}\end{bmatrix}}_{\text{mass}}+\underbrace{\mathbf{D}^\mathrm{d}\begin{bmatrix}\dot{\boldsymbol{\xi}}-\dot{\boldsymbol{\xi}}^\mathrm{d}\\ \boldsymbol{\omega}^\mathcal{T}-\boldsymbol{\omega}^\mathcal{D}\end{bmatrix}}_{\text{damper}}+\underbrace{\mathbf{K}^\mathrm{d}\begin{bmatrix}\boldsymbol{\xi}-\boldsymbol{\xi}^\mathrm{d}\\ \varepsilon_\mathcal{T}^\mathcal{D}\end{bmatrix}}_{\text{spring}}=\hat{\mathbf{f}}_\mathrm{e}\,, \tag{3.35}$$

with the desired mass matrix $\mathbf{M}^\mathrm{d}\in\mathbb{R}^{6\times6}$, the desired damper matrix $\mathbf{D}^\mathrm{d}\in\mathbb{R}^{6\times6}$, the desired spring matrix $\mathbf{K}^\mathrm{d}\in\mathbb{R}^{6\times6}$, the desired path-based position, velocity and acceleration

$\boldsymbol{\xi}^{\mathrm{d}}$, $\dot{\boldsymbol{\xi}}^{\mathrm{d}}$ and $\ddot{\boldsymbol{\xi}}^{\mathrm{d}}$, respectively. Note that the rotational part of the impedance controller (3.35) is implemented using unit quaternions, see, e. g., [32], by utilizing the vector part of the quaternion error $\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}}$ between the orientation of the TCP frame $\mathcal{T}$ and the orientation of the desired frame $\mathcal{D}$ and the angular velocity and acceleration of $\mathcal{D}$, i. e., $\boldsymbol{\omega}^{\mathcal{D}}$ and $\dot{\boldsymbol{\omega}}^{\mathcal{D}}$, respectively. In (3.35), $\boldsymbol{\omega}^{\mathcal{D}}$ and $\dot{\boldsymbol{\omega}}^{\mathcal{D}}$ correspond to the *desired* angular velocity and acceleration of the TCP frame, see Section 3.1. The vector part of the quaternion error $\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}}$ in (3.35) is calculated using the quaternion product $\otimes$ as

$$\{e_{\mathcal{T}}^{\mathcal{D}}, \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}}\} = \mathbf{Q}^{\mathcal{D}}(\theta^*) \otimes \left(\mathbf{Q}^{\mathcal{T}}\right)^{-1} , \tag{3.36}$$

with the desired orientation $\mathbf{Q}^{\mathcal{D}}(\theta^*)$ and the TCP orientation $\mathbf{Q}^{\mathcal{T}}$ expressed as unit quaternions. The vector $\hat{\mathbf{f}}_{\mathrm{e}}$ describes the external forces and torques transformed to the parallel transport frame $(\mathbf{e}_{\parallel}, \mathbf{e}_{\perp}, \mathbf{e}_{\pitchfork})$ consisting of forces along the path-based coordinates and torques in the base frame $\mathcal{B}$. It is computed using

$$\hat{\mathbf{f}}_{\mathrm{e}} = \left(\hat{\mathbf{J}}^{\dagger}(\mathbf{q})\right)^{\mathrm{T}} \boldsymbol{\tau}_{\mathrm{e}} , \tag{3.37}$$

where the generalized external forces $\boldsymbol{\tau}_{\mathrm{e}}$ are assumed to be measured. The desired mass-spring-damper dynamics is achieved by setting the virtual input $\mathbf{v}$ to

$$\mathbf{v} = \begin{bmatrix} \ddot{\boldsymbol{\xi}}^{\mathrm{d}} \\ \dot{\boldsymbol{\omega}}^{\mathcal{D}} \end{bmatrix} + \left(\mathbf{M}^{\mathrm{d}}\right)^{-1} \left(\hat{\mathbf{f}}_{\mathrm{e}} - \mathbf{D}^{\mathrm{d}} \begin{bmatrix} \dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}^{\mathrm{d}} \\ \boldsymbol{\omega}^{\mathcal{T}} - \boldsymbol{\omega}^{\mathcal{D}} \end{bmatrix} - \mathbf{K}^{\mathrm{d}} \begin{bmatrix} \boldsymbol{\xi} - \boldsymbol{\xi}^{\mathrm{d}} \\ \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}} \end{bmatrix} \right) . \tag{3.38}$$

Putting (3.38) back into the feedback linearization controller (3.32) yields the final control law

$$\boldsymbol{\tau} = \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) - \boldsymbol{\tau}_{\mathrm{e}} + \boldsymbol{\tau}_{\mathrm{n}} + \mathbf{M}(\mathbf{q}) \hat{\mathbf{J}}^{\dagger}(\mathbf{q}) \left(\underbrace{\begin{bmatrix} \ddot{\boldsymbol{\xi}}^{\mathrm{d}} \\ \dot{\boldsymbol{\omega}}^{\mathcal{D}} \end{bmatrix} + \left(\mathbf{M}^{\mathrm{d}}\right)^{-1} \left(\hat{\mathbf{f}}_{\mathrm{e}} - \mathbf{D}^{\mathrm{d}} \begin{bmatrix} \dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}^{\mathrm{d}} \\ \boldsymbol{\omega}^{\mathcal{T}} - \boldsymbol{\omega}^{\mathcal{D}} \end{bmatrix} - \mathbf{K}^{\mathrm{d}} \begin{bmatrix} \boldsymbol{\xi} - \boldsymbol{\xi}^{\mathrm{d}} \\ \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}} \end{bmatrix} \right)}_{\mathbf{v} \text{ from (3.38)}} - \dot{\hat{\mathbf{J}}}(\mathbf{q}) \dot{\mathbf{q}} \right) . \tag{3.39}$$

3.6 Nullspace Control

The state space of the introduced path-based coordinates

$$\left[\left(\boldsymbol{\xi} - \boldsymbol{\xi}^{\mathrm{d}}\right)^{\mathrm{T}} \quad \left(\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}^{\mathrm{d}}\right)^{\mathrm{T}} \quad \left(\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}}\right)^{\mathrm{T}} \quad \left(\boldsymbol{\omega}^{\mathcal{T}} - \boldsymbol{\omega}^{\mathcal{D}}\right)^{\mathrm{T}} \right]^{\mathrm{T}} \in \mathbb{R}^{12} , \tag{3.40}$$

which are used as new system output, do not cover the complete system state $\begin{bmatrix} \mathbf{q}^{\mathrm{T}} & \dot{\mathbf{q}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{14}$. Hence, the complete system comprises a nullspace and a nullspace controller is needed to stabilize the complete system. The nullspace controller allows to consider a secondary control objective, which can be used to keep the robot configuration away from singularities

or joint limits. To this end, a projection matrix is used to project the nullspace control input $\mathbf{v}_\mathrm{n} \in \mathbb{R}^7$ onto the nullspace torque $\boldsymbol{\tau}_\mathrm{n}$, reading as

$$\boldsymbol{\tau}_\mathrm{n} = \mathbf{M}(\mathbf{q})\Big(\mathbf{I} - \hat{\mathbf{J}}^\dagger(\mathbf{q})\hat{\mathbf{J}}(\mathbf{q})\Big)\mathbf{v}_\mathrm{n} \ , \tag{3.41}$$

with the identity matrix of matching dimension $\mathbf{I}$. This ensures that $\boldsymbol{\tau}_\mathrm{n}$ is only acting on the nullspace of $\hat{\mathbf{J}}(\mathbf{q})$, since (3.33) simplifies to

$$\begin{aligned}\begin{bmatrix}\ddot{\boldsymbol{\xi}}\\ \dot{\boldsymbol{\omega}}\end{bmatrix} &= \mathbf{v} + \hat{\mathbf{J}}(\mathbf{q})\mathbf{M}^{-1}(\mathbf{q})\underbrace{\mathbf{M}(\mathbf{q})\Big(\mathbf{I} - \hat{\mathbf{J}}^\dagger(\mathbf{q})\hat{\mathbf{J}}(\mathbf{q})\Big)\mathbf{v}_\mathrm{n}}_{\boldsymbol{\tau}_\mathrm{n} \text{ from (3.41)}}\\ &= \mathbf{v} + \Big(\hat{\mathbf{J}}(\mathbf{q}) - \hat{\mathbf{J}}(\mathbf{q})\Big)\mathbf{v}_\mathrm{n} = \mathbf{v} \ . \end{aligned} \tag{3.42}$$

A common choice for $\mathbf{v}_\mathrm{n}$ is the PD control law

$$\mathbf{v}_\mathrm{n} = -\mathbf{D}_\mathrm{n}\dot{\mathbf{q}} - \mathbf{K}_\mathrm{n}(\mathbf{q} - \mathbf{q}_0) \tag{3.43}$$

with the virtual resting position $\mathbf{q}_0 \in \mathbb{R}^7$ and the positive definite nullspace control matrices $\mathbf{K}_\mathrm{n}, \mathbf{D}_\mathrm{n} \in \mathbb{R}^{7\times 7}$. The nullspace damping matrix $\mathbf{D}_\mathrm{n}$ parametrizes the virtual dampers in the nullspace that counteract the joint velocities. The nullspace stiffness matrix $\mathbf{K}_\mathrm{n}$ specifies the virtual spring stiffness to push the robot joint configuration $\mathbf{q}$ as close to $\mathbf{q}_0$ as possible without influencing the actual TCP pose.

In human-robot interaction, the virtual spring in the nullspace may be counterintuitive. The nullspace controller (3.43) causes the robot to stabilize to a desired position in its nullspace due to the spring force term. However, the operator may also want to interact with the robot in its nullspace, e. g. because a certain elbow position is more convenient during the interaction. A more intuitive behavior is implemented by replacing the linear spring force in (3.43) by a nonlinear spring force $\boldsymbol{\tau}_\mathrm{kn}^\mathrm{T}(\mathbf{q}) = \begin{bmatrix}\tau_{\mathrm{kn},1}(q_1) & \tau_{\mathrm{kn},2}(q_2) & \dots & \tau_{\mathrm{kn},7}(q_7)\end{bmatrix}$ that is only acting close to the joint limits defined by

$$\tau_{\mathrm{kn},i}(q_i) = \begin{cases} k_\mathrm{n}\left(\frac{1}{(q_i - q_{\mathrm{min},i})^2} - \frac{1}{(\Delta q_i)^2}\right), & q_{\mathrm{min},i} \le q_i < q_{\mathrm{min},i} + \Delta q_i \\ 0, & q_{\mathrm{min},i} + \Delta q_i \le q_i < q_{\mathrm{max},i} - \Delta q_i \\ k_\mathrm{n}\left(\frac{1}{(\Delta q_i)^2} - \frac{1}{(q_{\mathrm{max},i} - q_i)^2}\right) & q_{\mathrm{max},i} - \Delta q_i \le q_i < q_{\mathrm{max},i} \end{cases} . \tag{3.44}$$

This spring force is designed as a barrier function. In vicinity to the joint limits, the force increases significantly and pushes the joints away from the respective limitations. On the contrary, beyond a certain distance Δq_i in the joint space from the joint limits, it has no effect. Figure 3.4 depicts the spring force $\tau_{\mathrm{kn},i}(q_i)$ over the joint angle q_i. Inserting the nonlinear spring force (3.44) back into the nullspace controller yields the nullspace controller

$$\boldsymbol{\tau}_\mathrm{n} = \mathbf{M}(\mathbf{q})\Big(\mathbf{I} - \hat{\mathbf{J}}^\dagger(\mathbf{q})\hat{\mathbf{J}}(\mathbf{q})\Big)(-\mathbf{D}_\mathrm{n}\dot{\mathbf{q}} + \boldsymbol{\tau}_\mathrm{kn}(\mathbf{q})) \ . \tag{3.45}$$

Depending on the use case and preference, either the nullspace control (3.43) or (3.45) or a combination of both may be used.

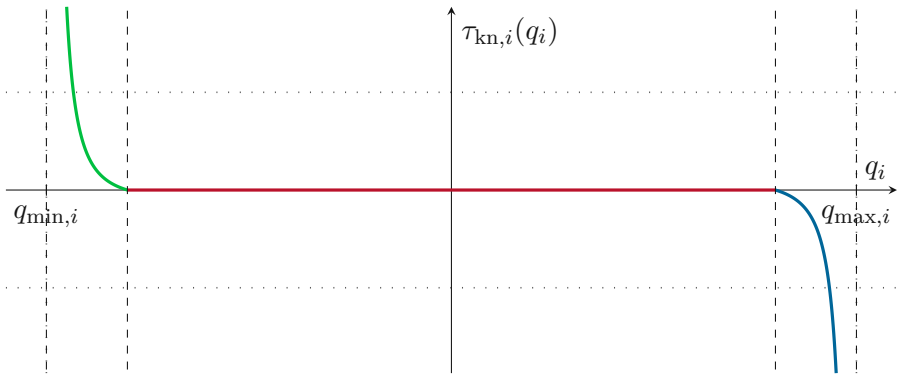


Figure 3.4: Nonlinear virtual spring force $\tau_{\mathrm{kn},i}(q_i)$, $i = 1, \ldots, 7$, acting as a barrier function on each robot joint in joint space.

4 User Interaction

In this work, two novel user interaction concepts based on PFC introduced in Chapter 3 are presented. These concepts are called *Path Snap-In* and *Path Switch*.

The Path Snap-In is similar to the gravity compensation mode. Hence, it lets the operator freely move the robot TCP in the workspace. As soon as the TCP is moved into the vicinity of a given path, it snaps on that path and locks in. In this way, often used targets (e. g. storage of tools) are easily and exactly reached. The idea originates from the line or edge snap functionality in conventional 2D or 3D CAD software.

The Path Switch enables online switching between multiple given paths by pushing the TCP towards the target path. At all times, the TCP is locked on and is moving along an active path. The control scheme is monitoring the applied external force. A belief system combines the active path with the external force and selects the most likely path according to the user intent. If the most likely path changes, a transition trajectory to the new path is generated online and the transition is initiated. The concept is useful for switching between multiple operating tasks in various order, controlled by the human operator.

In the following sections, the above concepts are explained in detail. First, the Path Snap-In, its idea and the control concept are introduced and a special case called Orientation Snap is discussed. Second, the Path Switch is described. This section covers the external force estimation, the belief system used to select the active path, as well as the generation of transitioning trajectories. The concepts introduced in this chapter are demonstrated in Chapter 5.

4.1 Path Snap-In

A widely used control scheme in HRI is the gravity compensation mode. As the name suggests, the gravitational forces $\mathbf{g}(\mathbf{q})$ are compensated by the robot and the TCP may be moved in the workspace by the operator by physical interaction with the robot. It is an intuitive mode of moving the TCP into a desired pose without needing to know any joint angles. This is especially useful during the teach-in of a robotic task. The flexibility comes with a lack of accuracy, since the control law does not proactively support the operator in his/her intentions.

The Path Snap-In mode mitigates this problem. It provides the flexibility of the gravity compensation and improves accuracy by assisting the operator in certain situations. The operator can freely move the TCP within the workspace, but as soon as the TCP comes close to a predefined path, the PFC takes over and the TCP is attracted towards this path. It stays locked on the path, until the operator pushes it away from the path. In that case, the TCP becomes free again. Figure 4.1 visualizes the concept. First, the TCP

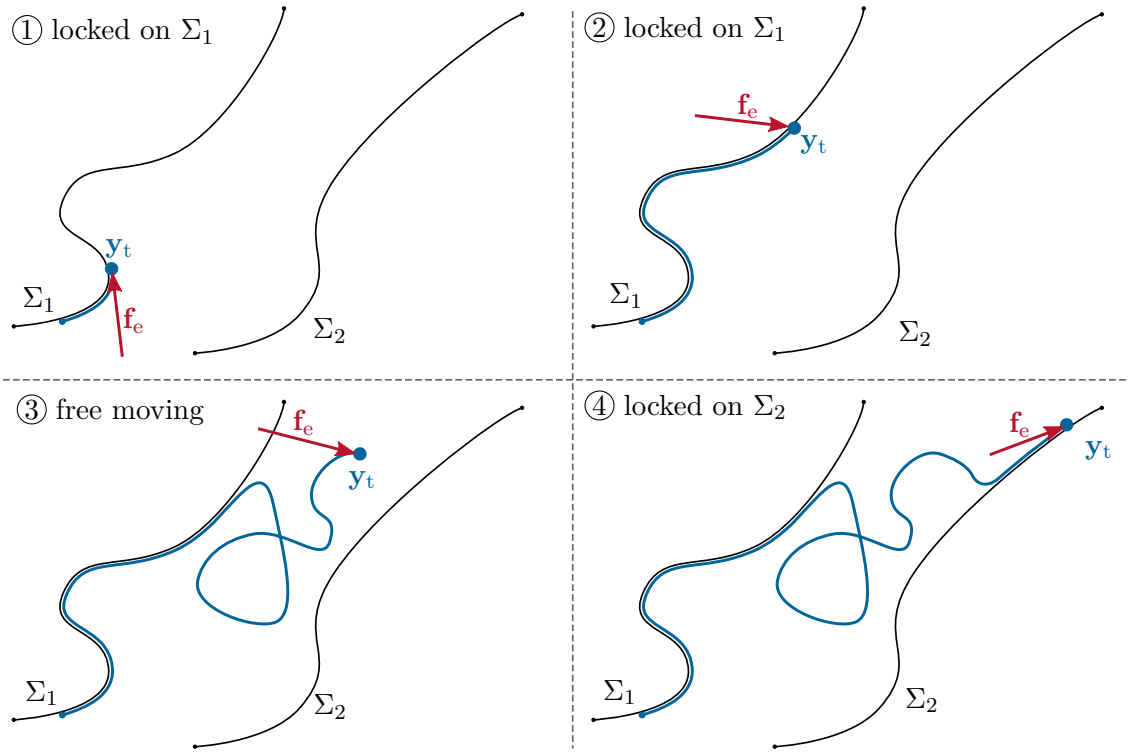


Figure 4.1: Path Snap-In concept with the TCP position $\mathbf{y}_t$ and the external force $\mathbf{f}_e$. The movement is depicted as blue line. In ① the TCP is moving along Σ_1. After pushing the TCP away from Σ_1 (②), the robot allows free gravity-compensated movement (③). In the vicinity of the path Σ_2, it locks in on this path (④).

is locked on path Σ_1 ①. Providing an external force tangential to the path, the TCP moves along the path Σ_1 ②. If the TCP is pushed away from the path strongly enough, the robot allows free gravity-compensated movement again ③. In vicinity of the next path Σ_2, the TCP locks on the path Σ_2 ④.

4.1.1 Control Concept

The control concept of Path Snap-In considers a set of paths $\{\Sigma_i, i \in \mathcal{I}\}$ according to definition (3.2) with an index set $\mathcal{I}$. Snapping in and out of a path is implemented by considering the virtual input $\mathbf{v}$ (3.38) for each path simultaneously. These multiple virtual inputs are weighted by a distance-dependent weighting factor μ, which selects the active path and eliminates the non-active ones. A weighted sum of virtual inputs is finally fed back into the PFC using (3.32). In order to stabilize the system distant from any path, an additional damping term is added. The individual steps are explained in the following. To update the weighting factor and select an active path, the shortest distance between TCP and each path is computed. Thus, the parallel transport frame $\mathcal{D}$ and the path-based coordinates $\boldsymbol{\xi}$ need to be computed for each path simultaneously. To distinguish the PFC for the different paths, the lower index $(.)_i$ is used, where i refers to the respective path.

Virtual Input Transformation

First, the virtual input $\mathbf{v}_i$ is calculated for each path $i \in \mathcal{I}$ according to (3.38). For each path, an individual optimal path parameter θ_i^* is found, which allows to construct the parallel transport frames $\mathcal{P}_i$ with the corresponding path-based coordinates $\boldsymbol{\xi}_i$. The virtual input $\mathbf{v}_i$ for each path $i \in \mathcal{I}$ reads as

$$\mathbf{v}_i = \begin{bmatrix} \mathbf{v}_{\xi,i} \\ \mathbf{v}_{\omega,i} \end{bmatrix} = \begin{bmatrix} \ddot{\boldsymbol{\xi}}_i^{\mathrm{d}} \\ \dot{\boldsymbol{\omega}}^{\mathcal{D}_i} \end{bmatrix} + \left(\mathbf{M}^{\mathrm{d}}\right)^{-1} \left(\hat{\mathbf{f}}_{\mathrm{e},i} - \mathbf{D}^{\mathrm{d}} \begin{bmatrix} \dot{\boldsymbol{\xi}}_i - \dot{\boldsymbol{\xi}}_i^{\mathrm{d}} \\ \boldsymbol{\omega}^{\mathcal{T}} - \boldsymbol{\omega}^{\mathcal{D}_i} \end{bmatrix} - \mathbf{K}^{\mathrm{d}} \begin{bmatrix} \boldsymbol{\xi}_i - \boldsymbol{\xi}_i^{\mathrm{d}} \\ \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i} \end{bmatrix} \right) . \tag{4.1}$$

Note that for the translational part, i. e. the first line of (4.1), the actual coordinates $\boldsymbol{\xi}_i$ and $\dot{\boldsymbol{\xi}}_i$ are different for each path $i \in \mathcal{I}$ because of the different parallel transport frames $\mathcal{P}_i$. For the rotational part, i. e. the second line of (4.1), $\boldsymbol{\omega}^{\mathcal{T}}$ is the same for all paths $i \in \mathcal{I}$, but the *desired* angular velocity $\boldsymbol{\omega}^{\mathcal{D}_i}$ and the *desired* angular acceleration $\dot{\boldsymbol{\omega}}^{\mathcal{D}_i}$ are specified individually for each path. The same applies to the quaternion error $\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}$. The vector $\mathbf{v}_{\xi,i}$ in (4.1) represents the virtual input in the path-based coordinate system $\mathcal{P}_i$. Since the individual virtual inputs $\mathbf{v}_i$ depend on the corresponding path-based frame $\mathcal{P}_i$, they cannot be compared directly.

To compare and sum up the virtual inputs of the paths, a common frame has to be chosen and every virtual input $\mathbf{v}_{\xi,i}$ must be transformed into this common frame. The base frame $\mathcal{B}$ is an obvious choice for this purpose and thus, the summation and actuation is formulated in $\mathcal{B}$. The virtual inputs $\mathbf{v}_{\xi,i}$ transformed into $\mathcal{B}$ are denoted by $\mathbf{v}_{\mathrm{t},i}$, $i \in \mathcal{I}$. The joint accelerations $\ddot{\mathbf{q}}_i^{\mathrm{t}}$ resulting from each individual virtual input $\mathbf{v}_{\xi,i}$ are calculated in the following. Based on (3.30) and (3.34), $\ddot{\mathbf{q}}_i^{\mathrm{t}}$ is written as

$$\ddot{\mathbf{q}}_i^{\mathrm{t}} = \mathbf{J}_{\xi,i}^{\dagger} \left(\mathbf{v}_{\xi,i} - \dot{\mathbf{J}}_{\xi,i} \dot{\mathbf{q}} \right) . \tag{4.2}$$

The joint acceleration due to $\mathbf{v}_{\xi,i}$ and $\mathbf{v}_{\mathrm{t},i}$ must be the same, since both virtual inputs are merely related by a transformation. Thus, $\ddot{\mathbf{q}}_i^{\mathrm{t}}$ can also be expressed using $\mathbf{v}_{\mathrm{t},i}$ and the translational geometric Jacobian $\mathbf{J}_{\mathrm{t}}$ from (2.6) as

$$\ddot{\mathbf{q}}_i^{\mathrm{t}} = \underbrace{\mathbf{J}_{\xi,i}^{\dagger} \left(\mathbf{v}_{\xi,i} - \dot{\mathbf{J}}_{\xi,i} \dot{\mathbf{q}} \right)}_{\text{path-based}} = \underbrace{\mathbf{J}_{\mathrm{t}}^{\dagger} \left(\mathbf{v}_{\mathrm{t},i} - \dot{\mathbf{J}}_{\mathrm{t}} \dot{\mathbf{q}} \right)}_{\text{Cartesian-based}} . \tag{4.3}$$

Solving for $\mathbf{v}_{\mathrm{t},i}$ gives the virtual input in the base frame $\mathcal{B}$

$$\begin{aligned} \mathbf{v}_{\mathrm{t},i} &= \dot{\mathbf{J}}_{\mathrm{t}} \dot{\mathbf{q}} + \mathbf{J}_{\mathrm{t}} \mathbf{J}_{\xi,i}^{\dagger} \left(\mathbf{v}_{\xi,i} - \dot{\mathbf{J}}_{\xi,i} \dot{\mathbf{q}} \right) \\ &= \mathbf{J}_{\mathrm{t}} \mathbf{J}_{\xi,i}^{\dagger} \mathbf{v}_{\xi,i} + \left(\dot{\mathbf{J}}_{\mathrm{t}} - \mathbf{J}_{\mathrm{t}} \mathbf{J}_{\xi,i}^{\dagger} \dot{\mathbf{J}}_{\xi,i} \right) \dot{\mathbf{q}} . \end{aligned} \tag{4.4}$$

Since $\mathcal{P}_i$ is a path-based moving frame depending on the path parameter θ_i^*, (4.4) is basically a transformation of $\mathbf{v}_{\xi,i}$ and $\mathbf{v}_{\mathrm{t},i}$ between moving frames. The term $\mathbf{J}_{\mathrm{t}} \mathbf{J}_{\xi,i}^{\dagger} \mathbf{v}_{\xi,i}$ is the static transformation and $\dot{\mathbf{J}}_{\mathrm{t}} \dot{\mathbf{q}} - \mathbf{J}_{\mathrm{t}} \mathbf{J}_{\xi,i}^{\dagger} \dot{\mathbf{J}}_{\xi,i} \dot{\mathbf{q}}$ is accounting for the relative movement between the frames. Equation (4.4) could also be formulated in the unit vectors $\left(\mathbf{e}_{\parallel}, \mathbf{e}_{\perp}, \mathbf{e}_{\pitchfork} \right)$ and

their time derivatives. However, since these vectors further depend on the time derivative of the path parameter $\dot{\theta}^*$, the formulation in (4.4) is more compact. Additionally, the Jacobians $\mathbf{J}_\mathrm{t}$ and $\mathbf{J}_{\xi,i}, i \in \mathcal{I}$, and their time derivatives are needed for the control concept later on.

For the virtual input of the angular acceleration $\mathbf{v}_{\omega,i}$, no transformation is needed, since it is already given in the base frame $\mathcal{B}$.

Weighting Factors

The virtual input $\mathbf{v}_i$ of each path is weighted with a weighting factor $\mu_i(d_i)$ based on the distance d_i to the path, $i \in \mathcal{I}$. In the context of the HRI concept Path Snap-In, a suitable weighting factor $\mu_i(d_i)$ should

- be 1 for small distances to the path $d_i \ll d_0$, with a suitable threshold distance $d_0 \in \mathbb{R}$,
- be 0 for large distances to the path $d_i \gg d_0$,
- and have a derivation of approximately 0 in the range around $d_i = 0$ to ensure that the applied force matches the original PFC in that range.

In order to satisfy these requirements, the weighting factor is constructed as a sigmoid function and is defined as

$$\mu_i(d_i) = 1 - \frac{1}{1 + \mathrm{e}^{-2c(d_i/d_0 - 1)}} \ , \tag{4.5}$$

where d_0 is the distance at which the weighting factor is reduced to $\mu_i(d_0) = 0.5$ and c specifies the slope of μ_i. A plot of $\mu_i(d_i)$ with different slope values c is shown in Figure 4.2. The sum of weighting factors μ should never exceed 1, i. e.

$$\mu = \sum_{i \in \mathcal{I}} \mu_i(d_i) \in [0, 1] \ . \tag{4.6}$$

Therefore, the parameters d_0 and c should be chosen such that there is no overlap between the volume of attraction of two paths, i. e. at maximum one path is attracting.

A translational weighting factor $\mu_{\mathrm{t},i}(d_{\mathrm{t},i})$ based on the translational distance $d_{\mathrm{t},i}$ and a rotational weighting factor $\mu_{\mathrm{r},i}(d_{\mathrm{r},i})$ based on the rotational distance $d_{\mathrm{r},i}$ are discussed in the following. Both weighting factors are defined according to (4.5), but with different distances $d_{\mathrm{t},i}$ and $d_{\mathrm{r},i}$ as well as different parameters $d_{0,\mathrm{t}}$, $d_{0,\mathrm{r}}$, c_t and c_r. The sum of the individual weighting factors according to (4.6) is denoted by μ_t for the translation and μ_r for the rotation. The translational distance $d_{\mathrm{t},i}$ is calculated as the Euclidean norm of the vector from the projected point on the i-th path $\boldsymbol{\sigma}_{\mathrm{t},i}(\theta_i^*)$ to the system output $\mathbf{y}_\mathrm{t}$, i. e. $d_{\mathrm{t},i} = \|\mathbf{y}_t - \boldsymbol{\sigma}_{\mathrm{t},i}(\theta_i^*)\|_2$. The rotational distance $d_{\mathrm{r},i}$ is computed from the quaternion error between two rotations and is chosen as

$$d_{\mathrm{r},i} = \left\| \boldsymbol{\varepsilon}_\mathcal{T}^{\mathcal{D}_i} \right\|_2 \tag{4.7}$$

$$\{e_\mathcal{T}^{\mathcal{D}_i}, \boldsymbol{\varepsilon}_\mathcal{T}^{\mathcal{D}_i}\} = \mathbf{Q}^{\mathcal{D}_i}(\theta_i^*) \otimes \left(\mathbf{Q}^\mathcal{T}\right)^{-1} \ . \tag{4.8}$$

Depending on the use case, the translational and rotational parts may be controlled as two separate and decoupled systems or as a combined system. If it is controlled separately, the position of the TCP can be locked in, while it is freely rotatable and vice versa. If it is controlled as a combined system, a lock-in of the position enforces also a lock-in of the orientation.

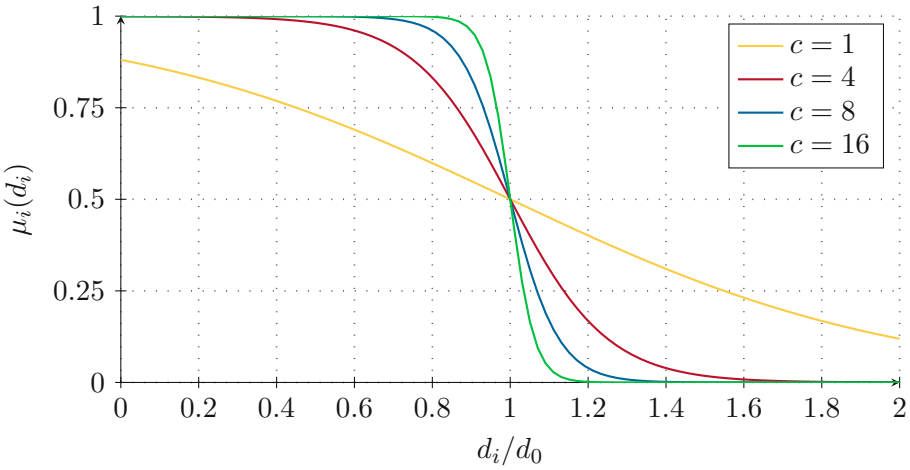


Figure 4.2: Weighting factor $\mu_i(d_i)$ as a function of the distance d_i for different slope values c.

Damping

When the TCP is distant from any path, it is not locked in and its behavior is specified and controlled using a separate controller. In this case, each weighting factor is close to zero, $\mu_i(d_i) \approx 0$, $\forall i \in \mathcal{I}$, and no force is fed back from any PFC. Consequently, the TCP behaves like a free-floating mass. That would work in theory, but since the controller is acting on a real system, an additional damping term $\mathbf{v}_{\text{dam}}$ is introduced to ensure closed-loop stability. To dampen the Cartesian movement, $\mathbf{v}_{\text{dam}}$ is constructed as

$$\mathbf{v}_{\text{dam}} = -\nu(\mu)\mathbf{D}_{\text{dam}} \begin{bmatrix} \dot{\mathbf{y}}_{\text{t}} \\ \boldsymbol{\omega}^{\mathcal{T}} \end{bmatrix} \tag{4.9}$$

with a positive definite damping matrix $\mathbf{D}_{\text{dam}} \in \mathbb{R}^{6\times 6}$ and the weighting factor for the damping $\nu(\mu)$. The damping term (4.9) should be active, if the TCP is *not* in the vicinity of any path, and be zero, if the TCP is in the vicinity of any path. If no path is attracting, $\mu \approx 0$ and $\nu(0) \overset{!}{=} 1$. If any path is attracting, $\mu \approx 1$ and $\nu(1) \overset{!}{=} 0$, see Figure 4.3. Hence, the requirements are similar to the requirements of the weighting factor $\mu_i(d_i)$ and therefore the weighting factor for the damping $\nu(\mu)$ is constructed similar to (4.5) as

$$\nu(\mu) = 1 - \frac{1}{1 + \mathrm{e}^{-2c_{\text{dam}}(\mu/0.5-1)}} \tag{4.10}$$

with $c_{\text{dam}} \in \mathbb{R}$ specifying the slope of $\nu(\mu)$ around $\mu = 0.5$. If the translation and rotation are controlled as two separate systems, the damping term $\mathbf{v}_{\text{dam}}$ splits into $\mathbf{v}_{\text{dam,t}}$ and

$\mathbf{v}_{\mathrm{dam,r}}$. The weighting factor for the damping $\nu(\mu)$ then splits into a translational part $\nu_{\mathrm{t}}(\mu_{\mathrm{t}})$ and a rotational part $\nu_{\mathrm{r}}(\mu_{\mathrm{r}})$, accordingly. In that case, instead of the damping matrix $\mathbf{D}_{\mathrm{dam}} \in \mathbb{R}^{6\times6}$, a translational damping matrix $\mathbf{D}_{\mathrm{dam,t}} \in \mathbb{R}^{3\times3}$ and a rotational damping matrix $\mathbf{D}_{\mathrm{dam,r}} \in \mathbb{R}^{3\times3}$ are utilized.

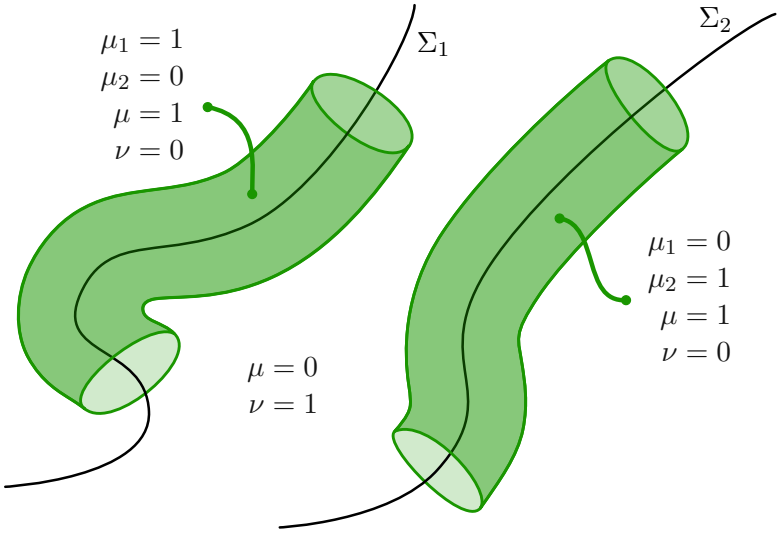


Figure 4.3: Two paths Σ_1 and Σ_2 with the corresponding weighting factors $\mu_1(d_1)$ and $\mu_2(d_2)$, the sum $\mu = \mu_1(d_1) + \mu_2(d_2)$ and the weighting factor for the damping $\nu(\mu)$. In a tube around each path, the corresponding path is active and its weighting factor $\mu_i \approx 1$. If the TCP is not inside any tube, $\nu(\mu) \approx 1$ holds.

Combining and Feeding Back the Virtual Inputs

The virtual input applied to the system is calculated as a weighted sum of the individual virtual inputs of the PFCs $\mathbf{v}_i$, $i \in \mathcal{I}$ plus the damping term $\mathbf{v}_{\mathrm{dam}}$ from (4.9). If the translational and the rotational part of the system should lock in simultaneously, the virtual inputs are combined as

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_{\mathrm{t}} \\ \mathbf{v}_{\omega} \end{bmatrix} = \sum_{i\in\mathcal{I}} \left(\mu_i(d_i) \begin{bmatrix} \mathbf{v}_{\mathrm{t},i} \\ \mathbf{v}_{\omega,i} \end{bmatrix} \right) - \nu(\mu)\mathbf{D}_{\mathrm{dam}} \begin{bmatrix} \dot{\mathbf{y}}_{\mathrm{t}} \\ \boldsymbol{\omega}^{\mathcal{T}} \end{bmatrix} . \tag{4.11}$$

In contrast, for a system where the translational snap-in and the rotational snap-in are decoupled, the combined virtual input $\mathbf{v}$ is chosen as

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_{\mathrm{t}} \\ \mathbf{v}_{\omega} \end{bmatrix} = \sum_{i\in\mathcal{I}} \left(\begin{bmatrix} \mu_{\mathrm{t},i}(d_{\mathrm{t},i})\mathbf{v}_{\mathrm{t},i} \\ \mu_{\mathrm{r},i}(d_{\mathrm{r},i})\mathbf{v}_{\omega,i} \end{bmatrix} \right) - \begin{bmatrix} \nu_{\mathrm{t}}(\mu_{\mathrm{t}})\mathbf{D}_{\mathrm{dam,t}}\dot{\mathbf{y}}_{\mathrm{t}} \\ \nu_{\mathrm{r}}(\mu_{\mathrm{r}})\mathbf{D}_{\mathrm{dam,r}}\boldsymbol{\omega}^{\mathcal{T}} \end{bmatrix} . \tag{4.12}$$

In both cases, the combined virtual input $\mathbf{v}$ in the base frame is inserted in the exact input-output linearization (3.32) as

$$\boldsymbol{\tau} = \mathbf{n} + \mathbf{M}\mathbf{J}^{\dagger} \left(\begin{bmatrix} \mathbf{v}_{\mathrm{t}} \\ \mathbf{v}_{\omega} \end{bmatrix} - \dot{\mathbf{J}}\dot{\mathbf{q}} \right) + \boldsymbol{\tau}_{\mathrm{n}} - \boldsymbol{\tau}_{\mathrm{e}} . \tag{4.13}$$

Note that in (4.13) and in the following, the function arguments are omitted to improve the clarity of presentation. It is worth mentioning that the nullspace control $\boldsymbol{\tau}_\mathrm{n}$ in (4.13) is invariant w.r.t. a transformation to or from a path-based frame. Analogous to (3.41), the nullspace control $\boldsymbol{\tau}_{\mathrm{n},i}$ for each PFC is given by

$$\boldsymbol{\tau}_{\mathrm{n},i} = \mathbf{M}\Big(\mathbf{I} - \hat{\mathbf{J}}_i^\dagger \hat{\mathbf{J}}_i\Big)\mathbf{v}_\mathrm{n} \ . \tag{4.14}$$

To further evaluate (4.14), a relation for the pseudoinverse of the path-based Jacobian $\hat{\mathbf{J}}^\dagger$ is needed, which is found as

$$\hat{\mathbf{J}}^\dagger = \mathbf{J}^\dagger \begin{bmatrix} \frac{1}{\beta}\mathbf{e}_\parallel & \mathbf{e}_\perp & \mathbf{e}_\pitchfork & \mathbf{0} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I} \end{bmatrix} . \tag{4.15}$$

The proof of (4.15) is given in Appendix A.1.1. With (4.15), equation (4.14) simplifies to

$$\begin{aligned}
\boldsymbol{\tau}_{\mathrm{n},i} &= \mathbf{M}(\mathbf{I} - \hat{\mathbf{J}}_i^\dagger \hat{\mathbf{J}}_i)\mathbf{v}_\mathrm{n} \\
&= \mathbf{M}\left(\mathbf{I} - \mathbf{J}^\dagger \begin{bmatrix} \frac{1}{\beta_i}\mathbf{e}_{\parallel,i} & \mathbf{e}_{\perp,i} & \mathbf{e}_{\pitchfork,i} & \mathbf{0} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \beta_i \mathbf{e}_{\parallel,i}^\mathrm{T} & \mathbf{0}_{1\times3} \\ \mathbf{e}_{\perp,i}^\mathrm{T} & \mathbf{0}_{1\times3} \\ \mathbf{e}_{\pitchfork,i}^\mathrm{T} & \mathbf{0}_{1\times3} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{J} \right) \mathbf{v}_\mathrm{n} \\
&= \mathbf{M}\left(\mathbf{I} - \mathbf{J}^\dagger \begin{bmatrix} \mathbf{e}_{\parallel,i}\mathbf{e}_{\parallel,i}^\mathrm{T} + \mathbf{e}_{\perp,i}\mathbf{e}_{\perp,i}^\mathrm{T} + \mathbf{e}_{\pitchfork,i}\mathbf{e}_{\pitchfork,i}^\mathrm{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{J} \right) \mathbf{v}_\mathrm{n} \\
&= \mathbf{M}\Big(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}\Big)\mathbf{v}_\mathrm{n} \\
&= \boldsymbol{\tau}_\mathrm{n} \ ,
\end{aligned} \tag{4.16}$$

which again yields the nullspace control given in the base frame $\boldsymbol{\tau}_\mathrm{n}$. The nullspace control can thus be designed separately and without considering the Path Snap-In functionality at all.

Parameter Choice

For the Path Snap-In functionality and its usability, the choice of the parameters c and d_0 in the weighting functions $\mu_i(d_i), i \in \mathcal{I}$, is crucial. Since a human operator must be able to push the TCP away from a locked-in path, the maximum attracting force generated by the Path Snap-In controller must be limited. A given force magnitude of $f_\mathrm{h,t}$ for translation and $f_\mathrm{h,r}$ for rotation are taken as reference values for the maximum admissible human force. Thus, the force exerted by the controller is not allowed to exceed $f_\mathrm{h,t}$ or $f_\mathrm{h,r}$. This is only relevant for the force acting transversal to the path ($\perp$ and $\pitchfork$) because the operator is pushing the TCP *away* from the path. While the virtual inertial and virtual damper force are only acting against an active motion, the virtual spring force is causing the attracting force and is investigated in the following. In other words, the operator will eventually be able to overcome the inertial and damper force, since they do not depend on the position. However, the spring force increases with an increasing distance to the

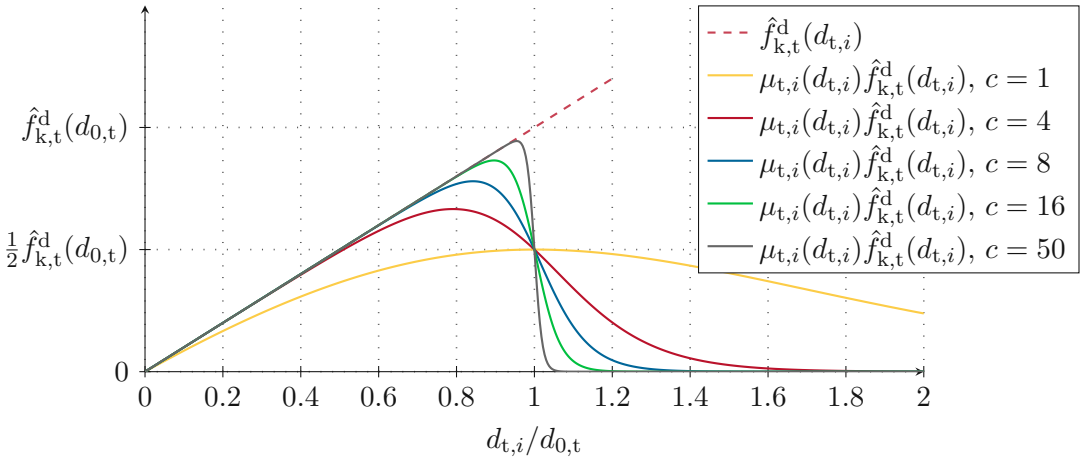


Figure 4.4: Transversal spring force $\hat{f}^{\mathrm{d}}_{\mathrm{k,t}}(d_{\mathrm{t},i})$ weighted with the weighting factor $\mu_{\mathrm{t},i}(d_{\mathrm{t},i})$ plotted over the distance $d_{\mathrm{t},i}/d_{0,\mathrm{t}}$ for different slope values c.

path. To push the TCP out of the path attraction zone, cf. Figure 4.3, the virtual spring force has to be overcome. In the coordinates of the i-th path, it is given by, see (3.35),

$$\hat{\mathbf{f}}^{\mathrm{d}}_{\mathrm{k}} = \begin{bmatrix} \hat{\mathbf{f}}^{\mathrm{d}}_{\mathrm{k,t}} \\ \hat{\mathbf{f}}^{\mathrm{d}}_{\mathrm{k,r}} \end{bmatrix} = -\mathbf{K}^{\mathrm{d}} \begin{bmatrix} \boldsymbol{\xi}_i - \boldsymbol{\xi}^{\mathrm{d}}_i \\ \boldsymbol{\varepsilon}^{\mathcal{D}_i}_{\mathcal{T}} \end{bmatrix} . \tag{4.17}$$

Note that only the transversal part is of importance. Assuming $\mathbf{K}^{\mathrm{d}}$ is a diagonal matrix, i. e. $\mathbf{K}^{\mathrm{d}} = \mathrm{diag}\left(k^{\mathrm{d}}_{\parallel}, k^{\mathrm{d}}_{\perp}, k^{\mathrm{d}}_{\pitchfork}, k^{\mathrm{d}}_{\mathrm{r}}, k^{\mathrm{d}}_{\mathrm{r}}, k^{\mathrm{d}}_{\mathrm{r}}\right)$, and $k^{\mathrm{d}}_{\perp} = k^{\mathrm{d}}_{\pitchfork}$ so that the force in the transversal plane is direction-independent, the magnitudes of the transversal spring force $\hat{f}^{\mathrm{d}}_{\mathrm{k,t}}(d_{\mathrm{t}})$ and rotational spring force $\hat{f}^{\mathrm{d}}_{\mathrm{k,r}}(d_{\mathrm{r}})$ yield

$$\hat{f}^{\mathrm{d}}_{\mathrm{k,t}}(d_{\mathrm{t}}) = \left\| \hat{\mathbf{f}}^{\mathrm{d}}_{\mathrm{k,t}}(d_{\mathrm{t}}) \right\|_2 = k^{\mathrm{d}}_{\perp} \left\| \mathbf{y}_{\mathrm{t}} - \boldsymbol{\sigma}_{\mathrm{t}}(\theta^*) \right\|_2 = k^{\mathrm{d}}_{\perp} d_{\mathrm{t}} \tag{4.18}$$

$$\hat{f}^{\mathrm{d}}_{\mathrm{k,r}}(d_{\mathrm{r}}) = \left\| \hat{\mathbf{f}}^{\mathrm{d}}_{\mathrm{k,r}}(d_{\mathrm{r}}) \right\|_2 = k^{\mathrm{d}}_{\mathrm{r}} \left\| \boldsymbol{\varepsilon}^{\mathcal{D}}_{\mathcal{T}} \right\|_2 = k^{\mathrm{d}}_{\mathrm{r}} d_{\mathrm{r}} . \tag{4.19}$$

As an example, Figure 4.4 shows the magnitude of the transversal spring force $\hat{f}^{\mathrm{d}}_{\mathrm{k,t}}(d_{\mathrm{t},i})$ as well as the weighted magnitudes $\mu_{\mathrm{t},i}(d_{\mathrm{t},i})\hat{f}^{\mathrm{d}}_{\mathrm{k,t}}(d_{\mathrm{t},i})$ with different slope values c. It shows that the maximum force for $c \to \infty$ is $\hat{f}^{\mathrm{d}}_{\mathrm{k,t}}(d_{0,\mathrm{t}})$. For the rotational spring force, this corresponds to $\hat{f}^{\mathrm{d}}_{\mathrm{k,r}}(d_{0,\mathrm{r}})$, see (4.19). By choosing $d_{0,\mathrm{t}}$ and $d_{0,\mathrm{r}}$ such that $\hat{f}^{\mathrm{d}}_{\mathrm{k,t}}(d_{0,\mathrm{t}}) < f_{\mathrm{h,t}}$ and $\hat{f}^{\mathrm{d}}_{\mathrm{k,r}}(d_{0,\mathrm{r}}) < f_{\mathrm{h,r}}$, it is guaranteed that for any chosen c, the transversal spring force (4.18) never exceeds $f_{\mathrm{h,t}}$ and the rotational spring force (4.19) never exceeds $f_{\mathrm{h,r}}$. This results in an upper limit for the reference distance $d_{0,\mathrm{t}}$ and $d_{0,\mathrm{r}}$ of

$$d_{0,\mathrm{t}} < \frac{f_{\mathrm{h,t}}}{k^{\mathrm{d}}_{\perp}} \tag{4.20a}$$

$$d_{0,\mathrm{r}} < \frac{f_{\mathrm{h,r}}}{k^{\mathrm{d}}_{\mathrm{r}}} . \tag{4.20b}$$

The slope parameter c can be arbitrarily chosen, since (4.20) already ensures that the safety limits for the interaction with the robot are respected. Selecting a smaller slope value c makes the transition between free moving and locked-in operation smoother. However, for a low value of c, e. g., $c = 1$, the weighting factor $\mu_i(d_i) = 1$ is never reached, see Figure 4.2. Consequently, when the slope parameter c is selected too low, the desired behavior of the PFC is not achieved even in close vicinity of a path, see also Figure 4.4. Selecting a larger slope value c makes the transition between free moving and locked-in operation on a path more abrupt. In HRI, an abrupt change in behavior of the robot should be avoided. Thus, the slope parameter c should not be selected too large. According to Figure 4.2 and Figure 4.4, a value of $c = 4$ is a reasonable compromise. Since different operators may have different preferences, the slope parameter c may be further tuned to adapt the behavior of the robot.

4.1.2 Orientation Snap

The teach-in of a robotic task is a possible use case for the Path Snap-In. Conventional methods based on teach pendants are precise, but lack flexibility. Methods based on gravity compensation on the other hand are more flexible, but lack precision. The proposed Path Snap-In closes the gap between both mentioned teach-in methods. It is flexible because the user may freely move the TCP in the workspace of the robot and it is precise when snapped in on any given path.

Instead of using geometric paths in Cartesian space, a given set of paths may also be a set of different predefined orientations in the general workspace of the robot. In such a way, the operator may for example move the TCP close to the vertical orientation, from which the TCP snaps precisely into the vertical orientation. In this work, this HRI function is referred to as Orientation Snap and is derived as a special case of the Path Snap-In. The Orientation Snap is introduced by defining a set of predefined orientations, such that the TCP locks into these predefined orientations. The goal is to have an orientation raster that is equally spaced w.r.t. application-specific Cartesian orientation angles. In that case, the translational and rotational snap are controlled separately, i. e. $\mu_{\mathrm{t},i} \neq \mu_{\mathrm{r},i}$, see (4.12). In the special case of the Orientation Snap, there is no projected path position $\boldsymbol{\sigma}_{\mathrm{t},i}(\theta_i^*)$ in which the TCP could possibly lock in. Additionally, the translational weighting factor μ_{t} is set to zero, which yields a virtual translational input $\mathbf{v}_{\mathrm{t}} = \mathbf{0}$. For the rotational weighting factors $\mu_{\mathrm{r},i}$, $i \in \mathcal{I}$, the set of valid orientations must be defined. The orientations are chosen based on the classical Euler angles. A single orientation is thus given by

$$\mathbf{R}^{\mathcal{D}}(\varphi, \vartheta, \psi) = \mathbf{R}_{\mathrm{z},\varphi}\mathbf{R}_{\mathrm{y},\vartheta}\mathbf{R}_{\mathrm{z},\psi} \ , \tag{4.21}$$

with the rotation matrices

$$\mathbf{R}_{\mathrm{y},\vartheta} = \begin{bmatrix} \cos\vartheta & 0 & \sin\vartheta \\ 0 & 1 & 0 \\ -\sin\vartheta & 0 & \cos\vartheta \end{bmatrix} \quad \text{and} \quad \mathbf{R}_{\mathrm{z},\varphi} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{4.22}$$

The three Euler angles φ, ϑ and ψ are used to parametrize the TCP orientation. With this parametrization, the angles φ and ϑ orient the z-axis of the TCP, while ψ parametrizes the

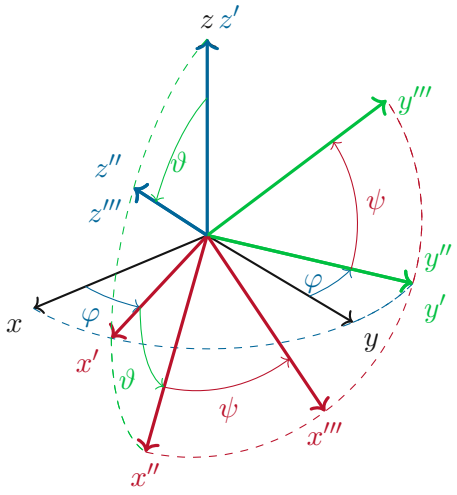
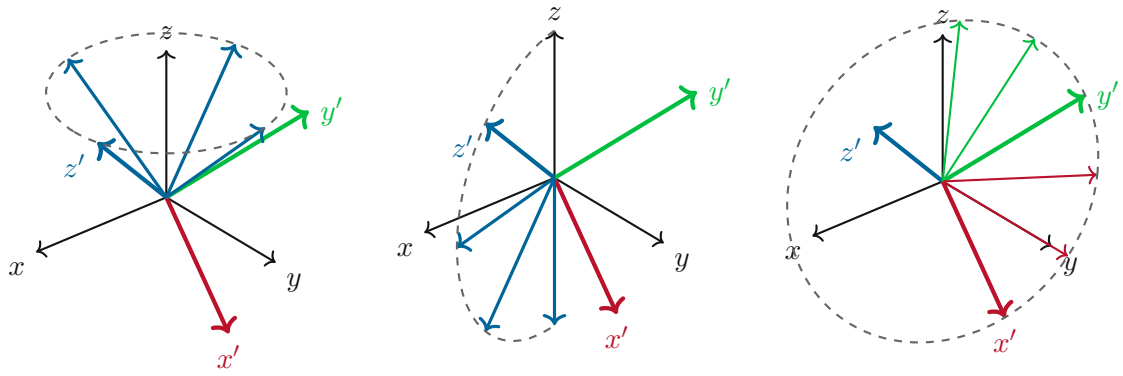


Figure 4.5: Euler angle parametrization of the TCP.

rotation around the z-axis of the TCP, see Figure 4.5. To define a set of fixed orientations to snap in, the range of the angular parameters $\varphi \in [0, 2\pi)$, $\vartheta \in [0, \pi]$ and $\psi \in [0, 2\pi)$ is discretized with an equidistant grid with the intervals $\Delta\varphi$, $\Delta\vartheta$ and $\Delta\psi$. With N_φ, N_ϑ and N_ψ specifying the number of orientations for each angle, the intervals $\Delta\varphi$ and $\Delta\psi$ are calulated by $\Delta\varphi = \frac{2\pi}{N_\varphi}$ and $\Delta\psi = \frac{2\pi}{N_\psi}$. The set $\{\mathbf{R}_{\mathrm{z},i\Delta\varphi} | i = 0, \ldots, N_\varphi - 1\}$ defines a set of equally spaced rotations for φ. Similarly, the set for ψ results in $\{\mathbf{R}_{\mathrm{z},k\Delta\psi} | k = 0, \ldots, N_\psi - 1\}$. The angle ϑ has a reduced range of π instead of 2π. Since 0 and π represent different rotations, they must both be included in the set of valid orientations for ϑ. For a number of N_ϑ orientations, the range $[0, \pi]$ is split into $N_\vartheta - 1$ intervals and this results in $\Delta\vartheta = \frac{\pi}{N_\vartheta - 1}$. This gives a set of orientations for ϑ of $\{\mathbf{R}_{\mathrm{y},j\Delta\vartheta} | j = 0, \ldots, N_\vartheta - 1\}$. Combining the sets of orientations for the individual axes φ, ϑ and ψ according to the Euler representation of (4.21) yields the set of all valid orientations for the Orientation Snap

$$\mathcal{R}_{\mathrm{os}} = \Big\{ \mathbf{R}^{\mathcal{D}_{i,j,k}} = \mathbf{R}_{\mathrm{z},i\Delta\varphi} \mathbf{R}_{\mathrm{y},j\Delta\vartheta} \mathbf{R}_{\mathrm{z},k\Delta\psi} \,|\, i = 0, \ldots, N_\varphi - 1, \; j = 0, \ldots, N_\vartheta - 1, \; k = 0, \ldots, N_\psi - 1 \Big\} . \tag{4.23}$$

Figure 4.6 shows the influence of each angular step. A change in $\Delta\varphi$ causes a rotation of the TCP around the vertical axis, a change in $\Delta\vartheta$ causes a rotation of the TCP around the horizontal axis and a change in $\Delta\psi$ causes a rotation of the TCP around the z-axis of the TCP. For $\vartheta = j\pi$, $j \in \mathbb{N}$, the rotations $\mathbf{R}_{\mathrm{z},\psi}$ and $\mathbf{R}_{\mathrm{z},\varphi}$ rotate around the same axis. This is per se no problem, since the rotations serve different purposes (aligning the TCP normal vs rotating the TCP) and the intervals may be different $\Delta\varphi \neq \Delta\psi$. However, it must be ensured that each orientation is unique, otherwise (4.6) cannot be satisfied. Therefore, each element of (4.23) that occurs more than once must be eliminated from $\mathcal{R}_{\mathrm{os}}$.



(a) A change in $\Delta\varphi$ causes a rotation of the TCP around the vertical axis.

(b) A change in $\Delta\vartheta$ causes a rotation of the TCP around the horizontal axis.

(c) A change in $\Delta\psi$ causes a rotation of the TCP around the z-axis of the TCP.

Figure 4.6: Change of orientation caused by the angular steps $\Delta\varphi$, $\Delta\vartheta$ and $\Delta\psi$.

The combination of the general Path Snap-In mode with the Orientation Snap builds the foundation for a more intuitive robot programming interface. Experiments validating the proposed concepts follow in Chapter 5.

4.2 Path Switch

With the introduced Path Snap-In concept, the operator can snap into different paths by moving the TCP into their vicinity and freely move the TCP in between. This is suitable for the teach-in of new tasks and for assisting a manual human operation. The downside is the limited force that can be applied by the robot to keep the TCP on the path, since a human operator must be able to push the robot away from the path it is currently locked into. To mitigate this downside, another mode called *Path Switch* is introduced in this section.

The general idea of Path Switch is that during a normal operation exactly one path is active at all times. By monitoring the external force applied to the TCP, the controller estimates the desired human action. If the TCP is pushed in the direction of a new path, the intent to switch to the new path is recognized. A belief system accumulates this intent while staying on the currently active path. If the accumulated intent to switch to the new path exceeds a given threshold, a transition trajectory is generated. The TCP follows the transition trajectory to the new path, after which the PFC of the new path becomes active. Figure 4.7 illustrates such a transition. First, the TCP moves along the path Σ_1. The applied external force, and thereby the intention of the operator, is accumulated by the belief system ①. Since the applied force stays active for a longer period of time, the belief system assumes that the operator wants to switch to Σ_2 and a transition trajectory is planned ②. The TCP moves along the transition trajectory ③ and after completing the transition, the PFC of the new active path Σ_2 becomes active again ④.

In this section, the control concept for the Path Switch is explained in detail. First,

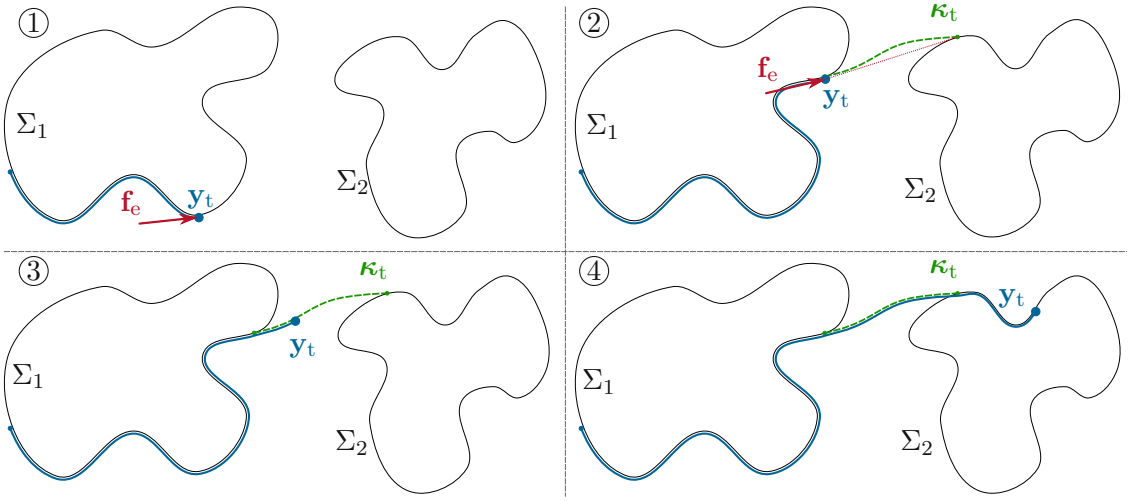


Figure 4.7: Path Switch concept with the TCP position $\mathbf{y}_t$ and the external force $\mathbf{f}_e$. The motion is depicted as blue line. In ①, the TCP is moving along Σ_1 while being pushed towards Σ_2. Due to the external force $\mathbf{f}_e$, the belief system is updated in the background. The external force is maintained until a transition to Σ_2 is initiated (②). The TCP moves along the transition trajectory $\boldsymbol{\kappa}_t$ (③) and moves along Σ_2 after finishing the transition (④).

a detection algorithm for the intention of switching to a new path is introduced. This is done by estimating the external force, geometrically searching for suitable paths that might be considered as possible switching candidates and updating the belief system responsible for the selection of the active path. Second, the generation of the transition trajectory is discussed. The general shape of the trajectory and the needed boundary conditions for a smooth transition between two paths is explained. The experimental validation of the concepts introduced in this chapter is given in Chapter 5.

4.2.1 Detection of the Intention to Switch

To react to the input of the human operator, the corresponding human intent must be estimated. To detect the intention of the operator to switch to a new path, multiple steps are involved. First, the applied force is estimated. The external force direction indicates which path the human wants to switch to. Second, based on that information, a belief system selects the most likely path and is responsible for initiating the transition to the new path.

External Force Estimation

It is assumed that the robot has no additional F/T sensor attached to its TCP. To estimate the external force applied to the TCP, the measurements of the joint angles, joint velocities and torques are utilized. The external torque is estimated via a momentum observer, see [33].

The generalized momentum $\mathbf{p}(t)$ is given by $\mathbf{p}(t) = \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}$. The time derivative of the generalized momentum $\dot{\mathbf{p}}(t)$ yields

$$\begin{aligned}\dot{\mathbf{p}}(t) &= \dot{\mathbf{M}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} \\ &= \mathbf{C}^{\mathrm{T}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau} + \boldsymbol{\tau}_{\mathrm{e}} \ ,\end{aligned} \tag{4.24}$$

by using (2.9) and the identity $\dot{\mathbf{M}}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{C}^{\mathrm{T}}(\mathbf{q},\dot{\mathbf{q}})$ (see, e. g., [33]). Since the external torque $\boldsymbol{\tau}_{\mathrm{e}}$ is not known, (4.24) can not be evaluated directly and an observer is used to estimate the generalized momentum $\mathbf{p}^{\mathrm{est}}(t)$ as well as the external torque $\boldsymbol{\tau}_{\mathrm{e}}^{\mathrm{est}}$. Choosing the observer dynamics as

$$\dot{\mathbf{p}}^{\mathrm{est}}(t) = \mathbf{C}^{\mathrm{T}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau} + \boldsymbol{\tau}_{\mathrm{e}}^{\mathrm{est}} \tag{4.25a}$$

$$\boldsymbol{\tau}_{\mathrm{e}}^{\mathrm{est}} = \mathbf{K}_{\mathrm{O}}\Big(\mathbf{p}(t) - \mathbf{p}^{\mathrm{est}}(t)\Big) \ , \tag{4.25b}$$

with the positive definite observer matrix $\mathbf{K}_{\mathrm{O}} \in \mathbb{R}^{7\times 7}$, yields the dynamics of the external torque estimation

$$\dot{\boldsymbol{\tau}}_{\mathrm{e}}^{\mathrm{est}} = \mathbf{K}_{\mathrm{O}}\Big(\dot{\mathbf{p}}(t) - \dot{\mathbf{p}}^{\mathrm{est}}(t)\Big) = \mathbf{K}_{\mathrm{O}}\Big(\boldsymbol{\tau}_{\mathrm{e}} - \boldsymbol{\tau}_{\mathrm{e}}^{\mathrm{est}}\Big) \ . \tag{4.26}$$

Thus, the momentum observer (4.25) provides a linear first-order estimation of the external torque $\boldsymbol{\tau}_{\mathrm{e}}^{\mathrm{est}}$.

In order to use the estimated force for finding path switching candidates, the force direction in the task space is needed. The estimated force in the base frame $\mathcal{B}$ is calculated as the projection of the estimated torque $\boldsymbol{\tau}_{\mathrm{e}}^{\mathrm{est}}$, yielding

$$\mathbf{f}_{\mathrm{e}}^{\mathrm{est}} = \Big(\mathbf{J}^{\dagger}(\mathbf{q})\Big)^{\mathrm{T}}\boldsymbol{\tau}_{\mathrm{e}}^{\mathrm{est}} = \begin{bmatrix}\mathbf{f}_{\mathrm{e,t}}^{\mathrm{est}} \\ \mathbf{f}_{\mathrm{e,r}}^{\mathrm{est}}\end{bmatrix} \ . \tag{4.27}$$

The direction of the external force $\mathbf{e}_{\mathrm{f}}$ is calculated by dividing the force by its magnitude

$$\mathbf{e}_{\mathrm{f}} = \frac{\mathbf{f}_{\mathrm{e,t}}^{\mathrm{est}}}{\left\|\mathbf{f}_{\mathrm{e,t}}^{\mathrm{est}}\right\|_2} \ . \tag{4.28}$$

In this work, a minimum force of $\left\|\mathbf{f}_{\mathrm{e,t}}^{\mathrm{est}}\right\|_2 = 5\,\mathrm{N}$ is needed to be considered as a valid applied force. Otherwise, it is assumed that no exernally applied force is present.

Search for new Path Candidates

A belief system is responsible for choosing an active path and is introduced in this section. The belief system is updated according to the human intent. If the human operator wants to switch to a new path, the TCP is pushed in the direction where the new path is found. Successively, any suitable path that could be intended by the human operator has to be found based on the direction of the external force $\mathbf{e}_{\mathrm{f}}$. Ideally, the push direction and the selected path would intersect at some point. In reality, the push direction will never be that exact. Thus, a path is defined to be a candidate, if it lies within a cone with

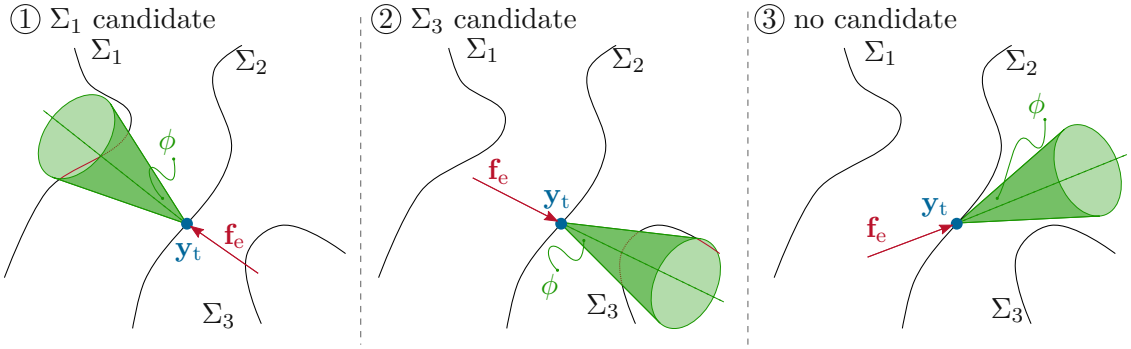


Figure 4.8: Search for new path candidates depending on the external force $\mathbf{f}_e$. The TCP $\mathbf{y}_t$ is currently on Σ_2. In ①, $\mathbf{f}_e$ points in the direction of Σ_1, making Σ_1 a candidate for an intended switch. In ②, the same goes for Σ_3. In ③, no path is found in the cone with the opening angle of ϕ, resulting in 0 candidates.

an opening angle of ϕ around the push direction $\mathbf{e}_f$ with the cone tip at the position of the TCP $\mathbf{y}_t$, see Figure 4.8. For a path Σ_i defined by $\boldsymbol{\sigma}_{t,i}(\theta_i), i \in \mathcal{I}$, this condition is mathematically formulated as

$$\exists\, \theta_i \in \Theta_i \quad \text{s.t. } \mathbf{e}_f^T \mathbf{e}_{d,i} = \mathbf{e}_f^T \frac{\boldsymbol{\sigma}_{t,i}(\theta_i) - \mathbf{y}_t}{\|\boldsymbol{\sigma}_{t,i}(\theta_i) - \mathbf{y}_t\|_2} > \cos\phi \ , \tag{4.29}$$

with $\mathbf{e}_{d,i}$ being the normalized distance vector from the TCP position $\mathbf{y}_t$ to the point on the path $\boldsymbol{\sigma}_{t,i}(\theta_i)$. If the path is a valid candidate, the closest point to the actual TCP is chosen as the reference point on the new path. It is found as a solution of the constrained minimization problem

$$\begin{aligned} \bar{\theta}_i^* = \underset{\theta_i \in \Theta_i}{\arg\min} \, \|\mathbf{y}_t - \boldsymbol{\sigma}_{t,i}(\theta_i)\|_2^2, \quad i \in \mathcal{I} \\ \text{s.t. } \mathbf{e}_f^T \mathbf{e}_{d,i} > \cos\phi \ . \end{aligned} \tag{4.30}$$

Hence, the optimal path parameter $\bar{\theta}_i^*$ satisfies (4.29) while minimizing the distance. To distinguish the solution of the constrained minimization problem (4.30) from the solution of the unconstrained minimization problem for the optimal path parameter θ^* (3.8), the optimal path parameter with constraints is denoted by an additional bar, i. e. $\bar{\theta}^*$. If a solution of the minimization problem (4.30) can be found, it is considered a candidate for a switch. Otherwise, it is omitted during the path selection. The admissible paths are selected by their indices and are stored in the set

$$\mathcal{J} = \left\{ j \,|\, j \in \mathcal{I}, \exists\, \bar{\theta}_j^* \text{ and } j \neq i_{act} \right\} \subset \mathcal{I} \ . \tag{4.31}$$

The index of the active path i_{act} is excluded from the set $\mathcal{J}$, since it is not an admissible option for a switching path.

Belief System

The belief system is used to select the active path and detect the operator's intent to switch to a new path. It is inspired by the work of Khoramshahi and Billard [20]. It is

assumed that the index set $\mathcal{I}$ is given as $\mathcal{I} = \{1, 2, \ldots, n\}$, which simplifies the following notation. The belief system stores beliefs b_i for each path $\{b_i \in [0, 1] \,|\, i \in \mathcal{I}\}$, where the sum of all beliefs must be one, i. e. $\sum_{i=1}^{n} b_i = 1$. A belief of $b_i = 1$ means that the system beliefs by 100% that the path i is the intended one. A value of $b_i = 0$ means that the belief system is certain that the path i is *not* intended by the human operator. At each time step exactly one active path is selected and its index is denoted by i_{act}. The beliefs of all paths are collected in the belief vector $\mathbf{b}^{\mathrm{T}} = \begin{bmatrix} b_1 & b_2 & \ldots & b_n \end{bmatrix}$, which is updated in each time step. The main difference between the belief system in this work and the belief system in [20] is that the belief system presented here is force-based, while the belief system in [20] is distance-based. The force-based system aggregates the intentions (provided by externally applied forces) in the background, while staying fixed on the active path. In distance-based approaches, however, a deviation from the path is needed to know that the intention has changed.

If the belief of a non-active path is rising, it is likely that the operator wants to switch to this path. If the belief of a path exceeds a given belief threshold b_{th} and the corresponding path is not active $i \neq i_{\text{act}}$, a transition is initiated. In the next section, the needed transition trajectory is generated.

The belief system distinguishes two cases. In the first case, no external force is applied. In that case, the best guess is comparing the distances between the TCP and each path. The closest path is most likely the intended path, and its belief thus increases the most. In the second case, an external force is applied. In that case, the direction of the applied external force is compared to the direction from the TCP to each adimissible path. The path with the most similarity gets the largest belief increase. The belief system performs updates based on the external force when the input exceeds the threshold of $f_{\text{e,t}}^{\text{est}} = 5\,\text{N}$. External forces below that threshold are assumed to originate from noise.

Starting with the first case, it is assumed that no external force is present. Hence, the minimization problem according to (4.30) does not apply and cannot be solved, since $\mathbf{e}_{\text{f}}$ is undefined. The best update that can be taken is based on the distances between the TCP and the closest points on the paths. To this end, the optimal path position for each path is calculated by solving the unconstrained minimization problem

$$\theta_i^* = \arg\min_{\theta_i \in \Theta_i} \|\mathbf{y}_{\text{t}} - \boldsymbol{\sigma}_{\text{t},i}(\theta_i)\|_2^2, \quad i \in \mathcal{I} \,. \tag{4.32}$$

Based on the solutions θ_i^*, the distances to the TCP are calculated as

$$d_i = \|\boldsymbol{\sigma}_{\text{t},i}(\theta_i^*) - \mathbf{y}_{\text{t}}\|_2, \quad i \in \mathcal{I} \,. \tag{4.33}$$

These distances are the basis for the belief update law. A small distance means the TCP is close to the given path. Its belief should therefore increase. On the other hand, a large distance should cause the corresponding belief to decrease. The belief update should thus be directly proportional to the negative distance. Based on these considerations, the belief update law is introduced as

$$\dot{\mathbf{b}} = \begin{bmatrix} \dot{b}_1 \\ \vdots \\ \dot{b}_n \end{bmatrix} = \Omega\left(\begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}, -\varepsilon_{\text{d}} \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} \right), \quad \varepsilon_{\text{d}} > 0 \,, \tag{4.34}$$

with the adaptation gain for the distance-based belief update $\varepsilon_\mathrm{d} \in \mathbb{R}$. The function $\Omega : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ implements a winner-takes-all (WTA) algorithm, which takes the current beliefs and the negative distance as arguments. It is needed to keep the belief system and its beliefs consistent and in the range between 0 and 1. Since there can only be one active path i_act, only one belief should be increasing, while all the others are decreasing. As a result, only one belief update $\dot{b}_i$ in (4.34) should be positive, while all the others are negative. For a sample implementation of the WTA algorithm $\Omega(\cdot,\cdot)$, see Appendix A.3.1. The final beliefs $\mathbf{b}$ are calculated by integrating the belief updates (4.34) as

$$\mathbf{b}(t) = \int_{t_0}^{t} \dot{\mathbf{b}}(\tau)\mathrm{d}\tau \ . \tag{4.35}$$

In the second case, the belief system uses the external force as main input. It compares the direction of the applied external force $\mathbf{f}_\mathrm{e,t}^\mathrm{est}$ with the direction from the current TCP position $\mathbf{y}_\mathrm{t}$ to each admissible path $\boldsymbol{\sigma}_{\mathrm{t},j}\left(\bar{\theta}_j^*\right), j \in \mathcal{J}$. The belief system accumulates these matching directions and aggregates thereby the intentions. The system can therefore update its beliefs in the background while staying fixed on the active path.

To estimate the intentions, the direction of the external force $\mathbf{e}_\mathrm{f}$ is compared to the directions from the TCP position $\mathbf{y}_\mathrm{t}$ to the positions on the paths $\boldsymbol{\sigma}_{\mathrm{t},j}\left(\bar{\theta}_j^*\right)$, $j \in \mathcal{J}$. A good match in the direction should cause a large belief increase and vice versa. However, there may be multiple paths lying in the same direction, but at different distances from the TCP. The comparison of the *normalized* direction vector $\mathbf{e}_{\mathrm{d},i}$ with the external force direction $\mathbf{e}_\mathrm{f}$ does not account for the different distances. To include that information in the comparison, a scaling is applied before comparing the directions. The scaling must ensure that at least the closest admissible path in any direction is reachable by the admissible human force $f_\mathrm{h,t}$ introduced in Section 4.1.1. With this scaling, each admissible path $j \in \mathcal{J}$ gets a switching force $\mathbf{f}_{\mathrm{s},j}$, which is basically the direction vector, scaled such that the closest admissible path has $\|\mathbf{f}_{\mathrm{s},j}\|_2 = f_\mathrm{h,t}$. In that way, no matter how large the distance between the TCP and the paths is, at least one is always reachable by the admissible human force in any possible direction, see Figure 4.9. This switching force $\mathbf{f}_{\mathrm{s},j}$ is then compared to the applied external force $\mathbf{f}_\mathrm{e,t}^\mathrm{est}$. Non-admissible paths $i \notin \mathcal{J}$ are assigned a switching force of $\mathbf{f}_{\mathrm{s},i} = \mathbf{0}$. To find the scaling, the distance from the TCP $\mathbf{y}_\mathrm{t}$ to the closest admissible path $j \in \mathcal{J}$ is needed and calculated by

$$d_\mathrm{min} = \min_{j \in \mathcal{J}} \left\| \boldsymbol{\sigma}_{\mathrm{t},j}\left(\bar{\theta}_j^*\right) - \mathbf{y}_\mathrm{t} \right\|_2, \quad j \in \mathcal{J} \ . \tag{4.36}$$

The switching force $\mathbf{f}_{\mathrm{s},i}, i \in \mathcal{I}$, for every path is then calculated by

$$\mathbf{f}_{\mathrm{s},i} = \begin{cases} f_\mathrm{h,t} \frac{\boldsymbol{\sigma}_{\mathrm{t},i}\left(\bar{\theta}_i^*\right) - \mathbf{y}_\mathrm{t}}{d_\mathrm{min}} & i \in \mathcal{J} \\ \mathbf{0} & i \notin \mathcal{J} \end{cases} . \tag{4.37}$$

Multiple paths may lie along the same force direction (e. g. Σ_3 and Σ_4 in Figure 4.9). To select the path further away (Σ_4 in Figure 4.9), a larger force than $f_\mathrm{h,t}$ needs to be applied by the operator or the switch must occur in multiple steps.

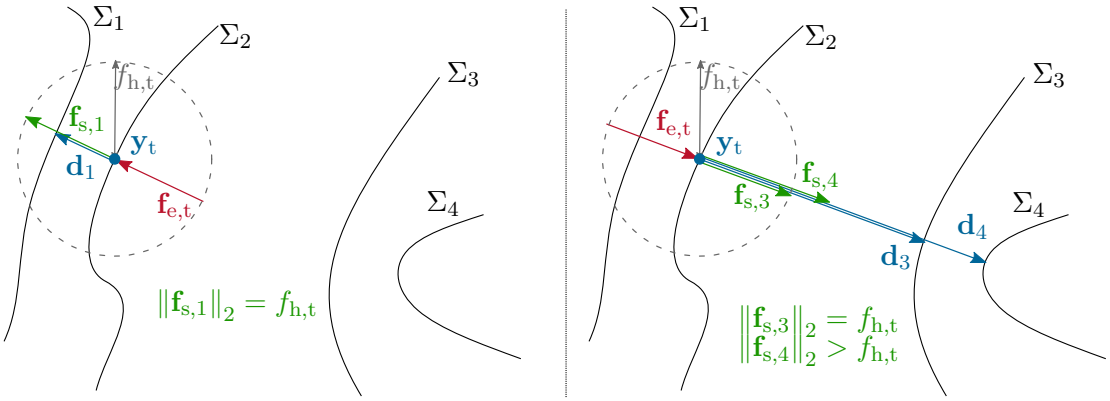


Figure 4.9: The switching force $\mathbf{f}_{\mathrm{s},i}$ is scaled, such that for the switching force of the closest admissible path $\|\mathbf{f}_{\mathrm{s},i}\|_2 = f_{\mathrm{h,t}}$ applies. On the left, $\mathcal{J} = \{1\}$ and thus $\|\mathbf{f}_{\mathrm{s},1}\|_2 = f_{\mathrm{h,t}}$. On the right, $\mathcal{J} = \{3, 4\}$ instead, because of the different direction of the external force $\mathbf{f}_{\mathrm{e,t}}$. The path Σ_3 is closer to $\mathbf{y}_\mathrm{t}$ than Σ_4. Thus, $\|\mathbf{f}_{\mathrm{s},3}\|_2 = f_{\mathrm{h,t}}$ and $\|\mathbf{f}_{\mathrm{s},4}\|_2 > f_{\mathrm{h,t}}$.

Based on the adaptation law in [20], the belief updates are calculated using the actually applied external force $\mathbf{f}_{\mathrm{e,t}}^{\mathrm{est}}$ as

$$\dot{\mathbf{b}} = \Omega\left(\mathbf{b}, -\varepsilon_\mathrm{f}\begin{bmatrix} f_{\mathrm{e},1} \\ \vdots \\ f_{\mathrm{e},n} \end{bmatrix}\right), \varepsilon_\mathrm{f} > 0 \text{ with } f_{\mathrm{e},i} = \left\|\mathbf{f}_{\mathrm{e,t}}^{\mathrm{est}} - \mathbf{f}_{\mathrm{s},i}\right\|_2 \tag{4.38}$$

and the adaptation gain $\varepsilon_\mathrm{f} \in \mathbb{R}$.

The presented belief update (4.38) can only be applied, if a suitable switching path is found, i. e. $\mathcal{J} \neq \{\}$. If the external force $\mathbf{f}_{\mathrm{e,t}}^{\mathrm{est}}$ points in a direction where no path is located, (4.30) cannot be solved for any $i \in \mathcal{I}$. In this case, the distance-based update law (4.34) is used again.

4.2.2 Transition Trajectory Generation

In order to switch from the current path and position (indexed by i) to the new path (indexed by i') without any sudden torque jumps, a transition trajectory is computed online. This transition trajectory is denoted by $\boldsymbol{\kappa}(t)$. During the transition, the robot is controlled by a trajectory tracking controller (TTC) formulated w.r.t the base frame. The translational part $\boldsymbol{\kappa}_\mathrm{t}(t)$ and rotational part $\boldsymbol{\kappa}_\mathrm{r}(t)$ of the trajectory $\boldsymbol{\kappa}(t)$ are designed separately, i. e.

$$\boldsymbol{\kappa}(t) = \begin{bmatrix} \boldsymbol{\kappa}_\mathrm{t}(t) \\ \boldsymbol{\kappa}_\mathrm{r}(t) \end{bmatrix}. \tag{4.39}$$

Figure 4.10 provides a visualization of such a transition from Σ_i to $\Sigma_{i'}$. The movement of the TCP $\mathbf{y}_\mathrm{t}(t)$ is depicted in blue, the paths Σ_i and $\Sigma_{i'}$ are shown in black and the transition trajectory $\boldsymbol{\kappa}_\mathrm{t}(t)$ is depicted in green. At first, the TCP is moving along Σ_i. The external force $\mathbf{f}_\mathrm{e}$ is depicted in red. As soon as the external force $\mathbf{f}_\mathrm{e}$ is applied, the

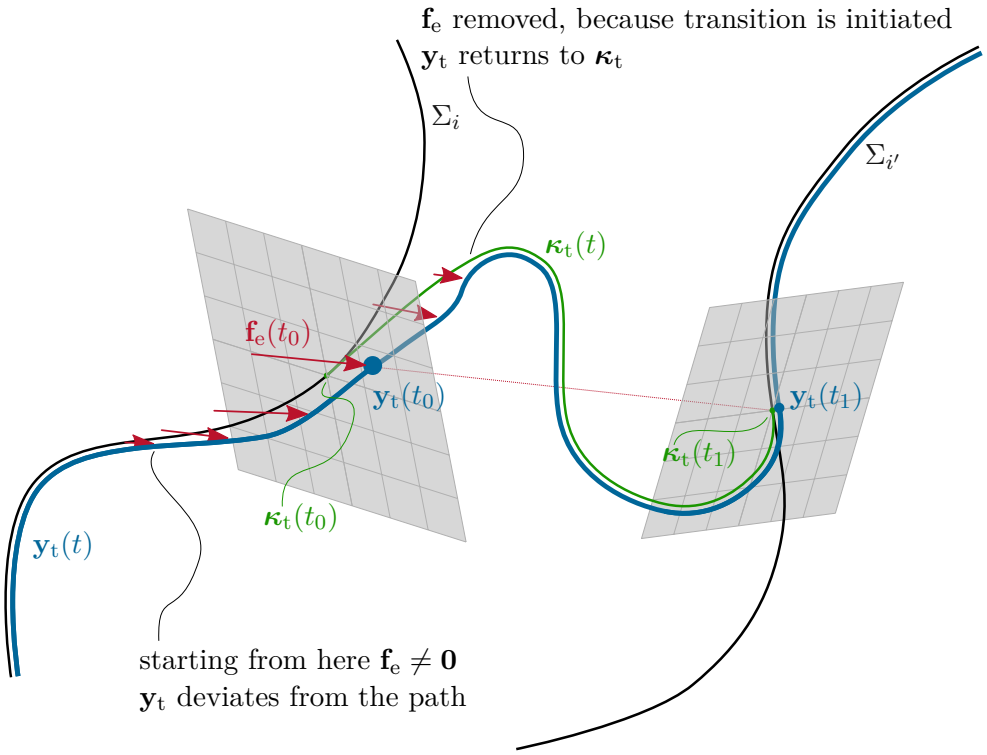


Figure 4.10: Trajectory of the TCP position $\mathbf{y}_t(t)$ during a transition from Σ_i to $\Sigma_{i'}$ with the online-generated transition trajectory $\boldsymbol{\kappa}_t(t)$ and the external force $\mathbf{f}_e(t)$.

TCP slightly deviates from the path Σ_i, since the system is impedance controlled by the PFC. The belief system is updated in the background and at $t = t_0$ the belief system decides that a transition to $\Sigma_{i'}$ is initiated. The online-computed transition trajectory $\boldsymbol{\kappa}(t)$ must be designed in a way that the TTC brings the TCP from Σ_i to $\Sigma_{i'}$, that there are no torque jumps at $t = t_0$ and $t = t_1$ and the motion should be close to a human-like motion. These requirements define the shape of the trajectory $\boldsymbol{\kappa}(t)$ and specify boundary conditions at $t = t_0$ and $t = t_1$. The boundary conditions are especially important at $t = t_0$, where a deviation from the path Σ_i is unavoidable due to the external force $\mathbf{f}_e$ that initiated the transition in the first place, see Figure 4.10.

Shape of the Trajectory

Studies have shown that human motions are performed in a jerk-minimizing way [34]. Thus, to make the robot movement as similar as possible to a human motion, the transition trajectory should be jerk-minimizing. It can be shown that a polynom of order 5 provides such characteristics [34].

The translational part of the transition trajectory $\boldsymbol{\kappa}_t(t)$ is chosen as a $5^{\text{th}}-$ order

polynomial, reading as

$$\boldsymbol{\kappa}_{\mathrm{t}}(t) = \sum_{k=0}^{5} \mathbf{a}_{\mathrm{t},k} \left(\frac{t - t_0}{T_\kappa} \right)^k , \; t \in [t_0, t_0 + T_\kappa] \tag{4.40}$$

with the transition duration T_κ, the start time of transition t_0 and the coefficients $\mathbf{a}_{\mathrm{t},k} \in \mathbb{R}^3$, $k = 0, 1, \ldots, 5$. For the rotational part, a simple polynomial is not feasible. It would not preserve the needed structure to represent an orientation, i. e. direcly interpolating quaternions would not give a *unit* quaternion and directly interpolating matrices would not give a valid *rotation* matrix in between the interpolation points. Hence, a spherical linear interpolation (SLERP) is used instead [35]. This interpolation can be constructed in unit quaternions as well as in rotation matrices. In this work, the interpolation is formulated as rotation matrix, since it simplifies the satisfaction of the boundary conditions in the next section. Thus, the rotational part of the trajectory (with respect to $\mathcal{B}$) is given by

$$\mathbf{R}_\kappa(t) = \exp\Big([\mathbf{a}_{\mathrm{r}}(t)]_\times \Big) \mathbf{R}_{\kappa_0} \tag{4.41a}$$

$$\mathbf{a}_{\mathrm{r}}(t) = \sum_{k=0}^{5} \mathbf{a}_{\mathrm{r},k} \left(\frac{t - t_0}{T_\kappa} \right)^k , \quad t \in [t_0, t_0 + T_\kappa] \; , \tag{4.41b}$$

with the coefficients $\mathbf{a}_{\mathrm{r},k} \in \mathbb{R}^3$, $k = 0, 1, \ldots, 5$ and the initial rotation matrix $\mathbf{R}_{\kappa_0}$. The matrix exponential $\exp(\mathbf{A})$ of a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ used in (4.41) is given by the power series

$$\exp(\mathbf{A}) = \sum_{m=0}^{\infty} \frac{1}{m!} \mathbf{A}^m \; , \tag{4.42}$$

with $\mathbf{A}^0 = \mathbf{I}$.

Boundary Conditions

As described in Section 4.2.1, an external force is used to initiate a Path Switch. This force is of course still applied at the beginning of the transition. Because the PFC is based on an impedance control law, a certain deviation from the desired path is inherently present, see Figure 4.10. This deviation results in a virtual spring force that is counteracting the external force and attracts the TCP towards the path. To obtain a smooth transition, the virtual spring force must be kept consistent when switching from the PFC control to the trajectory control. This requirement yields boundary conditions for the generated trajectory. An impedance controller formulated in inertial coordinates (of which the impedance parameters may differ from the PFC) is used as TTC. This controller is denoted by $\boldsymbol{\tau}_\kappa$. The positive definite TTC impedance matrices are denoted by $(\mathbf{M}_\kappa^{\mathrm{d}}, \mathbf{D}_\kappa^{\mathrm{d}}, \mathbf{K}_\kappa^{\mathrm{d}})$. Based on the PFC law (see (3.39))

$$\boldsymbol{\tau} = \mathbf{n} + \boldsymbol{\tau}_{\mathrm{n}} - \boldsymbol{\tau}_{\mathrm{e}} + \mathbf{M}\hat{\mathbf{J}}^\dagger \left(\begin{bmatrix} \ddot{\boldsymbol{\xi}}^{\mathrm{d}} \\ \dot{\boldsymbol{\omega}}^{\mathcal{D}} \end{bmatrix} + \left(\mathbf{M}^{\mathrm{d}} \right)^{-1} \left[\hat{\mathbf{f}}_{\mathrm{e}} - \underbrace{\mathbf{D}^{\mathrm{d}} \begin{bmatrix} \dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}^{\mathrm{d}} \\ \boldsymbol{\omega}^{\mathcal{T}} - \boldsymbol{\omega}^{\mathcal{D}} \end{bmatrix}}_{\text{damper}} - \underbrace{\mathbf{K}^{d} \begin{bmatrix} \boldsymbol{\xi} - \boldsymbol{\xi}^{\mathrm{d}} \\ \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}} \end{bmatrix}}_{\text{spring}} \right] - \dot{\hat{\mathbf{J}}}\dot{\mathbf{q}} \right) , \tag{4.43}$$

the TTC for the transition trajectory is given by

$$\boldsymbol{\tau}_\kappa = \mathbf{n}+\boldsymbol{\tau}_\mathrm{n}-\boldsymbol{\tau}_\mathrm{e}+\mathbf{M}\mathbf{J}^\dagger\left(\begin{bmatrix}\ddot{\boldsymbol{\kappa}}_\mathrm{t}\\ \dot{\boldsymbol{\omega}}^{\mathcal{K}}\end{bmatrix}+\left(\mathbf{M}_\kappa^\mathrm{d}\right)^{-1}\left[\mathbf{f}_\mathrm{e}-\underbrace{\mathbf{D}_\kappa^\mathrm{d}\begin{bmatrix}\dot{\mathbf{y}}_\mathrm{t}-\dot{\boldsymbol{\kappa}}_\mathrm{t}\\ \boldsymbol{\omega}^{\mathcal{T}}-\boldsymbol{\omega}^{\mathcal{K}}\end{bmatrix}}_{\text{damper}}-\underbrace{\mathbf{K}_\kappa^\mathrm{d}\begin{bmatrix}\mathbf{y}_\mathrm{t}-\boldsymbol{\kappa}_\mathrm{t}\\ \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{K}}\end{bmatrix}}_{\text{spring}}\right]-\dot{\mathbf{J}}\dot{\mathbf{q}}\right). \tag{4.44}$$

Note that the desired coordinates $(\boldsymbol{\xi}^\mathrm{d}, \dot{\boldsymbol{\xi}}^\mathrm{d}, \ddot{\boldsymbol{\xi}}^\mathrm{d})$ in (4.43) are replaced by the transition trajectory $(\boldsymbol{\kappa}_\mathrm{t}, \dot{\boldsymbol{\kappa}}_\mathrm{t}, \ddot{\boldsymbol{\kappa}}_\mathrm{t})$ in (4.44) during the transition, the path-based coordinates $\boldsymbol{\xi}$ and $\dot{\boldsymbol{\xi}}$ are replaced by the system output $\mathbf{y}_\mathrm{t}$ and its derivative $\dot{\mathbf{y}}_\mathrm{t}$, respectively, and the Jacobian $\mathbf{J}$ described in the base frame is used instead of the path-based Jacobian $\hat{\mathbf{J}}$. The vectors $\boldsymbol{\omega}^{\mathcal{K}}$ and $\dot{\boldsymbol{\omega}}^{\mathcal{K}}$ denote the desired angular velocity and desired angular acceleration of the trajectory, respectively, and $\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{K}}$ is the vector part of the quaternion error between the TCP frame $\mathcal{T}$ and the trajectory frame $\mathcal{K}$. Note that the desired impedance parameters change from $(\mathbf{M}^\mathrm{d}, \mathbf{D}^\mathrm{d}, \mathbf{K}^\mathrm{d})$ to $(\mathbf{M}_\kappa^\mathrm{d}, \mathbf{D}_\kappa^\mathrm{d}, \mathbf{K}_\kappa^\mathrm{d})$. The original impedance parameters $(\mathbf{M}^\mathrm{d}, \mathbf{D}^\mathrm{d}, \mathbf{K}^\mathrm{d})$ may not be suitable for the TTC, since the transition trajectory has to be executed consistently. While $\mathbf{K}^\mathrm{d} = \mathrm{diag}(k_\parallel^\mathrm{d} = 0, k_\perp^\mathrm{d}, k_\pitchfork^\mathrm{d}, \mathbf{k}_\mathrm{r})$ is a valid option for the PFC, it cannot be used for the transition, as it would not guarantee that the robot follows the tangential motion of the transition trajectory.

For a smooth transition, the torques $\boldsymbol{\tau}$ and $\boldsymbol{\tau}_\kappa$ must match at the beginning of the transition $t = t_0$ and at the end of the transition $t = t_1$, i. e. $\boldsymbol{\tau}(t_0) - \boldsymbol{\tau}_\kappa(t_0) = \mathbf{0}$ and $\boldsymbol{\tau}(t_1) - \boldsymbol{\tau}_\kappa(t_1) = \mathbf{0}$. Using (4.43) and (4.44), rearranging these matching conditions in a way that the according derivatives appear next to each other, yields

$$\begin{aligned}
&-\hat{\mathbf{J}}^\dagger\left(\left(\mathbf{M}^\mathrm{d}\right)^{-1}\underbrace{\mathbf{K}^\mathrm{d}\begin{bmatrix}\boldsymbol{\xi}-\boldsymbol{\xi}^\mathrm{d}\\ \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}}\end{bmatrix}}_{\text{virt. spring PFC}}\right)+\mathbf{J}^\dagger\left(\left(\mathbf{M}_\kappa^\mathrm{d}\right)^{-1}\underbrace{\mathbf{K}_\kappa^\mathrm{d}\begin{bmatrix}\mathbf{y}_\mathrm{t}-\boldsymbol{\kappa}_t\\ \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{K}}\end{bmatrix}}_{\text{virt. spring TTC}}\right)+\\
&-\hat{\mathbf{J}}^\dagger\left(\left(\mathbf{M}^\mathrm{d}\right)^{-1}\underbrace{\mathbf{D}^\mathrm{d}\begin{bmatrix}\dot{\boldsymbol{\xi}}-\dot{\boldsymbol{\xi}}^\mathrm{d}\\ \boldsymbol{\omega}^{\mathcal{T}}-\boldsymbol{\omega}^{\mathcal{D}}\end{bmatrix}}_{\text{virt. damper PFC}}\right)+\mathbf{J}^\dagger\left(\left(\mathbf{M}_\kappa^\mathrm{d}\right)^{-1}\underbrace{\mathbf{D}_\kappa^\mathrm{d}\begin{bmatrix}\dot{\mathbf{y}}_\mathrm{t}-\dot{\boldsymbol{\kappa}}_\mathrm{t}\\ \boldsymbol{\omega}^{\mathcal{T}}-\boldsymbol{\omega}^{\mathcal{K}}\end{bmatrix}}_{\text{virt. damper TTC}}\right)+\\
&\hat{\mathbf{J}}^\dagger\left(\begin{bmatrix}\ddot{\boldsymbol{\xi}}^\mathrm{d}\\ \dot{\boldsymbol{\omega}}^{\mathcal{D}}\end{bmatrix}+\left(\mathbf{M}^\mathrm{d}\right)^{-1}\hat{\mathbf{f}}_\mathrm{e}-\dot{\hat{\mathbf{J}}}\dot{\mathbf{q}}\right)-\mathbf{J}^\dagger\left(\begin{bmatrix}\ddot{\boldsymbol{\kappa}}_\mathrm{t}\\ \dot{\boldsymbol{\omega}}^{\mathcal{K}}\end{bmatrix}+\left(\mathbf{M}_\kappa^\mathrm{d}\right)^{-1}\mathbf{f}_\mathrm{e}-\dot{\mathbf{J}}\dot{\mathbf{q}}\right)=\mathbf{0},\quad t\in\{t_0,t_1\}\,.
\end{aligned} \tag{4.45}$$

The closed-loop system for the PFC takes $\boldsymbol{\xi}^\mathrm{d}, \dot{\boldsymbol{\xi}}^\mathrm{d}, \ddot{\boldsymbol{\xi}}^\mathrm{d}, \mathbf{R}^{\mathcal{D}}, \boldsymbol{\omega}^{\mathcal{D}}$ and $\dot{\boldsymbol{\omega}}^{\mathcal{D}}$ as input and provides a torque $\boldsymbol{\tau}$ according to (4.43) as control input to the plant. In the same way, the closed-loop system for the TTC takes $\boldsymbol{\kappa}_\mathrm{t}, \dot{\boldsymbol{\kappa}}_\mathrm{t}, \ddot{\boldsymbol{\kappa}}_\mathrm{t}, \mathbf{R}_\kappa, \boldsymbol{\omega}^{\mathcal{K}}$ and $\dot{\boldsymbol{\omega}}^{\mathcal{K}}$ as input and provides a torque $\boldsymbol{\tau}_\kappa$ according to (4.44) as control input to the plant. To match the influence of each input, the torque $\boldsymbol{\tau}$ originating from $\boldsymbol{\xi}^\mathrm{d}$ and the torque $\boldsymbol{\tau}_\kappa$ originating from $\boldsymbol{\kappa}_\mathrm{t}$ must be the same at the boundaries between the PFC and the TTC. The same holds true for the other inputs of the closed-loop system for the PFC and the closed-loop system for

the TTC, i. e. $\dot{\boldsymbol{\xi}}^{\mathrm{d}} \leftrightarrow \dot{\boldsymbol{\kappa}}_{\mathrm{t}}$, $\ddot{\boldsymbol{\xi}}^{\mathrm{d}} \leftrightarrow \ddot{\boldsymbol{\kappa}}_{\mathrm{t}}$, $\mathbf{R}^{\mathcal{D}} \leftrightarrow \mathbf{R}_{\kappa}$, $\boldsymbol{\omega}^{\mathcal{D}} \leftrightarrow \boldsymbol{\omega}^{\mathcal{K}}$ and $\dot{\boldsymbol{\omega}}^{\mathcal{D}} \leftrightarrow \dot{\boldsymbol{\omega}}^{\mathcal{K}}$. In another way of interpreting these matching conditions, the torque $\boldsymbol{\tau}$ originating from the virtual spring, damper or inertial force of the PFC must be the same as the torque $\boldsymbol{\tau}_{\kappa}$ originating from the virtual spring, damper or inertial force of the TTC, respectively, at $t = t_0$ and $t = t_1$. Since (4.43) and (4.44) are linear equations in the inputs $\boldsymbol{\xi}^{\mathrm{d}}, \dot{\boldsymbol{\xi}}^{\mathrm{d}}, \ddot{\boldsymbol{\xi}}^{\mathrm{d}}, \mathbf{R}^{\mathcal{D}}, \boldsymbol{\omega}^{\mathcal{D}}, \dot{\boldsymbol{\omega}}^{\mathcal{D}}$ and $\boldsymbol{\kappa}_{\mathrm{t}}, \dot{\boldsymbol{\kappa}}_{\mathrm{t}}, \ddot{\boldsymbol{\kappa}}_{\mathrm{t}}, \mathbf{R}_{\kappa}, \boldsymbol{\omega}^{\mathcal{K}}, \dot{\boldsymbol{\omega}}^{\mathcal{K}}$, this approach is valid. With these considerations, each line in (4.45) is individually set to $\mathbf{0}$, which certainly is a solution to (4.45). The first line of (4.45) represents the virtual spring forces. Setting this line to $\mathbf{0}$ yields

$$-\hat{\mathbf{J}}^{\dagger}\left(\left(\mathbf{M}^{\mathrm{d}}\right)^{-1}\underbrace{\mathbf{K}^{\mathrm{d}}\begin{bmatrix}\boldsymbol{\xi}-\boldsymbol{\xi}^{\mathrm{d}}\\ \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}}\end{bmatrix}}_{\text{virt. spring PFC}}\right)+\mathbf{J}^{\dagger}\left(\left(\mathbf{M}_{\kappa}^{\mathrm{d}}\right)^{-1}\underbrace{\mathbf{K}_{\kappa}^{\mathrm{d}}\begin{bmatrix}\mathbf{y}_{\mathrm{t}}-\boldsymbol{\kappa}_{\mathrm{t}}\\ \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{K}}\end{bmatrix}}_{\text{virt. spring TTC}}\right)=\mathbf{0},\quad t\in\{t_0,t_1\}\ . \tag{4.46}$$

Equation (4.46) is solved for the interesting boundary values of the transition trajectory $\boldsymbol{\kappa}_{\mathrm{t}}$ and $\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{K}}$, resulting in a mapping from $(\boldsymbol{\xi}^{\mathrm{d}}, \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}})$ to $(\boldsymbol{\kappa}_{\mathrm{t}}, \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{K}})$ of

$$\begin{bmatrix}\boldsymbol{\kappa}_{\mathrm{t}}\\ -\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{K}}\end{bmatrix}=\begin{bmatrix}\mathbf{y}_{\mathrm{t}}\\ \mathbf{0}\end{bmatrix}-\left(\mathbf{K}_{\kappa}^{\mathrm{d}}\right)^{-1}\mathbf{M}_{\kappa}^{\mathrm{d}}\mathbf{J}\hat{\mathbf{J}}^{\dagger}\left(\left(\mathbf{M}^{\mathrm{d}}\right)^{-1}\mathbf{K}^{\mathrm{d}}\begin{bmatrix}\boldsymbol{\xi}-\boldsymbol{\xi}^{\mathrm{d}}\\ \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}}\end{bmatrix}\right),\quad t\in\{t_0,t_1\}\ . \tag{4.47}$$

Additionally, with the assumption of diagonal impedance parameter matrices $\mathbf{M}^{\mathrm{d}}$, $\mathbf{D}^{\mathrm{d}}$, $\mathbf{K}^{\mathrm{d}}$, $\mathbf{M}_{\kappa}^{\mathrm{d}}$, $\mathbf{D}_{\kappa}^{\mathrm{d}}$, $\mathbf{K}_{\kappa}^{\mathrm{d}} \in \mathbb{R}^{6\times 6}$ with positive entries, combined with the identity for the pseudoinverse of the path-based Jacobian $\hat{\mathbf{J}}^{\dagger}$ (4.15), the translational and rotational part of (4.47) are decoupled and provide one matching condition for the translational part $\boldsymbol{\xi}^{\mathrm{d}} \leftrightarrow \boldsymbol{\kappa}_{\mathrm{t}}$ and one matching condition for the rotational part $\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}} \leftrightarrow \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{K}}$. Hence, with (4.47) the mapping for the inputs $\boldsymbol{\xi}^{\mathrm{d}} \leftrightarrow \boldsymbol{\kappa}_{\mathrm{t}}$ and $\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}} \leftrightarrow \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{K}}$ of the closed-loop system for the PFC and of the closed-loop system for the TTC is found.

Repeating the above procedure for the second and third line of (4.45) yields the boundary conditions

$$\begin{bmatrix}\dot{\boldsymbol{\kappa}}_{\mathrm{t}}\\ \boldsymbol{\omega}^{\mathcal{K}}\end{bmatrix}=\begin{bmatrix}\dot{\mathbf{y}}_{\mathrm{t}}\\ \boldsymbol{\omega}^{\mathcal{T}}\end{bmatrix}-\left(\mathbf{D}_{\kappa}^{\mathrm{d}}\right)^{-1}\mathbf{M}_{\kappa}^{\mathrm{d}}\mathbf{J}\hat{\mathbf{J}}^{\dagger}\left(\mathbf{M}^{\mathrm{d}}\right)^{-1}\mathbf{D}^{\mathrm{d}}\begin{bmatrix}\dot{\boldsymbol{\xi}}-\dot{\boldsymbol{\xi}}^{\mathrm{d}}\\ \boldsymbol{\omega}^{\mathcal{T}}-\boldsymbol{\omega}^{\mathcal{D}}\end{bmatrix},\qquad t\in\{t_0,t_1\}\ , \tag{4.48}$$

$$\begin{aligned}\begin{bmatrix}\ddot{\boldsymbol{\kappa}}_{\mathrm{t}}\\ \dot{\boldsymbol{\omega}}^{\mathcal{K}}\end{bmatrix}&=\mathbf{J}\hat{\mathbf{J}}^{\dagger}\begin{bmatrix}\ddot{\boldsymbol{\xi}}^{\mathrm{d}}\\ \dot{\boldsymbol{\omega}}^{\mathcal{D}}\end{bmatrix}+\left(\dot{\mathbf{J}}-\mathbf{J}\hat{\mathbf{J}}^{\dagger}\dot{\hat{\mathbf{J}}}\right)\dot{\mathbf{q}}+\mathbf{J}\hat{\mathbf{J}}^{\dagger}\left(\mathbf{M}^{\mathrm{d}}\right)^{-1}\hat{\mathbf{f}}_{\mathrm{e}}-\left(\mathbf{M}_{\kappa}^{\mathrm{d}}\right)^{-1}\mathbf{f}_{\mathrm{e}}\\ &\overset{\text{cf. (4.4)}}{=}\begin{bmatrix}\ddot{\mathbf{y}}_{\mathrm{t}}\\ \dot{\boldsymbol{\omega}}^{\mathcal{T}}\end{bmatrix}-\mathbf{J}\hat{\mathbf{J}}^{\dagger}\begin{bmatrix}\ddot{\boldsymbol{\xi}}-\ddot{\boldsymbol{\xi}}^{\mathrm{d}}\\ \dot{\boldsymbol{\omega}}^{\mathcal{T}}-\dot{\boldsymbol{\omega}}^{\mathcal{D}}\end{bmatrix}+\mathbf{J}\hat{\mathbf{J}}^{\dagger}\left(\mathbf{M}^{\mathrm{d}}\right)^{-1}\hat{\mathbf{f}}_{\mathrm{e}}-\left(\mathbf{M}_{\kappa}^{\mathrm{d}}\right)^{-1}\mathbf{f}_{\mathrm{e}},\qquad t\in\{t_0,t_1\}\ .\end{aligned} \tag{4.49}$$

The equations (4.47), (4.48) and (4.49) provide in total 36 boundary conditions, which is enough to specify the 36 components of the coefficients $\mathbf{a}_{\mathrm{t},k}$ in (4.40) and $\mathbf{a}_{\mathrm{r},k}$ in (4.41).

For $t = t_0$, the equations (4.47), (4.48) and (4.49) are directly evaluated, since the state of the system is known. However, to evaluate (4.47), (4.48) and (4.49) at $t = t_1$, the future state of the system would be needed, since the trajectory generation is done at $t = t_0$ and

$t_1 > t_0$, which is not known beforehand. To this end, it can be assumed that the external force $\mathbf{f}_\mathrm{e}$, which initiated the transition, has vanished until the end of transition and there is no external force $\mathbf{f}_\mathrm{e}$ applied at $t = t_1$. This is a reasonable assumption, since an operator will stop to push the TCP, once it starts to move in the desired direction, cf. Figure 4.10. Without any external force $\mathbf{f}_\mathrm{e}$, the actual path-based coordinates $\boldsymbol{\xi}_{i'}(t_1)$ and the actual orientation $\mathbf{R}^{\mathcal{T}}(t_1)$ are expected to match the desired path-based coordinates $\boldsymbol{\xi}^\mathrm{d}_{i'}(t_1)$ and the desired orientation $\mathbf{R}^{\mathcal{D}_{i'}}(t_1)$, respectively. The same assumption holds true for the derivatives of the path-based coordinates $\dot{\boldsymbol{\xi}}_{i'}(t_1)$, $\dot{\boldsymbol{\xi}}^\mathrm{d}_{i'}(t_1)$ and $\ddot{\boldsymbol{\xi}}_{i'}(t_1)$, $\ddot{\boldsymbol{\xi}}^\mathrm{d}_{i'}(t_1)$ as well as for the angular velocity $\boldsymbol{\omega}^{\mathcal{T}}(t_1)$ and $\boldsymbol{\omega}^{\mathcal{D}_{i'}}(t_1)$ and angular acceleration $\dot{\boldsymbol{\omega}}^{\mathcal{T}}(t_1)$ and $\dot{\boldsymbol{\omega}}^{\mathcal{D}_{i'}}(t_1)$. Thus, with

$$\boldsymbol{\xi}_{i'}(t_1) - \boldsymbol{\xi}^\mathrm{d}_{i'}(t_1) = \mathbf{0} \;, \tag{4.50a}$$

$$\boldsymbol{\varepsilon}^{\mathcal{D}_{i'}}_{\mathcal{T}}(t_1) = \mathbf{0} \;, \tag{4.50b}$$

$$\dot{\boldsymbol{\xi}}_{i'}(t_1) - \dot{\boldsymbol{\xi}}^\mathrm{d}_{i'}(t_1) = \mathbf{0} \;, \tag{4.50c}$$

$$\boldsymbol{\omega}^{\mathcal{T}}(t_1) - \boldsymbol{\omega}^{\mathcal{D}_{i'}}(t_1) = \mathbf{0} \;, \tag{4.50d}$$

$$\ddot{\boldsymbol{\xi}}_{i'}(t_1) - \ddot{\boldsymbol{\xi}}^\mathrm{d}_{i'}(t_1) = \mathbf{0} \;, \tag{4.50e}$$

$$\dot{\boldsymbol{\omega}}^{\mathcal{T}}(t_1) - \dot{\boldsymbol{\omega}}^{\mathcal{D}_{i'}}(t_1) = \mathbf{0} \;, \tag{4.50f}$$

the evaluation of (4.47), (4.48) and (4.49) at $t = t_1$ simplifies to

$$\begin{bmatrix} \boldsymbol{\kappa}_\mathrm{t}(t_1) \\ \mathbf{R}_\kappa(t_1) \end{bmatrix} = \begin{bmatrix} \mathbf{y}_\mathrm{t}(t_1) \\ \mathbf{R}^{\mathcal{T}}(t_1) \end{bmatrix} \tag{4.51}$$

$$\begin{bmatrix} \dot{\boldsymbol{\kappa}}_\mathrm{t}(t_1) \\ \boldsymbol{\omega}^{\mathcal{K}}(t_1) \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{y}}_\mathrm{t}(t_1) \\ \boldsymbol{\omega}^{\mathcal{T}}(t_1) \end{bmatrix} \tag{4.52}$$

$$\begin{bmatrix} \ddot{\boldsymbol{\kappa}}_\mathrm{t}(t_1) \\ \dot{\boldsymbol{\omega}}^{\mathcal{K}}(t_1) \end{bmatrix} = \begin{bmatrix} \ddot{\mathbf{y}}_\mathrm{t}(t_1) \\ \dot{\boldsymbol{\omega}}^{\mathcal{T}}(t_1) \end{bmatrix} \;. \tag{4.53}$$

That means, the position of the TCP $\mathbf{y}_\mathrm{t}$, the velocity of the TCP $\dot{\mathbf{y}}_\mathrm{t}$ and the acceleration of the TCP $\ddot{\mathbf{y}}_\mathrm{t}$, as well as the orientation of the TCP $\mathbf{R}^{\mathcal{T}}$, the angular velocity of the TCP $\boldsymbol{\omega}^{\mathcal{T}}$ and the angular acceleration of the TCP $\dot{\boldsymbol{\omega}}^{\mathcal{T}}$, at the end of transition $t = t_1$ directly correspond to the corresponding quantities of the transition trajectory.

For the position $\mathbf{y}_\mathrm{t}(t_1)$, the parallel transport frame $\mathcal{P}_{i'}$ is positioned at the point on the path that was selected in Section 4.2.1, i. e. in the most general case it is

$$\boldsymbol{\kappa}_\mathrm{t}(t_1) = \mathbf{y}_\mathrm{t}(t_1) = \boldsymbol{\sigma}_{\mathrm{t},i'}\left(\bar{\theta}^*_{i'}\right) + \mathbf{e}_{\perp,i'}\left(\bar{\theta}^*_{i'}\right)\xi^\mathrm{d}_{\perp,i'}(t_1) + \mathbf{e}_{\pitchfork,i'}\left(\bar{\theta}^*_{i'}\right)\xi^\mathrm{d}_{\pitchfork,i'}(t_1) \;. \tag{4.54}$$

Note that the calculation of $\xi_\parallel$ is based on an integral equation (3.18). Since the calculation of $\xi_{\parallel,i'}$ only provides a useful value, if $\Sigma_{i'}$ is the active path (i. e. regarding the transition only if $t \geq t_1$), the integral equation for $\xi_{\parallel,i'}$ starts at $t = t_1$ with an initial value of $\xi_{\parallel,i'}(t_1) = \xi^\mathrm{d}_{\parallel,i'}(t_1)$. The time derivative of the translational part of the transition trajectory at the end of transition $\boldsymbol{\kappa}_\mathrm{t}(t_1)$ is given according to (3.27) as

$$\dot{\boldsymbol{\kappa}}_\mathrm{t}(t_1) = \dot{\mathbf{y}}_\mathrm{t}(t_1) = \begin{bmatrix} \frac{1}{\beta_{i'}}\mathbf{e}_{\parallel,i'}\left(\bar{\theta}^*_{i'}\right) & \mathbf{e}_{\perp,i'}\left(\bar{\theta}^*_{i'}\right) & \mathbf{e}_{\pitchfork,i'}\left(\bar{\theta}^*_{i'}\right) \end{bmatrix} \dot{\boldsymbol{\xi}}^\mathrm{d}_{i'}(t_1) \;, \tag{4.55}$$

see Appendix A.1.2. Time derivative of (4.55) yields

$$\ddot{\boldsymbol{\kappa}}_{\mathrm{t}}(t_1) = \Big[\tfrac{\mathrm{d}}{\mathrm{d}t}\Big(\tfrac{1}{\beta_{i'}}\Big)\mathbf{e}_{\parallel,i'}\Big(\bar{\theta}^*_{i'}\Big) + \tfrac{1}{\beta_{i'}}\dot{\mathbf{e}}_{\parallel,i'}\Big(\bar{\theta}^*_{i'}\Big) \quad \dot{\mathbf{e}}_{\perp,i'}\Big(\bar{\theta}^*_{i'}\Big) \quad \dot{\mathbf{e}}_{\pitchfork,i'}\Big(\bar{\theta}^*_{i'}\Big)\Big]\dot{\boldsymbol{\xi}}^{\mathrm{d}}_{i'}(t_1) + \Big[\tfrac{1}{\beta_{i'}}\mathbf{e}_{\parallel,i'}\Big(\bar{\theta}^*_{i'}\Big) \quad \mathbf{e}_{\perp,i'}\Big(\bar{\theta}^*_{i'}\Big) \quad \mathbf{e}_{\pitchfork,i'}\Big(\bar{\theta}^*_{i'}\Big)\Big]\ddot{\boldsymbol{\xi}}^{\mathrm{d}}_{i'}(t_1)\ . \tag{4.56}$$

Equations (4.54), (4.55) and (4.56) specify the boundary conditions of the translational part of the trajectory at the end of transition $\boldsymbol{\kappa}_{\mathrm{t}}(t_1)$ in the general case, in which the expected parallel transport frame $\mathcal{P}_{i'}$ needs to be constructed for $t = t_1$ beforehand. In a special case, the desired path coordinates in the transversal plane $\xi^{\mathrm{d}}_{\perp}$, $\xi^{\mathrm{d}}_{\pitchfork}$ (and their time derivatives $\dot{\xi}^{\mathrm{d}}_{\perp}$, $\dot{\xi}^{\mathrm{d}}_{\pitchfork}$, $\ddot{\xi}^{\mathrm{d}}_{\perp}$, $\ddot{\xi}^{\mathrm{d}}_{\pitchfork}$) are zero $\xi^{\mathrm{d}}_{\perp} = \xi^{\mathrm{d}}_{\pitchfork} = \dot{\xi}^{\mathrm{d}}_{\perp} = \dot{\xi}^{\mathrm{d}}_{\pitchfork} = \ddot{\xi}^{\mathrm{d}}_{\perp} = \ddot{\xi}^{\mathrm{d}}_{\pitchfork} = 0$, i. e. the only desired motion is *along* the path. In this case, (4.54), (4.55) and (4.56) simplify and the construction of the parallel transport frame $\mathcal{P}_{i'}$ at $t = t_1$ is not needed to be done beforehand. With these assumptions, $\alpha(\mathbf{y}_{\mathrm{t}}(t)) = 0$ follows from (3.13) and $\beta(\mathbf{y}_{\mathrm{t}}(t)) = 1 = \text{const.}$ due to (3.16). Consequently, (4.54), (4.55) and (4.56) simplify to

$$\boldsymbol{\kappa}_{\mathrm{t}}(t_1) = \boldsymbol{\sigma}_{\mathrm{t},i'}\Big(\bar{\theta}^*_{i'}\Big) \tag{4.57}$$

$$\dot{\boldsymbol{\kappa}}_{\mathrm{t}}(t_1) = \mathbf{e}_{\parallel,i'}\Big(\bar{\theta}^*_{i'}\Big)\dot{\xi}^{\mathrm{d}}_{\parallel,i'}(t_1) \tag{4.58}$$

$$\ddot{\boldsymbol{\kappa}}_{\mathrm{t}}(t_1) = \dot{\mathbf{e}}_{\parallel,i'}\Big(\bar{\theta}^*_{i'}\Big)\dot{\xi}^{\mathrm{d}}_{\parallel,i'}(t_1) + \mathbf{e}_{\parallel,i'}\Big(\bar{\theta}^*_{i'}\Big)\ddot{\xi}^{\mathrm{d}}_{\parallel,i'}(t_1)\ . \tag{4.59}$$

The tangential unit vector $\mathbf{e}_{\parallel,i'}(\theta^*)$ is directly given by the defintion of the path (3.5). The time derivative of the tangential unit vector $\dot{\mathbf{e}}_{\parallel,i'}(\theta^*)$ additionally needs the time derivative of the optimal path parameter $\dot{\theta}^*_{i'}$, which is specified by $\dot{\xi}_{\parallel,i}(t_1) = \dot{\xi}^{\mathrm{d}}_{\parallel,i}(t_1)$ and (3.21).

For the rotational part, the orientation of the path $\mathbf{R}^{\mathcal{D}_{i'}}\Big(\bar{\theta}^*_{i'}\Big)$, the angular velocity of the path-based frame $\boldsymbol{\omega}^{\mathcal{D}_{i'}}\Big(\bar{\theta}^*_{i'}, \dot{\theta}^*_{i'}\Big)$ and the angular acceleration of the path-based frame $\dot{\boldsymbol{\omega}}^{\mathcal{D}_{i'}}\Big(\bar{\theta}^*_{i'}, \dot{\theta}^*_{i'}, \ddot{\theta}^*_{i'}\Big)$ are directly evaluated, resulting in the boundary conditions

$$\mathbf{R}_{\kappa}(t_1) = \mathbf{R}^{\mathcal{D}_{i'}}\Big(\bar{\theta}^*_{i'}\Big) \tag{4.60}$$

$$\boldsymbol{\omega}^{\mathcal{K}}(t_1) = \boldsymbol{\omega}^{\mathcal{D}_{i'}}\Big(\bar{\theta}^*_{i'}, \dot{\theta}^*_{i'}\Big) \tag{4.61}$$

$$\dot{\boldsymbol{\omega}}^{\mathcal{K}}(t_1) = \dot{\boldsymbol{\omega}}^{\mathcal{D}_{i'}}\Big(\bar{\theta}^*_{i'}, \dot{\theta}^*_{i'}, \ddot{\theta}^*_{i'}\Big)\ . \tag{4.62}$$

The time derivatives of the optimal path parameter $\dot{\theta}^*_{i'}$, $\ddot{\theta}^*_{i'}$ are specified by the assumptions (4.50), the path definition $\boldsymbol{\sigma}_{\mathrm{t},i'}(\theta_{i'})$ and the desired path-dependent coordinates $\dot{\xi}^{\mathrm{d}}_{\parallel,i}(t_1)$, cf. (3.21).

Parameter Calculation

In this section, explicit values for the coefficients $\mathbf{a}_{\mathrm{t},k}$ and $\mathbf{a}_{\mathrm{r},k}, k = 0, 1, \ldots, 5$, of the polynomials (4.40) and (4.41) are found. Starting with the translational part, the time

derivatives of the transition trajectory $\boldsymbol{\kappa}_\mathrm{t}(t)$ are needed and calculated from (4.40),

$$\boldsymbol{\kappa}_\mathrm{t}(t) = \sum_{k=0}^{5} \mathbf{a}_{\mathrm{t},k} \left(\frac{t-t_0}{T_\kappa}\right)^k \tag{4.63a}$$

$$\dot{\boldsymbol{\kappa}}_\mathrm{t}(t) = \sum_{k=1}^{5} \mathbf{a}_{\mathrm{t},k} \frac{k}{T_\kappa} \left(\frac{t-t_0}{T_\kappa}\right)^{k-1} \tag{4.63b}$$

$$\ddot{\boldsymbol{\kappa}}_\mathrm{t}(t) = \sum_{k=2}^{5} \mathbf{a}_{\mathrm{t},k} \frac{k(k-1)}{T_\kappa^2} \left(\frac{t-t_0}{T_\kappa}\right)^{k-2} . \tag{4.63c}$$

Evaluating (4.63) at $t = t_0$ and $t = t_1 = t_0 + T_\kappa$ yields

$$\boldsymbol{\kappa}_\mathrm{t}(t_0) = \mathbf{a}_{\mathrm{t},0} \tag{4.64a}$$

$$\dot{\boldsymbol{\kappa}}_\mathrm{t}(t_0) = \frac{\mathbf{a}_{\mathrm{t},1}}{T_\kappa} \tag{4.64b}$$

$$\ddot{\boldsymbol{\kappa}}_\mathrm{t}(t_0) = \frac{2\mathbf{a}_{\mathrm{t},2}}{T_\kappa^2} \tag{4.64c}$$

$$\boldsymbol{\kappa}_\mathrm{t}(t_1) = \mathbf{a}_{\mathrm{t},0} + \mathbf{a}_{\mathrm{t},1} + \mathbf{a}_{\mathrm{t},2} + \mathbf{a}_{\mathrm{t},3} + \mathbf{a}_{\mathrm{t},4} + \mathbf{a}_{\mathrm{t},5} \tag{4.64d}$$

$$\dot{\boldsymbol{\kappa}}_\mathrm{t}(t_1) = \frac{\mathbf{a}_{\mathrm{t},1} + 2\mathbf{a}_{\mathrm{t},2} + 3\mathbf{a}_{\mathrm{t},3} + 4\mathbf{a}_{\mathrm{t},4} + 5\mathbf{a}_{\mathrm{t},5}}{T_\kappa} \tag{4.64e}$$

$$\ddot{\boldsymbol{\kappa}}_\mathrm{t}(t_1) = \frac{2\mathbf{a}_{\mathrm{t},2} + 6\mathbf{a}_{\mathrm{t},3} + 12\mathbf{a}_{\mathrm{t},4} + 20\mathbf{a}_{\mathrm{t},5}}{T_\kappa^2} . \tag{4.64f}$$

Together with the boundary conditions (4.47), (4.48) and (4.49), the system of equations (4.64) fully define the coefficients $\mathbf{a}_{\mathrm{t},k}, k = 0, 1, \ldots, 5$, of the translational transition trajectory $\boldsymbol{\kappa}_\mathrm{t}(t)$.

The above procedure is repeated for the rotational trajectory $\mathbf{R}_\kappa(t)$ in (4.41). To satisfy the rotational part of the boundary conditions (4.47), (4.48) and (4.49), the angular velocity $\boldsymbol{\omega}^\mathcal{K}(t)$ and the angular acceleration $\dot{\boldsymbol{\omega}}^\mathcal{K}(t)$ of the rotational path of $\boldsymbol{\kappa}(t)$ are needed. The angular velocity is calculated as

$$\begin{aligned}\left[\boldsymbol{\omega}^\mathcal{K}(t)\right]_\times = \dot{\mathbf{R}}_\kappa(t)(\mathbf{R}_\kappa(t))^\mathrm{T} &= \exp\Big([\mathbf{a}_\mathrm{r}(t)]_\times\Big)[\dot{\mathbf{a}}_\mathrm{r}(t)]_\times \mathbf{R}_{\kappa_0}(\mathbf{R}_{\kappa_0})^\mathrm{T} \exp\Big([\mathbf{a}_\mathrm{r}(t)]_\times\Big)^\mathrm{T} \\ &= \exp\Big([\mathbf{a}_\mathrm{r}(t)]_\times\Big)[\dot{\mathbf{a}}_\mathrm{r}(t)]_\times \exp\Big([\mathbf{a}_\mathrm{r}(t)]_\times\Big)^\mathrm{T} .\end{aligned} \tag{4.65}$$

Differentiating (4.65) w.r.t. the time and using the skew symmetric identity $[\cdot]_\times^\mathrm{T} + [\cdot]_\times = \mathbf{0}$ yields

$$\left[\dot{\boldsymbol{\omega}}^\mathcal{K}(t)\right]_\times = \exp\Big([\mathbf{a}_\mathrm{r}(t)]_\times\Big)[\ddot{\mathbf{a}}_\mathrm{r}(t)]_\times \exp\Big([\mathbf{a}_\mathrm{r}(t)]_\times\Big)^\mathrm{T} . \tag{4.66}$$

Evaluating (4.41), (4.65) and (4.66) at $t = t_0$ yields the set of equations

$$\mathbf{R}_\kappa(t_0) = \exp\Big([\mathbf{a}_\mathrm{r}(t_0)]_\times\Big)\mathbf{R}_{\kappa_0} \tag{4.67a}$$

$$\left[\boldsymbol{\omega}^\mathcal{K}(t_0)\right]_\times = [\dot{\mathbf{a}}_\mathrm{r}(t_0)]_\times \tag{4.67b}$$

$$\left[\dot{\boldsymbol{\omega}}^\mathcal{K}(t_0)\right]_\times = [\ddot{\mathbf{a}}_\mathrm{r}(t_0)]_\times , \tag{4.67c}$$

from which the initial conditions for $\mathbf{a}_\mathrm{r}(t)$ at $t = t_0$ follow as

$$\mathbf{a}_\mathrm{r}(t_0) = \mathbf{0} \tag{4.68a}$$
$$\dot{\mathbf{a}}_\mathrm{r}(t_0) = \boldsymbol{\omega}^{\mathcal{K}}(t_0) \tag{4.68b}$$
$$\ddot{\mathbf{a}}_\mathrm{r}(t_0) = \dot{\boldsymbol{\omega}}^{\mathcal{K}}(t_0)\ . \tag{4.68c}$$

Evaluation of the same equations (4.41), (4.65) and (4.66) at $t = t_1$ and the introduction of the abbreviation $\mathbf{R}_{\kappa_1} = \mathbf{R}_\kappa(t_1)$ yields another set of equations

$$\mathbf{R}_\kappa(t_1) = \exp\Big([\mathbf{a}_\mathrm{r}(t_1)]_\times\Big)\mathbf{R}_{\kappa_0} = \mathbf{R}_{\kappa_1} \tag{4.69a}$$
$$\Big[\boldsymbol{\omega}^{\mathcal{K}}(t_1)\Big]_\times = \mathbf{R}_{\kappa_1}\mathbf{R}_{\kappa_0}^\mathrm{T}[\dot{\mathbf{a}}_\mathrm{r}(t_1)]_\times\mathbf{R}_{\kappa_0}\mathbf{R}_{\kappa_1}^\mathrm{T} \tag{4.69b}$$
$$\Big[\dot{\boldsymbol{\omega}}^{\mathcal{K}}(t_1)\Big]_\times = \mathbf{R}_{\kappa_1}\mathbf{R}_{\kappa_0}^\mathrm{T}[\ddot{\mathbf{a}}_\mathrm{r}(t_1)]_\times\mathbf{R}_{\kappa_0}\mathbf{R}_{\kappa_1}^\mathrm{T}\ . \tag{4.69c}$$

The equations (4.69) are solved for the initial conditions of $\mathbf{a}_\mathrm{r}(t)$ at $t = t_1$ resulting in

$$[\mathbf{a}_\mathrm{r}(t_1)]_\times = \log\Big(\mathbf{R}_{\kappa_1}\mathbf{R}_{\kappa_0}^\mathrm{T}\Big) \tag{4.70a}$$
$$[\dot{\mathbf{a}}_\mathrm{r}(t_1)]_\times = \mathbf{R}_{\kappa_0}\mathbf{R}_{\kappa_1}^\mathrm{T}\Big[\boldsymbol{\omega}^{\mathcal{K}}(t_1)\Big]_\times\mathbf{R}_{\kappa_1}\mathbf{R}_{\kappa_0}^\mathrm{T} \tag{4.70b}$$
$$[\ddot{\mathbf{a}}_\mathrm{r}(t_1)]_\times = \mathbf{R}_{\kappa_0}\mathbf{R}_{\kappa_1}^\mathrm{T}\Big[\dot{\boldsymbol{\omega}}^{\mathcal{K}}(t_1)\Big]_\times\mathbf{R}_{\kappa_1}\mathbf{R}_{\kappa_0}^\mathrm{T}\ . \tag{4.70c}$$

With the above sets of equations (4.68) and (4.70), together with the boundary conditions (4.47), (4.48) and (4.49), the coefficients $\mathbf{a}_{\mathrm{r},k}$ of the rotational transition trajectory (4.41) are completely specified.

Transition Duration

The specification of the transition duration T_κ is discussed in this section. To keep the TCP velocity within its limits, the transition to a more distant path needs a longer transition duration T_κ than a transition to a closer path. A transition is always composed of a simultaneous translation and rotation, for which the independent transition durations T_t for the translation and T_r for the rotation are introduced.

Based on the distance between the current position $\mathbf{y}_\mathrm{t}(t_0)$ and the goal position $\boldsymbol{\sigma}_{\mathrm{t},i'}\left(\bar{\theta}_{i'}^*\right)$, the translational transition duration T_t is chosen as

$$T_\mathrm{t} = c_\mathrm{t}\left\|\boldsymbol{\sigma}_{\mathrm{t},i'}\left(\bar{\theta}_{i'}^*\right) - \mathbf{y}_\mathrm{t}(t_0)\right\|_2\ , \tag{4.71}$$

with the translational transition duration parameter $c_\mathrm{t} \in \mathbb{R}$. In the same way, based on the quaternion distance between the unit quaternion representation of the orientation of the TCP $\mathbf{Q}^{\mathcal{T}}(t_0)$ and the unit quaternion representation of the desired orientation $\mathcal{D}_{i'}$ $\mathbf{Q}^{\mathcal{D}_{i'}}$, a rotational transition duration T_r is calculated by

$$T_\mathrm{r} = c_\mathrm{r}\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_{i'}}\right\|_2 \tag{4.72}$$
$$\left\{e_{\mathcal{T}}^{\mathcal{D}_{i'}}, \boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_{i'}}\right\} = \mathbf{Q}^{\mathcal{D}_{i'}} \otimes \left(\mathbf{Q}^{\mathcal{T}}(t_0)\right)^{-1}\ , \tag{4.73}$$

with the rotational transition duration parameter $c_\mathrm{r} \in \mathbb{R}$.

The transition duration is found by taking the larger value of the translational transition duration T_t and the rotational transition duration T_r. Hence, the transition duration T_κ is chosen as

$$T_\kappa = \max(T_\mathrm{t}, T_\mathrm{r}) \ . \tag{4.74}$$

At this point, the transition trajectory $\boldsymbol{\kappa}(t)$ is completely specified.

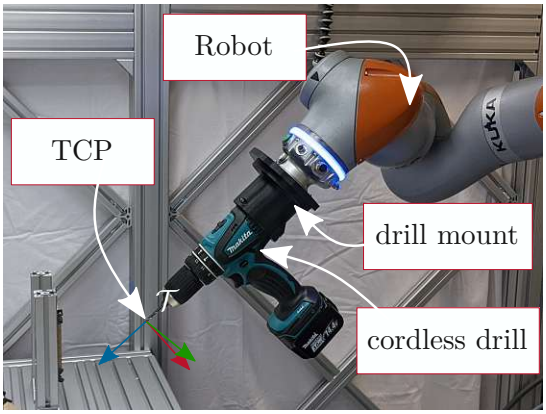
5 Experiments

This chapter presents three experiments for the human-robot interaction modes introduced in Chapter 4. A drilling task with three drilling paths is chosen as representative scenario to demonstrate the teach-in process as well as the collaborative task execution. The presented experiments are based on the same drilling task.

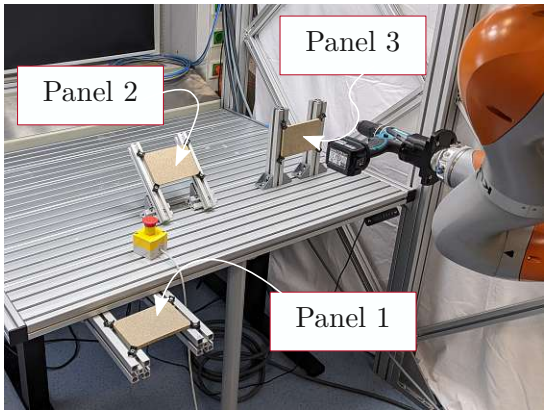
In the first experiment (Section 5.1), the teach-in process is presented, in which the operator can freely move the cordless drill within the workspace of the robot, while the robot carries the load of the drill. Using the Orientation Snap introduced in Section 4.1.2, the orientation of the cordless drill snaps exactly into predefined orientations and the operator programs the paths for the drilling task. The collaborative task execution is presented in two separate experiments. The goal of the task is to drill three consecutive holes in the three mounted plywood panels, which are mounted in different orientations in the robot workspace, see Figure 5.1b. The human operator performs the drilling and operates the cordless drill, while the robot stabilizes the drill on the paths.

In the second experiment (Section 5.2), the drilling task is executed with the Path Snap-In mode introduced in Section 4.1 and in the third experiment (Section 5.3), the same drilling task is executed with the Path Switch mode introduced in Section 4.2. In the collaborative task execution, the robot stabilizes the drill on the drilling paths, while the human operator controls the drill and selects the drilling sequence. With the Path Snap-In mode, the robot only locks in the vicinity of a drilling path. Distant from the drilling paths, the robot only carries the load of the cordless drill, while the movement is controlled by the human operator. Thus, the transition between different drilling paths mainly depends on the human interaction. With the Path Switch mode in contrast, the cordless drill is never *freely* movable. The drill is either stabilized on a drilling path, along which it can be moved, or it is transitioning between two drilling paths, in which case it moves along a defined trajectory.

The experimental setup is depicted in Figure 5.1 and consists of the robot KUKA LBR iiwa 14 R820, a power supply, three plywood panels mounted on a table and a desktop computer. The cordless drill MAKITA DHP446 is attached to the robot endeffector with a 3D-printed drill mount, which uses a clamping mechanism to hold the cordless drill, see Figure 5.1a. With this mount, the cordless drill is completely operable by the human, while being attached to the robot flange. The drawings of the drill mount is found in Appendix A.4. The robot is controlled via the industrial fieldbus system ETHERCAT. The controller is implemented in MATLAB/SIMULINK and runs in the real-time automation software BECKHOFF TWINCAT on the desktop computer. In the experimental setup, no F/T sensor is available. Thus, the external torque $\boldsymbol{\tau}_\mathrm{e}$ is not fed back in the control concept. For the belief system, however, the estimation of the external force $\mathbf{f}_\mathrm{e}^\mathrm{est}$ according to (4.25) and (4.27) is used.



(a) Cordless drill attached to the robot endeffector with a 3D-printed drill mount.



(b) Experimental setup with the cordless drill and the three plywood panels mounted in different orientations in the robot workspace.

Figure 5.1: Experimental setup for the selected drilling task.

5.1 Teach-In Experiment

The first experiment demonstrates the teach-in of the robotic task. To teach the drilling paths to the robot, the tip of the drill must be positioned at the desired drill hole location of each plywood panel in the correct drilling orientation. The Orientation Snap introduced in Section 4.1.2 assists the operator to exactly match the alignment of the drill bit with the desired drilling direction.

To be able to snap into the desired drilling orientations, N_φ, N_ϑ and N_ψ must be chosen, such that the drilling orientations are included in the set $\mathcal{R}_{\text{os}}$ of (4.23). Figure 5.1b shows the orientations of the mounted plywood panels. To snap into the orientation of panel 2, an angular grid of $\Delta\vartheta = 45° = \frac{\pi}{4}$ is needed, thus $N_\vartheta = 5$. For φ and ψ, an angular grid of $\Delta\varphi = \Delta\psi = 90° = \frac{\pi}{2}$ is chosen, hence, $N_\varphi = N_\psi = 4$. With the selected values $N_\varphi = 4$, $N_\vartheta = 5$ and $N_\psi = 4$, all of the desired drilling orientations are included in $\mathcal{R}_{\text{os}}$, cf. (4.23). Based on the pose of the drill when the tip of the drill touches the respective plywood panel in the desired drill hole location, the paths Σ_1, Σ_2 and Σ_3 are constructed as straight paths in the direction of the drill axis starting at the corresponding plywood panel. Figure 5.2 depicts the paths resulting from the teach-in process Σ_1, Σ_2 and Σ_3 as well as the plywood panels. To ease the three-dimensional imagination, Figure 5.2 and the following three-dimensional figures include projections of the 3D-depictions onto the xy-, yz- and xz-planes. Table A.3 summarizes the chosen controller parameters for this experiment.

The movement of the drill during the teach-in process is depicted in Figure 5.3. The color of the traversed trajectory indicates the rotational weighting factor μ_{r}. A low rotational weighting factor μ_{r}, i. e. where the trajectory is blue, indicates that the robot is only carrying the load of the drill and does not proactively support the operator otherwise. In that case, the drill is freely moveable *and* rotatable within the workspace of the robot.

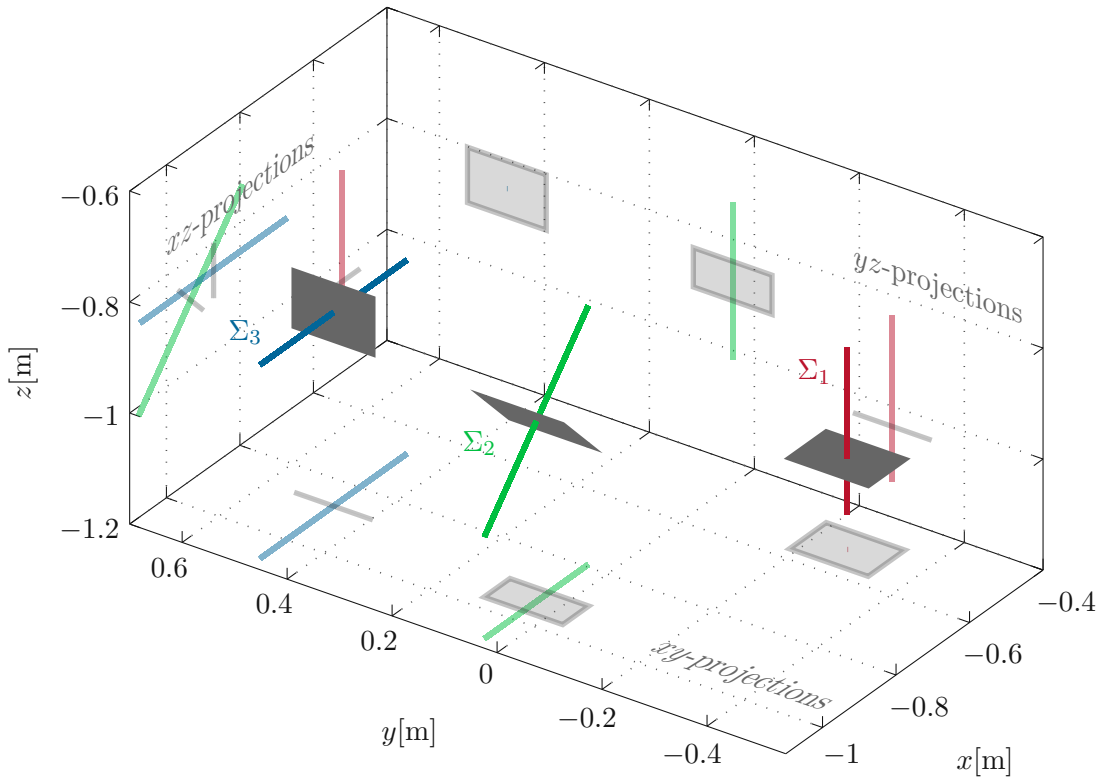


Figure 5.2: The drilling paths Σ_1, Σ_2 and Σ_3 resulting from the teach-in process together with the mounted plywood panels.

A large rotational weighting factor μ_r, i. e. where the trajectory is red, indicates that the drill is snapped into a defined orientation. In that case, the robot stabilizes the orientation, while a translational movement of the drill is still allowed. To teach a new pose, at first the orientation of the TCP $\mathbf{R}^\mathcal{T}$ is aligned with the desired orientation of the path to teach. In the second step, the position of the tip of the drill $\mathbf{y}_\mathrm{t}$ is moved to the position to teach, while the assistance from the Orientation Snap keeps the TCP in the same orientation.

The complete teach-in process for the drilling task is split into multiple steps, specified by encircled numbers ⊙ in Figure 5.3. The tip of the drill starts at ① and is approached to the first plywood panel ②. The position of the drill tip $\mathbf{y}_\mathrm{t}$ and the orientation of the drill $\mathbf{R}^\mathcal{T}$ are stored in the robot program and then the robot is moved towards panel 2 by the operator. Between ② and ③, the orientation of the drill has to be adjusted because the drill orientation for panel 2 is different than the drill orientation for panel 1. Thus, the operator releases the orientation snap by turning the TCP orientation away from the currently stabilized orientation and the color of the trajectory in Figure 5.3 changes to blue. At ③, the drill snaps into the orientation for panel 2, which is indicated by the color change of the trajectory to red, and thus $\mu_\mathrm{r} \approx 1$. Note that the position of the drill $\mathbf{y}_\mathrm{t}$ and the orientation of the drill $\mathbf{R}^\mathcal{T}$ are controlled separately. Thus, the orientation of the TCP $\mathbf{R}^\mathcal{T}$ can be snapped, although the position of the TCP $\mathbf{y}_\mathrm{t}$ is not snapped. At the position ③, the orientation is already snapped correctly, while the drill head still needs

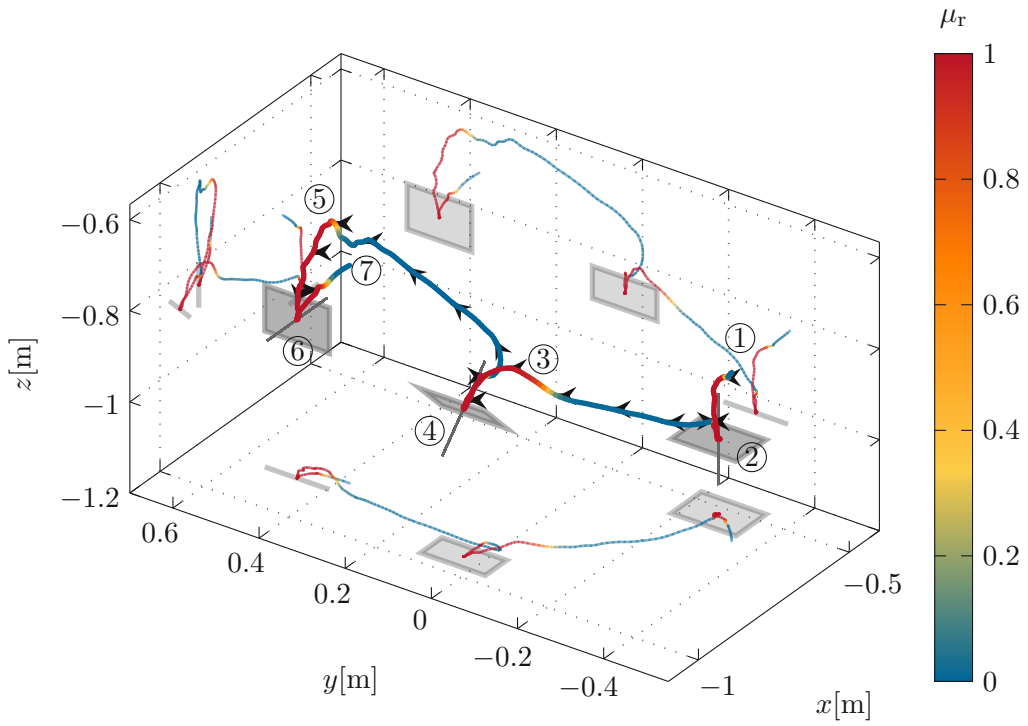


Figure 5.3: Trajectory of the TCP $\mathbf{y}_t$ in the workspace during the teach-in of the robotic task, where the color represents the value of the rotational weighting factor μ_r.

to be approached to the plywood panel ④. At the contact point ④, the position of the drill tip $\mathbf{y}_t$ and the orientation of the drill $\mathbf{R}^{\mathcal{T}}$ are stored in the robot program. Finally, the same procedure is repeated for panel 3. Between ④ and ⑤, the Orientation Snap is inactive, at ⑤ the orientation snaps into the desired drilling orientation for panel 3 and the tip of the drill is approached to the desired drill hole location of the plywood panel ⑥. At this point, the values of $\mathbf{y}_t$ and $\mathbf{R}^{\mathcal{T}}$ are stored in the robot program. At the end of the teach-in process, the drill is taken back from the panels and $\mathbf{y}_t$ ends at ⑦.

After the teach-in process, the stored positions $\mathbf{y}_t$ at ②, ④ and ⑥ in combination with the stored orientations $\mathbf{R}^{\mathcal{T}}$ at ②, ④ and ⑥ are used to construct the paths Σ_1, Σ_2 and Σ_3 as depicted in Figure 5.2.

To further analyze the behavior of the system during the teach-in process and the corresponding signals, Figure 5.4 plots the rotational weighting factor $\mu_{r,i}$ for the relevant orientations of panel 1 ($i = 1$), panel 2 ($i = 2$) and panel 3 ($i = 3$), the norm of the corresponding vector parts of the quaternion errors $\left\|\varepsilon_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$, the norm of the estimated external rotational force $\left\|\mathbf{f}_{e,r}^{est}\right\|_2$ and the estimated external translational force $\left\|\mathbf{f}_{e,t}^{est}\right\|_2$ and the rotational damping weighting factor ν_r over time. The encircled numbers ① – ⑥ in Figure 5.4 correspond to the locations in Figure 5.3. Analyzing the progression of the rotational weighting factor $\mu_{r,i}$, one can see that the rotational weighting factor $\mu_{r,i}$ alternates between $\mu_{r,i} = 0$ and $\mu_{r,i} = 1$ for the individual orientations. It raises

when the orientation of the TCP $\mathbf{R}^{\mathcal{T}}$ moves close to the respective predefined orientation $\mathbf{R}^{\mathcal{D}_i}, i \in \{1,2,3\}$. At first, between ① and ②, the norm of the rotational error between the TCP $\mathbf{R}^{\mathcal{T}}$ and the desired orientation $\mathbf{R}^{\mathcal{D}_1}$ for panel 1, displayed in the plot for $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$, decreases, because the operator rotates the drill close to the desired drilling orientation for panel 1. In reaction to the decreasing error $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_1}\right\|_2$, the rotational weighting factor for the specfic drilling orientation $\mu_{\mathrm{r},1}$ increases and reaches 1. The larger rotational weighting factor $\mu_{\mathrm{r},1}$ means that the robot now takes control over the orientation of the TCP and thus controls the rotational error $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_1}\right\|_2$ for the orientation of panel 1 to $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_1}\right\|_2 \approx 0$. The same behavior is visible for the transition into the drilling orientation of panel 2 (transition from ② to ③), and for the transition into the drilling orientation of panel 3 (transition from ④ to ⑤).

The rotational weighting factor $\mu_{\mathrm{r},i}$ decreases when the drill is rotated out of the respective predefined orientation from the set $\mathcal{R}_{\mathrm{os}}$. To rotate the drill out of a snapped orientation $\mathbf{R}^{\mathcal{D}_i}$, the external force $\mathbf{f}_{\mathrm{e}}$ applied by the human operator has to increase, which causes the error $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$ to raise and consequently $\mu_{\mathrm{r},i}$ to fall. Although this increase in the external force $\mathbf{f}_{\mathrm{e}}$ at the transition from $\mu_{\mathrm{r},i} = 1$ to $\mu_{\mathrm{r},i} = 0$ at $t = 17\,\mathrm{s}$ and at $t = 30\,\mathrm{s}$ is slightly recognizable in the estimation $\mathbf{f}_{\mathrm{e}}^{\mathrm{est}}$ in Figure 5.4, it is not as significant as expected. This peak is even smaller than at the transition from $\mu_{\mathrm{r},i} = 0$ to $\mu_{\mathrm{r},i} = 1$ at $t = 8\,\mathrm{s}$ and at $t = 22\,\mathrm{s}$. This leads to the conclusion that the estimation of $\mathbf{f}_{\mathrm{e}}^{\mathrm{est}}$ according to (4.25) and (4.27) is not accurate in a dynamic transition when the TCP is not locked onto a distinct path. Especially the estimation of the external force $\mathbf{f}_{\mathrm{e}}^{\mathrm{est}}$ at the transition between $\mu_{\mathrm{r},i} = 0$ and $\mu_{\mathrm{r},i} = 1$ seems inaccurate. This deviation from the real value may result from the increasingly dynamic motion, as soon as an orientation is attracting. To validate the estimation of the external force $\mathbf{f}_{\mathrm{e}}^{\mathrm{est}}$ in the given scenario, a F/T sensor may be attached to the endeffector and the values of the F/T sensor may be compared to the estimation $\mathbf{f}_{\mathrm{e}}^{\mathrm{est}}$. However, since the external force $\mathbf{f}_{\mathrm{e}}^{\mathrm{est}}$ is only included as additional information in Figure 5.4 and is not fed back into the system, such a validation is not included in this work.

The rotational weighting factor for the damping ν_{r} evolves exactly opposite to the sum of rotational weighting factors μ_{r}. It is decreasing with decreasing error $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$ and increasing with increasing error $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$. From a control perspective, this means the virtual input for the rotational damping $\mathbf{v}_{\mathrm{dam,r}}$ according to (4.9) takes over when the Orientation Snap is not active, cf. (4.12). This matches the desired behavior described in Section 4.1.1. The different behavior of the controller when the orientation is snapped-in compared to the behavior when the rotation is damped is visible in the shape of the error signals $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$ in Figure 5.4. When the orientation is snapped-in, e. g. between ③ and ④, the errors $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$ for all orientations are approximately constant. The orientation is stabilized during these time periods. When the orientation is *not* snapped-in and only the damping is active, i. e. $\nu_{\mathrm{r}} \approx 1$, e. g. between ② and ③, the errors $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$ vary, since there is no preferred orientation to be stabilized by the controller.

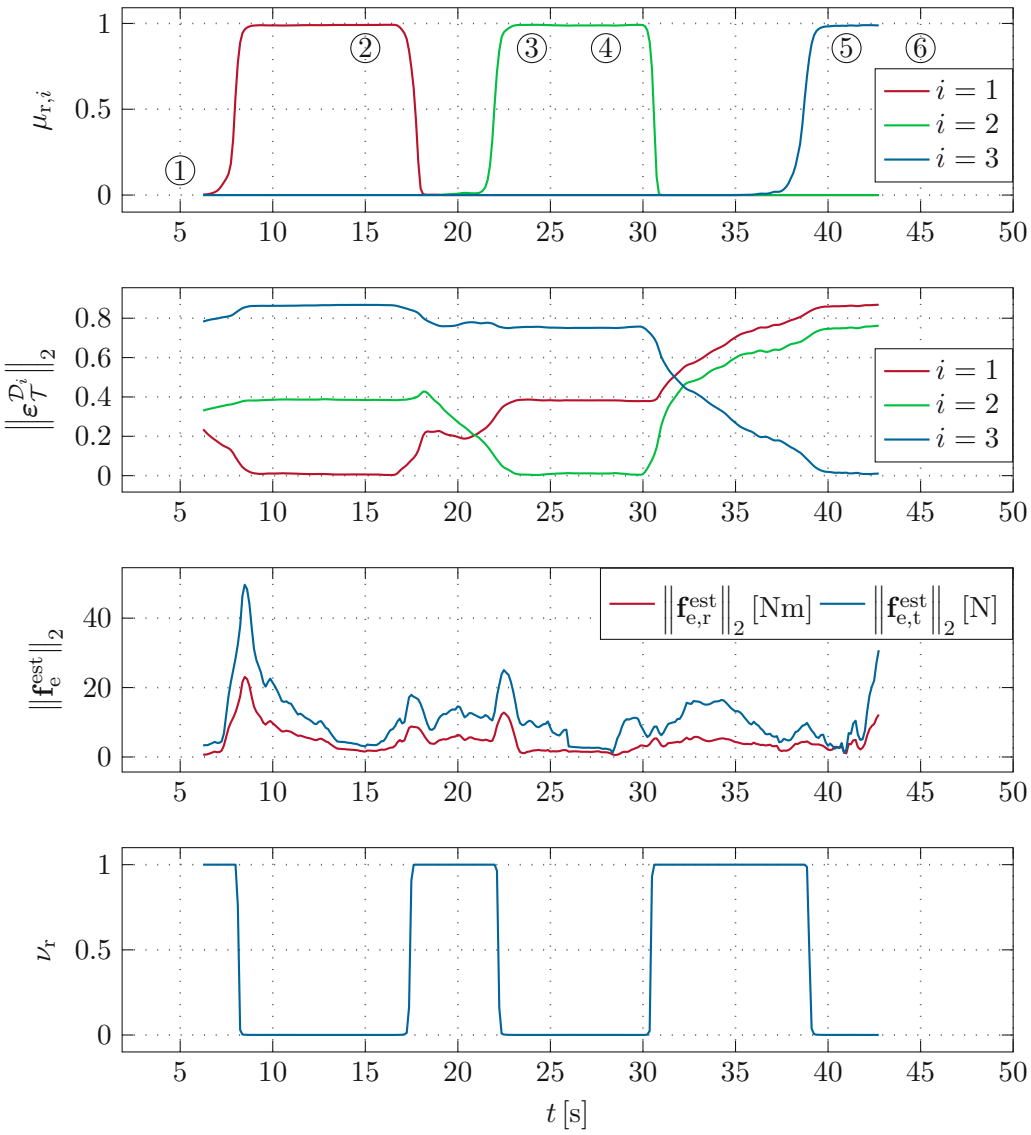


Figure 5.4: Rotational weighting factor $\mu_{r,i}$, norm of the rotational error $\left\|\varepsilon_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$ for the relevant orientations of panel 1, 2 and 3, norm of the estimated external force $\left\|\mathbf{f}_e^{est}\right\|_2$ and rotational weighting factor for the damping ν_r plotted over time during the robotic teach-in process.

5.2 Path Snap-In Experiment

With the performed teach-in process of Section 5.1, the drilling paths Σ_1, Σ_2 and Σ_3 are defined and this section presents the collaborative task execution. First, the collaborative task execution is performed with the Path Snap-In introduced in Section 4.1. The task execution utilizing the Path Switch mode follows in Section 5.3.

In this experiment, a hole is drilled in each of the three plywood panels in one consecutive

procedure. The robot provides support during the drilling process by keeping the drill on the path, while the operator's task is to operate the drill and move it along the path. The drilling paths Σ_1, Σ_2 and Σ_3 are defined as combinations of translation and rotation (in contrast to the teach-in experiment in Section 5.1). Therefore, translation and rotation are controlled simultaneously according to (4.11). To keep the robot within its limits, the weighting factor μ_i is chosen as

$$\mu_i = \max(\mu_{\mathrm{t},i}, \mu_{\mathrm{r},i}) \ , \tag{5.1}$$

such that the corresponding virtual inputs $\mathbf{v}_{\mathrm{t},i}$ and $\mathbf{v}_{\omega,i}$ according to (4.11) are only applied if both weighting factors originating from the translational distance $d_{\mathrm{t},i}$ *and* the rotational distance $d_{\mathrm{r},i}$ are large enough.

For the task execution, the impedance matrices $\mathbf{K}^{\mathrm{d}}$, $\mathbf{D}^{\mathrm{d}}$ and $\mathbf{M}^{\mathrm{d}}$ are chosen to be diagonal and the same for every path, see Appendix A.2.2. Especially, the tangential virtual spring parameter $k_{\parallel}^{\mathrm{d}}$ is chosen as $k_{\parallel}^{\mathrm{d}} = 0$, which allows the operator to move the drill along the path without a restoring force along the path direction. Additionally, the drill should be positioned *on* the path and thus, the desired path-based coordinates in transversal direction $\xi_{\perp}^{\mathrm{d}}, \xi_{\pitchfork}^{\mathrm{d}}$ and their derivatives $\dot{\xi}_{\perp}^{\mathrm{d}}, \dot{\xi}_{\pitchfork}^{\mathrm{d}}, \ddot{\xi}_{\perp}^{\mathrm{d}}, \ddot{\xi}_{\pitchfork}^{\mathrm{d}}$ are chosen as $\xi_{\perp}^{\mathrm{d}} = \xi_{\pitchfork}^{\mathrm{d}} = \dot{\xi}_{\perp}^{\mathrm{d}} = \dot{\xi}_{\pitchfork}^{\mathrm{d}} = \ddot{\xi}_{\perp}^{\mathrm{d}} = \ddot{\xi}_{\pitchfork}^{\mathrm{d}} = 0$.

Figure 5.5 shows the traversed trajectory of the TCP $\mathbf{y}_{\mathrm{t}}$ in the workspace during the collaborative drilling process. The color of the traversed trajectory corresponds to the sum of weighting factors μ from (4.6), which is $\mu = 1$ when snapped into any of the paths (red) and $\mu = 0$ distant to all paths (blue). Similar to the teach-in process presented in Section 5.1, a large value of $\mu \approx 1$ means that the controller stabilizes the drill on a path. If $\mu \approx 0$, the robot does not actively stabilize the robot on a path. Comparing Figure 5.5 with Figure 5.3, two major differences are noticable. During the teach-in process, only the Orientation Snap was used and hence, the TCP position $\mathbf{y}_{\mathrm{t}}$ never had a desired position to snap into. The traversed trajectory of the TCP $\mathbf{y}_{\mathrm{t}}$ in Figure 5.3 has a curvy and random shape. This aspect only partially applies to the traversed trajectory of $\mathbf{y}_{\mathrm{t}}$ in Figure 5.5 with the Path Snap-In mode. The trajectory has a similar curvy and random shape as long as the weighting factor μ is close to zero $\mu \approx 0$. However, the trajectory is distinct when the weighting factor μ is close to one $\mu \approx 1$, i. e. in the vicinity of the drilling paths Σ_1, Σ_2 and Σ_3. This demonstrates the desired behavior of the Path Snap-In mode: The TCP remains freely moveable and rotatable within the robot's workspace distant from any path, while providing support in the vicinity of the paths. Comparing the projections onto the xz- and xy- plane of Figure 5.5 with the projections onto the xz- and xy- plane of Figure 5.3, one can see that the trajectory pierces through the panels of Figure 5.5, while it just touches the panels of Figure 5.3. This shows that the holes were actually drilled in the collaborative execution of the drilling task.

To further analyze the drilling process, the evolution of the position of the drill $\mathbf{y}_{\mathrm{t}}$ along the paths Σ_1, Σ_2 and Σ_3 is discussed in the following. Figure 5.6 shows the evolution of the tangential path parameters $\xi_{\parallel,i}$ and their derivatives $\dot{\xi}_{\parallel,i}$ for the three drilling paths Σ_i, $i \in \{1, 2, 3\}$ together with the corresponding weighting factors μ_i over time.

Note that the PFC updates each optimal path parameter θ_i^*, all of the path-based coordinates $\boldsymbol{\xi}_i$ and all of the derivatives of the path-based coordinates $\dot{\boldsymbol{\xi}}_i$, $i \in \{1, 2, 3\}$, in every time step. However, their values only become relevant, if the corresponding belief

μ_i has risen. Otherwise, i. e. if $\mu_i = 0$, the PFC for the specific path Σ_i is not acting on the robot, and thus, the path-based coordinates $\boldsymbol{\xi}_i$ and their derivatives $\dot{\boldsymbol{\xi}}_i$ of that specific path Σ_i provide no relevant information. To highlight this fact, the values of $\xi_{\parallel,i}$ and $\dot{\xi}_{\parallel,i}$, $i \in \{1,2,3\}$ in Figure 5.6 are plotted as dashed lines when no path is selected, and emphasized as thick lines with colored background if the distinct path is selected, i. e. the according weighting factor is $\mu_i \approx 1$. Since the calculation of $\xi_{\parallel,i}$ is based on an integral equation, cf. (3.18), the initial values $\theta_{0,i}$ define the further evolution of $\xi_{\parallel,i}$. In the plot of Figure 5.6, the initial values are chosen, such that $\xi_{\parallel,i} = 0$ as soon as the path Σ_i gets selected. This eases the comparison between the different paths Σ_i and has no influence on the control behavior, since $k_{\parallel}^{\mathrm{d}} = 0$.

Figure 5.6 shows the distinct steps of the drilling task execution. The operator drills the first hole on the path Σ_1 at $t = 6\,\mathrm{s}$ and at $t = 17\,\mathrm{s}$, moves on the the second panel on path Σ_2, where the corresponding weighting factor μ_2 rises between $t = 21\,\mathrm{s}$ and $t = 34\,\mathrm{s}$, and finally goes to Σ_3 at $t = 45\,\mathrm{s}$ to end the execution. Comparing the waveforms in Figure 5.6 of the tangential path parameters $\xi_{\parallel,i}$ and the derivatives $\dot{\xi}_{\parallel,i}$, $i \in \{1,2,3\}$ of the individual paths Σ_1, Σ_2 and Σ_3 during the emphasized active time spans, similarities are clearly noticeable. Since the act of drilling is the same along the individual drilling paths Σ_1, Σ_2 and Σ_3, these similarities are not surprising. Extracting a single drilling

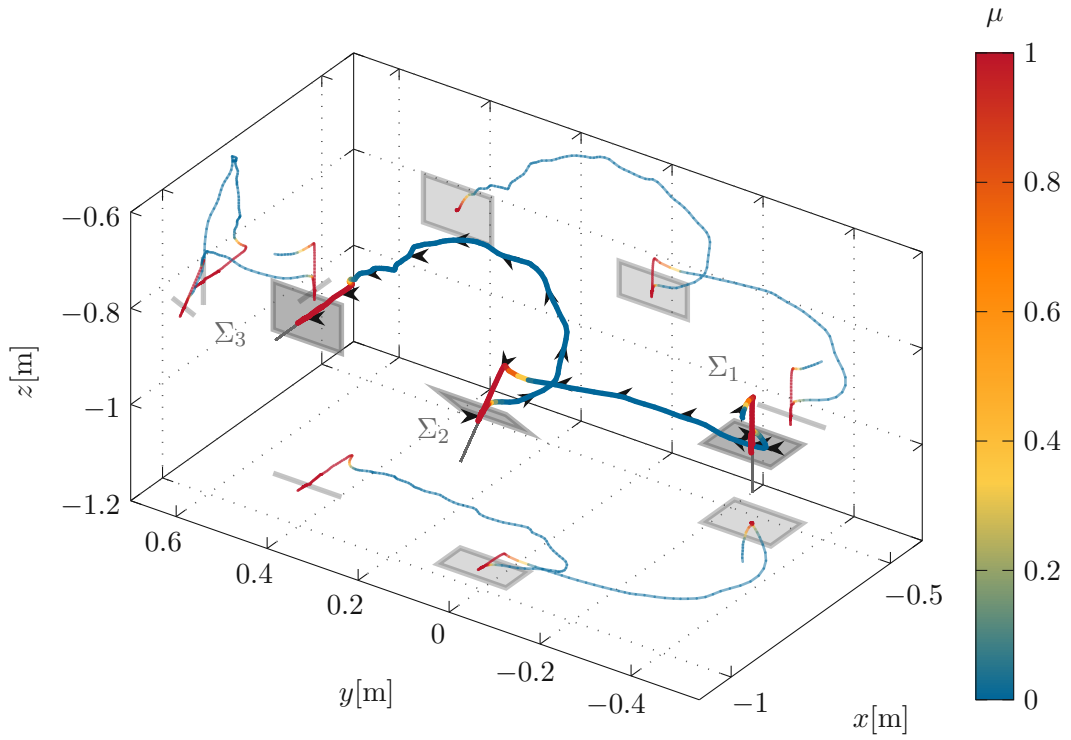


Figure 5.5: Trajectory of the TCP $\mathbf{y}_{\mathrm{t}}$ in the workspace during the task execution with the Path Snap-In, where the color represents the sum of the weighting factors μ from (4.6).

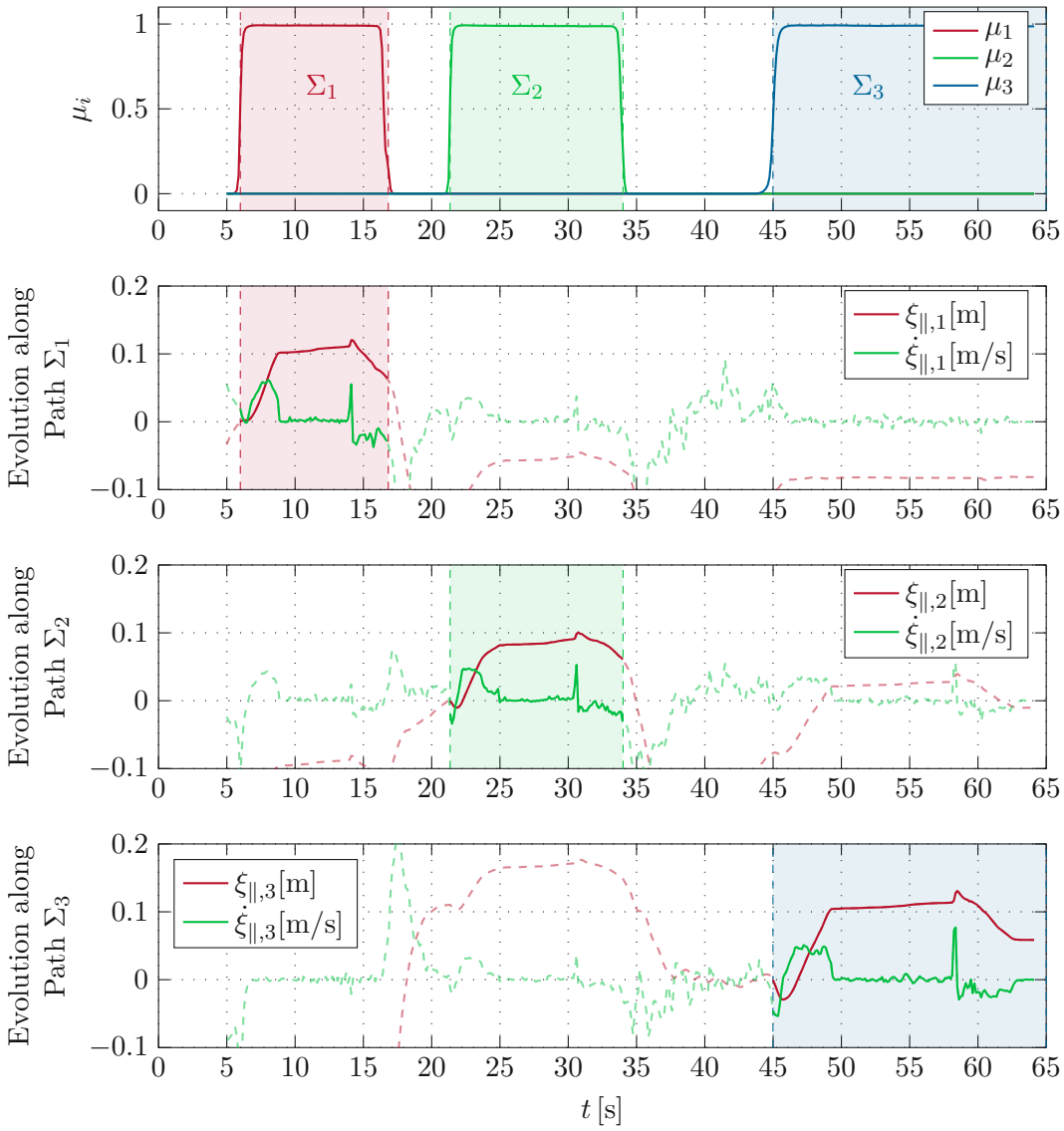


Figure 5.6: Weighting factor μ_i for each path Σ_i with the tangential path parameter $\xi_{\parallel,i}$ and its derivative $\dot{\xi}_{\parallel,i}$ during the Snap-In.

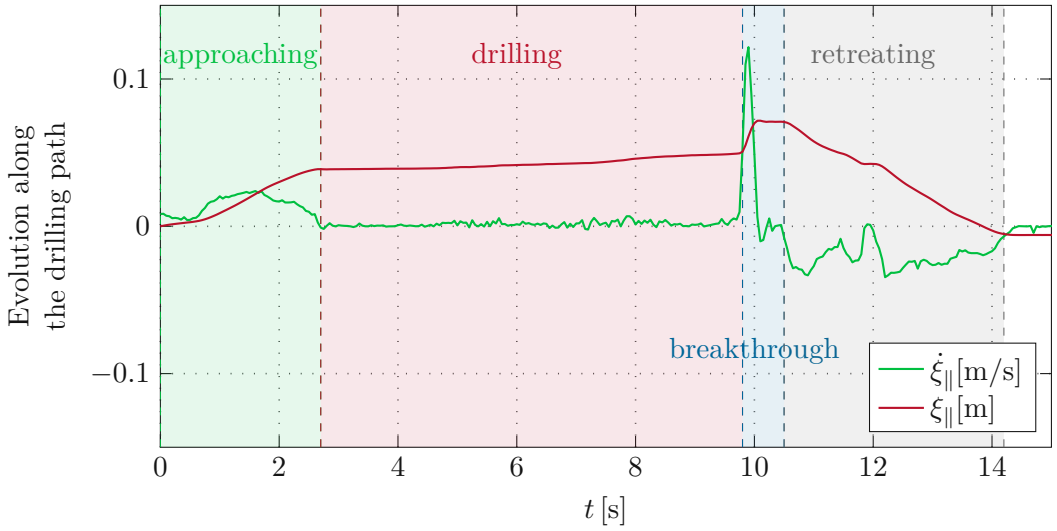


Figure 5.7: Path evolution of a single drilling process with four distinguishable phases.

process, four different phases are distinguishable, plotted in Figure 5.7. In the first phase, the tip of the drill is approached to the plywood panel. This can be seen by a rather fast increase of $\xi_{\parallel}$ in Figure 5.7, which means the tip is moving rather fast along the path in comparison to the subsequent phases. The second phase is the actual drilling, during which the position of the drill only changes gradually, i. e. $\xi_{\parallel}$ is increasing slowly as the drill cuts through the plywood. At the same time, the interaction forces between the drill head and the plywood are noticeable in the variations and fluctations of $\dot{\xi}_{\parallel}$. The end of the actual drilling process is visible as a steep increase of $\xi_{\parallel}$ and a large value of $\dot{\xi}_{\parallel}$. This signalizes the breakthrough through the panel. This steep evolution of the path parameter at the breakthrough is caused by the operator, who keeps on pushing the drill through the plywood. As soon as there is no plywood anymore to counteract the operator's force, the force causes an abrupt increase in the path parameter. At the end, the drill is taken back through the drilled hole, recognizable at the decreasing value of $\xi_{\parallel}$ in the last phase.

After the analysis along the direction of the paths, the error in transversal direction as well as the rotational error are analyzed next. Figure 5.8 plots the norm of the vector part of the quaternion error $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$, as an absolute value for the rotational error and the norm of the transversal deviation from the path, $\tilde{\xi}_i = \sqrt{\xi_{\perp,i}^2 + \xi_{\pitchfork,i}^2}, i \in \{1,2,3\}$, as an absolute value for the transversal error, and the weighting factor μ over time.

In subplots 2 and 3, the rotational error $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$ and the transversal error $\tilde{\xi}_i$ are plotted over the complete time horizon and in subplots 4 and 5, the rotational error $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$ and the transversal error $\tilde{\xi}_i$ are only plotted during the time span of the drilling process on the respective path Σ_i (more precisely, when $\mu_i > 0.95$). Comparing Figure 5.8 with Figure 5.4, similarities between the Orientation Snap in Section 5.1 and the Path Snap-In discussed here, are noticeable. The weighting factor μ in Figure 5.8 behaves analogous to μ_{r} in Figure 5.4, meaning it alternates between $\mu = 0$ and $\mu = 1$ in the same way. The

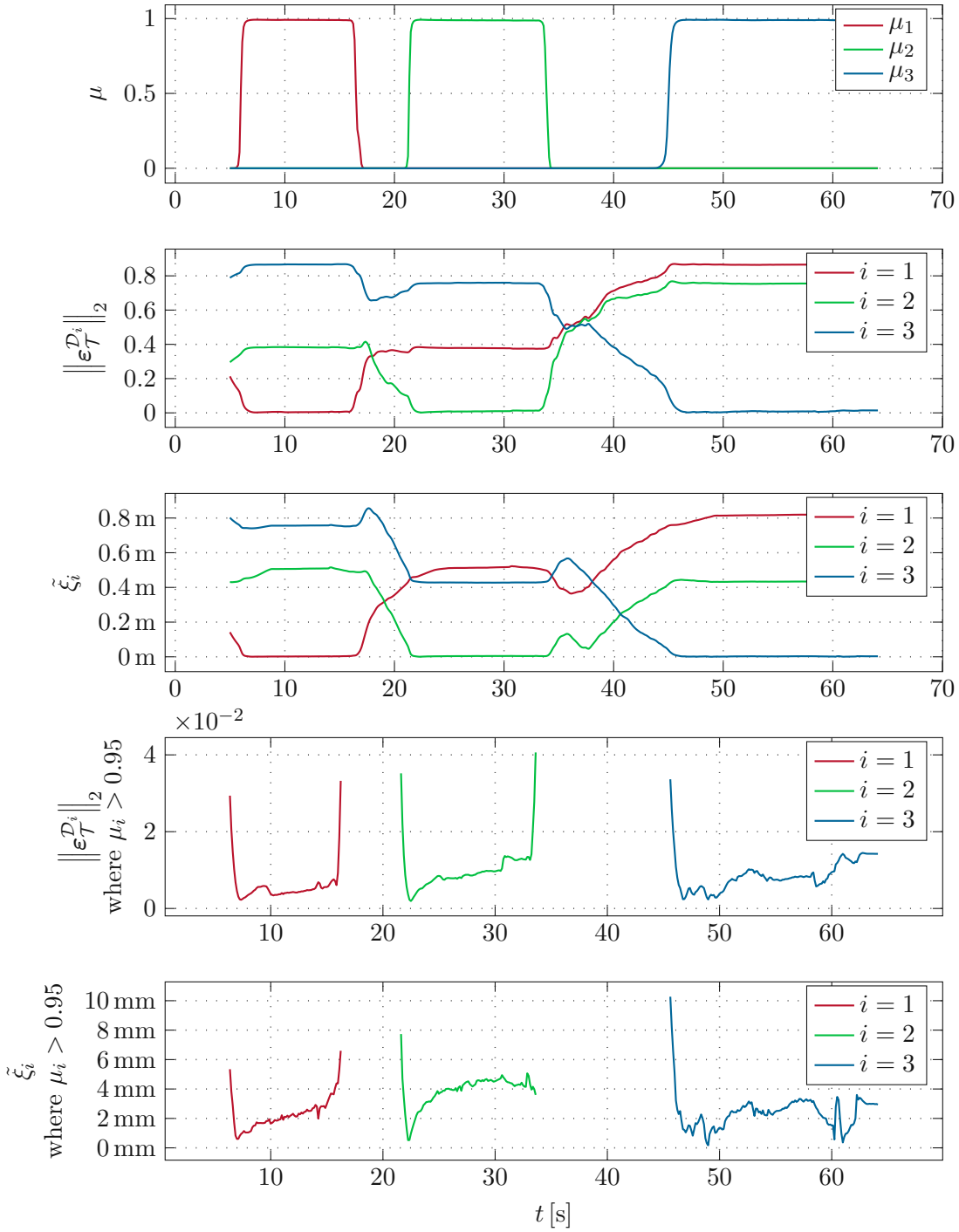


Figure 5.8: Rotational error $\left\|\varepsilon_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$ and transversal error $\tilde{\xi}_i$ between the TCP and the drilling paths Σ_i as well as the corresponding weighting factor μ_i plotted over time.

rotational error $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$ in Figure 5.8 first approaches $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_1}\right\|_2 = 0$ for panel 1 at $t = 7\,\mathrm{s}$, while $\mu = 1$, then for panel 2 $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_2}\right\|_2 = 0$ at $t = 22\,\mathrm{s}$, and finally for panel 3 $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_3}\right\|_2 = 0$ at $t = 47\,\mathrm{s}$, similar to the waveforms in Figure 5.4. Since the drilling task and the drilling paths Σ_i are the same as during the teach-in, these similarities between $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$, μ and μ_r are not surprising.

The difference between the execution with the Path Snap-In and the Teach-in with the Orientation Snap lies in the translational part. Figure 5.8 shows an additional waveform, i. e. the translational error in the transversal direction $\tilde{\xi}_i$, which was not included in Figure 5.4 for the teach-in process. When $\mu_i = 1$, i. e. the Path Snap-In for Σ_i is active, *both* the rotational error $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$ and the transversal error $\tilde{\xi}_i$ are controlled to $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2 \approx 0$ and $\tilde{\xi}_i \approx 0$. This shows that the controller stabilizes the position *and* the orientation of the drill on the drilling path. Consequently, to snap into a path Σ_i, the drill must be positioned close to the path in the correct orientation. A position $\mathbf{y}_\mathrm{t}$ close to the path $\boldsymbol{\sigma}_{\mathrm{t},i}(\theta_i^*)$ or an orientation $\mathbf{R}^{\mathcal{T}}$ close to the path orientation $\mathbf{R}^{\mathcal{D}_i}$ alone is not enough. To prove this statement with measurement data, in the subplots 4 and 5 of Figure 5.8 only values of the rotational error $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$ and the transversal error $\tilde{\xi}_i$ are plotted for time spans in which the respective weighting factor $\mu_i > 0.95$. Thus, subplots 4 and 5 contain the same signals as subplots 2 and 3, with a scaled-up y-axis. This representation shows that a change in the weighting factor μ_i is caused by a change in the rotational error $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$, the transversal error $\tilde{\xi}_i$ or both. In subplot 4 and 5 at $t = 16\,\mathrm{s}$, both the rotational error and the transversal error increase, causing the TCP to be released from the Path Snap-In of Σ_1. At $t = 33\,\mathrm{s}$ on the contrary, an increase in the rotational error alone is enough for the TCP to be released from the Path Snap-In of Σ_2, see subplots 4 and 5 in Figure 5.8. When the belief drops below $\mu_2 = 0.95$ (i. e. when the depictions of $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_2}\right\|_2$ in subplot 4 and $\tilde{\xi}_2$ in subplot 5 end), the rotational error $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_2}\right\|_2$ has increased, while the translational error $\tilde{\xi}_2$ has not. This leads to the conclusion that the Path Snap-In was released due to the rotational error alone. Thus, the data proves that as soon as the operator significantly increases the error in *either* the orientation *or* the position, the Path Snap-In is released. To keep the TCP locked on a path both the orientation and the position must match.

The error during a single drilling process of this experiment lies at a maximum of about $\tilde{\xi}_i = 5\,\mathrm{mm}$ for the translation and about 0.01 for $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2$, see Figure 5.8. In axis-angle representation, the total deviation angle is obtained by $\phi_{\mathcal{T}}^{\mathcal{D}_i} = 2\,\mathrm{atan2}\left(\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2, e_{\mathcal{T}}^{\mathcal{D}_i}\right)$ which is a more intuitive value for the rotational error. The deviation angle $\phi_{\mathcal{T}}^{\mathcal{D}_i}$ thus reaches a maximum of $\phi_{\mathcal{T}}^{\mathcal{D}_i} = 1.1°$ during the drilling process. To release the Path Snap-In for a path, a deviation of about $\tilde{\xi}_i = 5\,\mathrm{mm}$ from the path in translation combined with a deviation of about $\left\|\boldsymbol{\varepsilon}_{\mathcal{T}}^{\mathcal{D}_i}\right\|_2 = 0.04$ or $\phi_{\mathcal{T}}^{\mathcal{D}_i} = 4.6°$ from the path orientation is required. This causes a decrease of the weigthing factor μ_i to $\mu_i = 0.95$. These values will be compared with the values for the Path Switch in the next section.

5.3 Path Switch Experiment

In this section, the collaborative execution of the drilling task with the Path Switch introduced in Section 4.2 is discussed. The experiment presented in this section is similar to the Path Snap-In experiment of Section 5.2, i. e. three holes are drilled in the three plywood panels. This time, the assistance of the Path Switch is utilized, which stabilizes the drill on the programmed paths Σ_1, Σ_2 and Σ_3, specified during the teach-in process in Section 5.1. The chosen control parameters are given in Appendix A.2. With the same considerations as for the experiment with the Path Snap-In in Section 5.2, the desired virtual spring in tangential direction $k_{\parallel}^{\mathrm{d}}$ is chosen as $k_{\parallel}^{\mathrm{d}} = 0$ and the desired path-based coordinates in transversal direction as well as their derivatives are chosen as $\xi_{\perp}^{\mathrm{d}} = \xi_{\pitchfork}^{\mathrm{d}} = \dot{\xi}_{\perp}^{\mathrm{d}} = \dot{\xi}_{\pitchfork}^{\mathrm{d}} = \ddot{\xi}_{\perp}^{\mathrm{d}} = \ddot{\xi}_{\pitchfork}^{\mathrm{d}} = 0$.

Figure 5.9 shows the traversed trajectory of the TCP $\mathbf{y}_{\mathrm{t}}$ during the collaborative execution of the drilling task with the Path Switch mode. Since the Path Switch mode does not use weighting factors μ_i for its control concept, the color of the traversed trajectory in Figure 5.9 does not correspond to a weighting factor μ_i. Instead, the color signalizes the currently active controller. A red section of the traversed trajectory signalizes that the PFC (4.43) is applied, while a blue part of the traversed trajectory indicates that the TTC (4.44) is applied. The change from *stabilizing the TCP on a drilling path* to *transitioning the TCP between two drilling paths* is a continuous transition for the Path Snap-In, while it is a discrete switch for the Path Switch. The color palettes of Figure 5.5 and Figure 5.9 underline this circumstance. While the color palette in Figure 5.5 is a continuous transition from red to blue, there are only two distinct colors in Figure 5.9.

Comparing the traversed trajectory $\mathbf{y}_{\mathrm{t}}$ *on* the drilling paths Σ_1, Σ_2 and Σ_3 of the Path Switch (red part of the trajectory in Figure 5.9) to the traversed trajectory $\mathbf{y}_{\mathrm{t}}$ *on* the drilling paths Σ_1, Σ_2 and Σ_3 of the Path Snap-In (red part of the trajectory in Figure 5.5), hardly any differences are noticeable. A more detailed comparison of the rotational errors and transversal errors follows below. The significant difference lies in the system behavior while *transitioning* from one drilling path Σ_i to another drilling path $\Sigma_{i'}$. As already discussed in Section 5.2, the traversed trajectory between two drilling paths of the task execution with the Path Snap-In has a curvy and random shape, because the human operator is responsible to move the drill from one drilling path Σ_i to the next drilling path $\Sigma_{i'}$. During the Path Switch in contrast, the transition trajectory $\boldsymbol{\kappa}(t)$ is explicitly generated by the system, according to the considerations in Section 4.2.2. Thus, the transition between two drilling paths Σ_i and $\Sigma_{i'}$ follows a smooth and direct trajectory, as depicted in Figure 5.9.

The transition between two drilling paths Σ_i and $\Sigma_{i'}$ is initiated by the human operator with the external force $\mathbf{f}_{\mathrm{e,t}}$. This applied force is also depicted in Figure 5.9. The green arrows show the estimation of the external force vector $\mathbf{f}_{\mathrm{e,t}}^{\mathrm{est}}$ according to (4.25) and (4.27). The plotted force in Figure 5.9 shows only forces influencing the control concept, i. e. where $\left\|\mathbf{f}_{\mathrm{e,t}}^{\mathrm{est}}\right\|_2 > 50\,\mathrm{Nm}$. Note that the external forces $\mathbf{f}_{\mathrm{e,t}}^{\mathrm{est}}$ in Figure 5.9 are applied at the beginning of each transition and they point towards the desired drilling path $\Sigma_{i'}$ to which the TCP transition occurs. The practical experiment revealed that it is difficult for the human operator to align the applied force direction $\mathbf{e}_{\mathrm{f}}$ with the direction in which

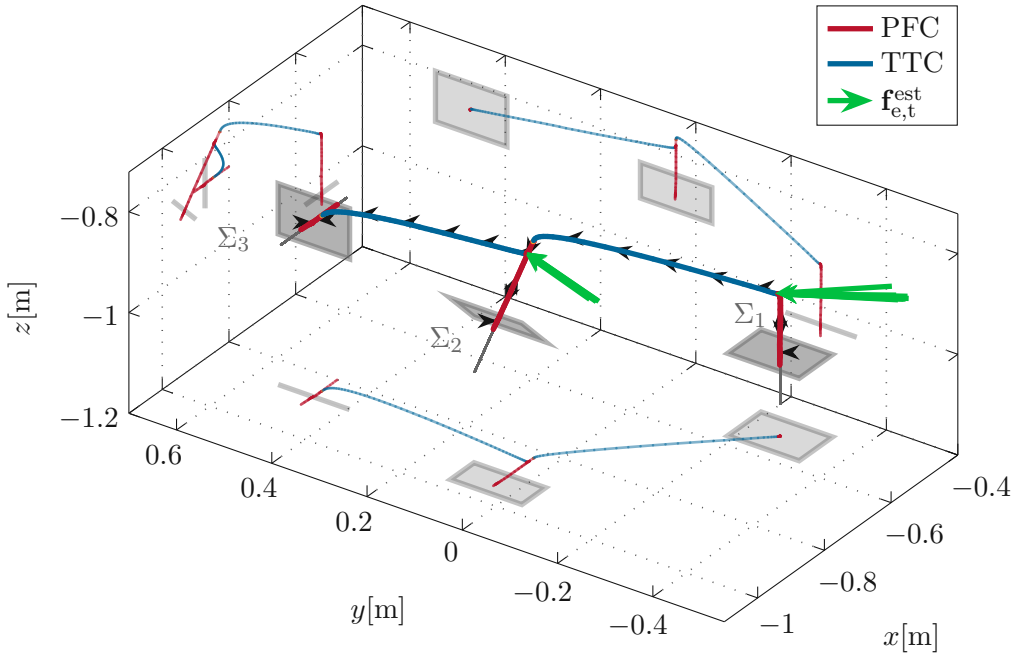


Figure 5.9: Trajectory of the TCP $\mathbf{y}_{\mathrm{t}}$ in the workspace during the task execution with the Path Switch mode, where the color represents the currently acting controller (PFC or TTC). The externally applied force $\mathbf{f}_{\mathrm{e,t}}^{\mathrm{est}}$ is depicted as green arrows.

the desired path $\Sigma_{i'}$ lies. The paths Σ_1, Σ_2 and Σ_3 are not visible to the human and the operator can only estimate the correct direction for the push. To help the human operator, the search angle ϕ is chosen rather large at a value of $\phi = 45\,°$. This eases the initiation of a transition. As the paths are rather distinct in the presented drilling scenario, the value of $\phi = 45\,°$ is still reasonable. In Figure 5.9, it is noticeable that for the transition from Σ_2 to Σ_3, the vectors of the external force $\mathbf{f}_{\mathrm{e,t}}^{\mathrm{est}}$ do not exactly point in the same direction as the TCP $\mathbf{y}_{\mathrm{t}}$ is moving during the transition. Referring to the minimization problem (4.30), $\bar{\theta}_{i'}^{*}$ and thus the position $\boldsymbol{\sigma}_{\mathrm{t},i'}\left(\bar{\theta}_{i'}^{*}\right)$ where the transition trajectory $\boldsymbol{\kappa}_{\mathrm{t}}(t)$ ends, is selected as the closest point on the new path that lies in a cone with the angular opening ϕ. With increasing values for the angular opening ϕ, larger deviations from the push direction $\mathbf{e}_{\mathrm{f}}$ are allowed. This is visible at the transition from Σ_2 to Σ_3.

To analyze the evolution of the state of the system over time, Figure 5.10 depicts the beliefs b_1, b_2 and b_3 for the individual drilling paths Σ_1, Σ_2 and Σ_3, the translational part of the external force estimation $\mathbf{f}_{\mathrm{e,t}}^{\mathrm{est}}$ in x-, y- and z- direction ($f_{\mathrm{e,t,x}}^{\mathrm{est}}$, $f_{\mathrm{e,t,y}}^{\mathrm{est}}$ and $f_{\mathrm{e,t,z}}^{\mathrm{est}}$, respectively) and the tangential path-based coordinate $\xi_{\parallel,i_{\mathrm{act}}}$ and its derivative $\dot{\xi}_{\parallel,i_{\mathrm{act}}}$.

The gray areas indicate a transition controlled by the TTC. During the transition, the beliefs b_1, b_2 and b_3 as well as $\xi_{\parallel,i}$ and $\dot{\xi}_{\parallel,i}, i \in \{1,2,3\}$ do not exist or have no significant meaning, so these signals are not included in the plot. The path evolution represented by $\xi_{\parallel,i_{\mathrm{act}}}$ and $\dot{\xi}_{\parallel,i_{\mathrm{act}}}$, when PFC is active, shows the analogous drilling pattern compared to Figure 5.7. This is not surprising, since the executed task is the same.

The transition behavior between the paths is discussed next. Figure 5.10 shows that

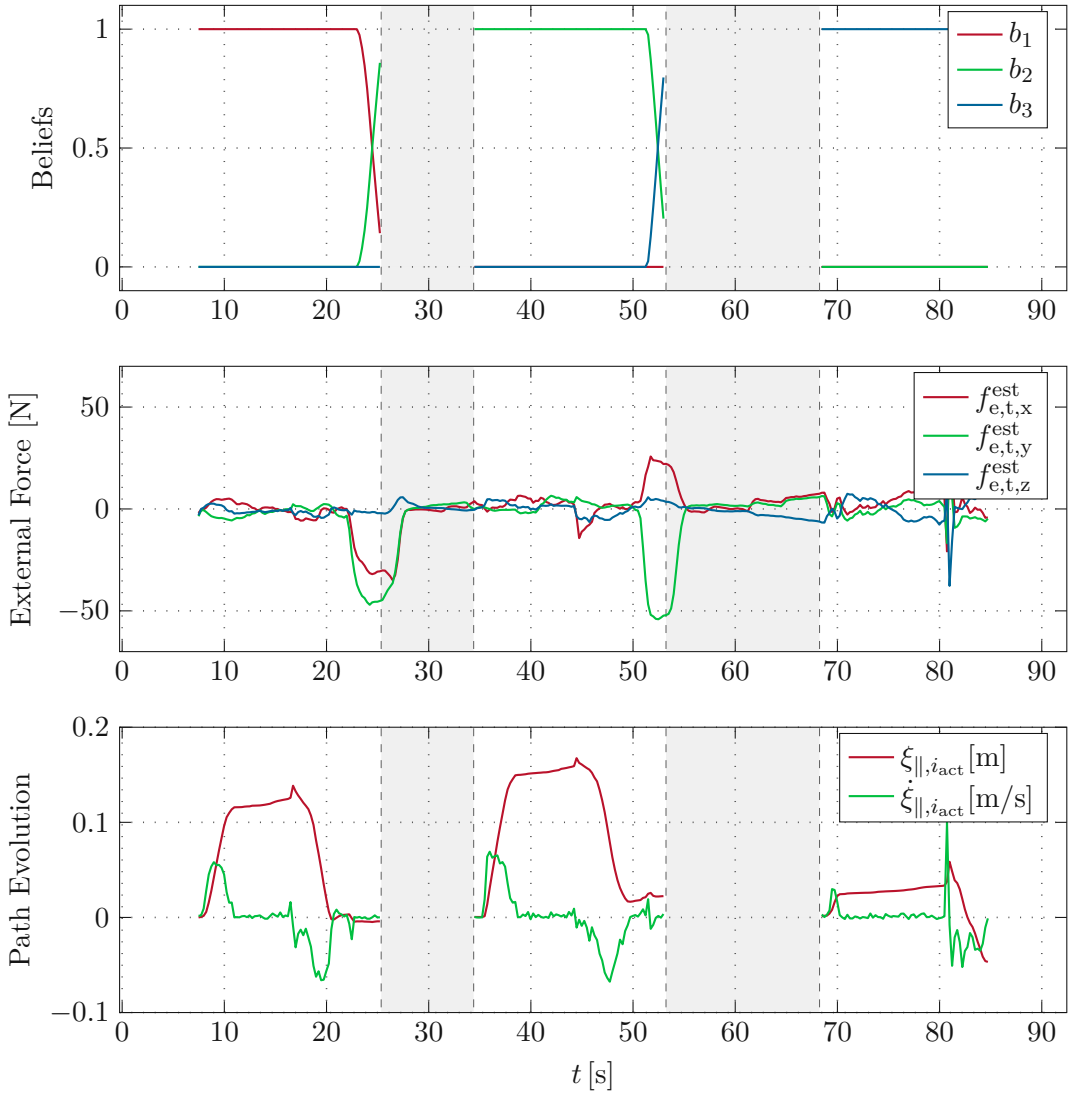


Figure 5.10: Beliefs of each path b_i, $i \in \{1, 2, 3\}$ with the external force applied $\mathbf{f}^{\text{est}}_{\text{e,t}}$ and the evolution along the path. White background means the PFC is active and gray background means the TTC is active.

before each transition block (i. e. at $t = 22\,\text{s}$ and $t = 51\,\text{s}$), the magnitude of the external force $\mathbf{f}^{\text{est}}_{\text{e,t}}$ increases. The belief system checks the applied forces and calculates the values of the beliefs b_i, $i \in \{1, 2, 3\}$. After about 2-3 s (i. e. at $t = 25\,\text{s}$ and $t = 53\,\text{s}$, respectively), the new belief $b_{i'}$ (where $i' = 2$ at $t = 25\,\text{s}$ and $i' = 3$ at $t = 53\,\text{s}$, respectively) has risen to $b_{i'} = 0.8$, causing the initiation of the transition to $\Sigma_{i'}$. At the same time, the operator does not note the initiation of the transition right away and keeps pushing the robot for another 2-3 s (i. e. until $t = 27\,\text{s}$ and $t = 55\,\text{s}$, respectively). Then, the human operator notices that the robot started to move towards the desired path $\Sigma_{i'}$. At this time, the human operator starts to remove the applied force and $\mathbf{f}^{\text{est}}_{\text{e,t}}$ decreases. This matches the

assumptions taken in Section 4.2.2 that the transition-initiating force is still present at the beginning of the transition and is released towards the end.

Figure 5.11 shows the norm of the transversal error of the PFC $\tilde{\xi}_{i_{\text{act}}} = \sqrt{\xi^2_{\perp,i_{\text{act}}} + \xi^2_{\pitchfork,i_{\text{act}}}}$, the norm of the translational error of the TTC $\tilde{y}_{\text{t}} = \left\| \mathbf{y}_{\text{t}} - \mathbf{y}^{\text{d}}_{\text{t}} \right\|_2$, the norm of the velocity error of the PFC in transversal direction $\dot{\tilde{\xi}}_{i_{\text{act}}} = \sqrt{\dot{\xi}^2_{\perp,i_{\text{act}}} + \dot{\xi}^2_{\pitchfork,i_{\text{act}}}}$, the norm of the velocity error of the TTC $\dot{\tilde{y}}_{\text{t}} = \left\| \dot{\mathbf{y}}_{\text{t}} - \dot{\mathbf{y}}^{\text{d}}_{\text{t}} \right\|_2$, the norm of the vector part of the quaternion error $\left\| \boldsymbol{\varepsilon}^{\mathcal{D}_{i_{\text{act}}}}_{\mathcal{T}} \right\|_2$ and $\left\| \boldsymbol{\varepsilon}^{\mathcal{K}}_{\mathcal{T}} \right\|_2$, respectively, and the norm of the angular velocity error $\tilde{\omega}_{i_{\text{act}}} = \left\| \boldsymbol{\omega}^{\mathcal{T}} - \boldsymbol{\omega}^{\mathcal{D}_{i_{\text{act}}}} \right\|_2$ and $\tilde{\omega} = \left\| \boldsymbol{\omega}^{\mathcal{T}} - \boldsymbol{\omega}^{\mathcal{K}} \right\|_2$, respectively, during the task execution.

For the translational part of the errors, it makes a difference, if the PFC or the TTC is applied. As long as the PFC is applied (white background in Figure 5.11), $\tilde{\xi}_{i_{\text{act}}}$ and $\dot{\tilde{\xi}}_{i_{\text{act}}}$ are plotted and when the TTC is applied (gray background in Figure 5.11), $\tilde{y}_{\text{t}}$ and $\dot{\tilde{y}}_{\text{t}}$ are plotted. For the rotational part of the errors, no distinction is needed.

The positional error during a single drilling process reaches a value of maximum $\tilde{\xi}_{i_{\text{act}}} = 3.8\,\text{mm}$. For the initiation of a transition, a maximum error of $\tilde{\xi}_{i_{\text{act}}} = 3.5\,\text{mm}$ is measured. Although both mentioned values are in the same range, the belief system does not react to the deviations caused by the drilling process, cf. Figure 5.10. The interaction forces caused by the drilling force fluctuate more and at higher frequencies than the constantly held exernal human force $\mathbf{f}_{\text{e,t}}$, which is applied to start a transition. Therefore, the external force estimation $\mathbf{f}^{\text{est}}_{\text{e,t}}$ depicted in Figure 5.10 only reacts to the human applied force. During the drilling process, a maximum rotational error of $\left\| \boldsymbol{\varepsilon}^{\mathcal{D}_{i_{\text{act}}}}_{\mathcal{T}} \right\|_2 = 0.008$ or $\phi^{\mathcal{D}_{i_{\text{act}}}}_{\mathcal{T}} = 0.9°$ is measured. At the start of a transition, a maximum rotational error of $\left\| \boldsymbol{\varepsilon}^{\mathcal{D}_{i_{\text{act}}}}_{\mathcal{T}} \right\|_2 = 0.01$ or $\phi^{\mathcal{D}_{i_{\text{act}}}}_{\mathcal{T}} = 1.1°$ is found. These values for the transversal and rotational errors are close to the values for the errors during the experiment with the Path Snap-In in Section 5.2. Since the compliance parameters chosen for the experiment with the Path Switch are about twice as large as the compliance parameters chosen for the experiment with the Path Snap-In, a smaller error in the experiment with the Path Switch is expected. This is not visible in the measured data because the errors during the experiment with the Path Switch and the errors during the experiment with the Path Snap-In are both about $\tilde{\xi}_{i_{\text{act}}} \approx \tilde{\xi}_i \approx 3.5\,\text{mm} - 5\,\text{mm}$ and $\phi^{\mathcal{D}_{i_{\text{act}}}}_{\mathcal{T}} \approx \phi^{\mathcal{D}_i}_{\mathcal{T}} \approx 0.9° - 1.1°$ during the drilling process. A possible explanation is that the human-robot collaboration and interaction is not a task with high repeatability. The behavior of the human changes from experiment to experiment. The same errors in both experiments may result from different applied external forces. For a more sophisticated comparsion, a F/T sensor is required to be attached to the endeffector. In this way, the actual force applied by the human is measured exactly. Only the combination of the known exciting force and measured errors $\tilde{\xi}_{i_{\text{act}}}$, $\tilde{\xi}_i$ and $\left\| \boldsymbol{\varepsilon}^{\mathcal{D}_{i_{\text{act}}}}_{\mathcal{T}} \right\|_2$, $\left\| \boldsymbol{\varepsilon}^{\mathcal{D}_i}_{\mathcal{T}} \right\|_2$ or $\phi^{\mathcal{D}_{i_{\text{act}}}}_{\mathcal{T}}$, $\phi^{\mathcal{D}_i}_{\mathcal{T}}$ enables a complete analysis of the resulting behavior of the system.

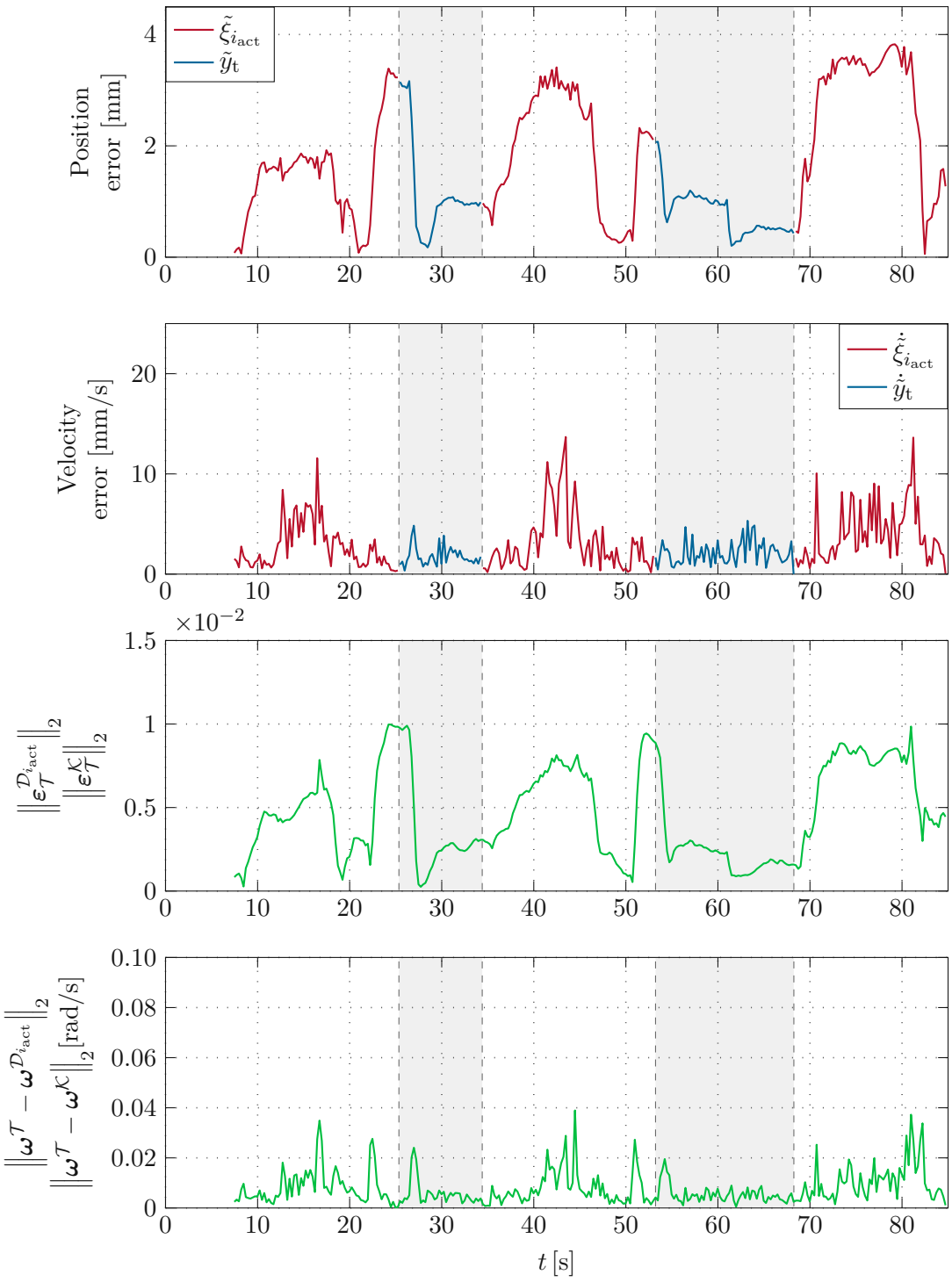


Figure 5.11: Translational position and velocity errrors as well as rotation and angular velocity error for the PFC and TTC during the execution of the drilling task with the Path Switch mode.

6 Conclusion and Outlook

In this work, three novel methods for human-robot interaction (HRI) are introduced, which are called *Path Snap-In*, *Path Switch* and *Orientation Snap*. These methods can either be used for the teach-in of a robotic task or during the collaborative execution of a robotic task. The findings of the work are validated in an exemplary drilling scenario. In this scenario, the robot carries the load of a cordless drill and stabilizes it on the desired drilling paths, while the human operates the drill and guides it through the drilling process. In this experiment, three drilling paths are defined between which the human operator moves the robot in arbitrary sequence.

In Chapter 2, the mathematical model of the robot is derived. The used notation for the remainder of the work is explained and the coordinate frames are introduced.

The HRI modes are based on the concept of path following control (PFC), which is explained in Chapter 3. First, a definition of parametrized paths and the distinction between PFC and trajectory tracking control (TTC) is given. To make use of the path representation, a local coordinate frame is introduced. In order to define this local frame, the optimal position along the path must be found, which is formulated as a minimization problem for the path parameter. After obtaining the path-based coordinates, an exact input/output feedback linearization transforms the nonlinear system dynamics to a linear behavior from the new virtual input to the desired control variables. Path-based impedance control is introduced, with which the TCP behaves like a spring-mass-damper system with user-defined dynamics in the path-based frame. Finally, a nullspace controller is designed to stabilize the nullspace.

Based on the PFC, the novel HRI modes are introduced in Chapter 4. The Path Snap-In is assisting the operator in the vicinity of a given path and has no influence distant to the paths. In the Path Snap-In mode, the robot TCP is attracted to and snaps into a path in close vincinity. The human operator can release the locked-in state by manually pushing the TCP away from the path. For a robotic teach-in process, a special case of the Path Snap-In mode, called Orientation Snap, is used. The Orientation Snap attracts and stabilizes the TCP on predefined orientations, once the human operator turns the TCP to a certain proximity of a predefined orientation. Additionally, the Path Snap-In can be used in a collaborative task execution. For the control concept, the virtual inputs of each individual PFC are combined in a weighted sum with a distance-dependent weighting factor. In contrast, the Path Switch mode always stabilizes the TCP on an active path. It estimates the external force provided by the human operator and evaluates the force direction and magnitude w.r.t. other paths known to the system. This way, the system updates its belief about which path is intended by the operator and should be selected. To transition between two paths, a transition trajectory is planned online and traversed using a TTC, bringing the TCP to the new path intended by the human.

The introduced concepts are validated in a laboratory experiment in Chapter 5, in

which three drilling paths are programmed with the support of the Orientation Snap. The collaborative execution of the drilling task is demonstrated with the Path Snap-In mode and the Path Switch mode.

The experiments demonstrate the working principle of the novel interaction modes, but there are still some open topics to be further analyzed. In future work, a more in-depth analysis of the impedance parameters and the interaction thresholds may be done. The parameter choice strongly depends on human preference, and there is no single choice that fits for all. Some operators may like a more responsive behavior of the robot while it may intimidate others. Additionally, during the design of the drilling experiment, the limited range of the used KUKA robot has been noticed. The control allows the movement into any possible pose and the cordless drill mounted to the endeffector has a large distance between the TCP and the robot wrist. Consequently, the robot sometimes ends up in joint limitations. Thus, in a future setup, the robot arm should be mounted on a 2-dimensional gantry system, which increases the range of motion and provides more flexibility to the human-robot interactions. Regarding the Path Switch mode, the currently used trajectory generator does not take possible obstacles into account. In a real working environment, a more sophisticated trajectory generator is needed. The design of such a trajectory generator may be included in future work.

Overall, the concepts introduced in this work open up possibilities for a more intuitive programming as well as collaborative execution of robotic tasks. This makes robotic systems more interesting for small- to medium-sized enterprises because it decreases the need for highly skilled robot programmers and significantly increases flexibility and productivity. The collaborative task execution combines the accuracy and the physical support of the robot with the reasoning of the human, providing an accurate and smart solution at the same time.

A Appendix

A.1 Proofs

A.1.1 Pseudoinverse of path-based Jacobian

In the following, the pseudoinverse of the path-based Jacobian is discussed. The pseudoinverse of the path-based Jacobian $\hat{\mathbf{J}}^\dagger = \hat{\mathbf{J}}^\mathrm{T}\left(\hat{\mathbf{J}}\hat{\mathbf{J}}^\mathrm{T}\right)^{-1}$ is a *right* pseudoinverse, such that $\hat{\mathbf{J}}\hat{\mathbf{J}}^\dagger = \mathbf{I}$, but $\hat{\mathbf{J}}^\dagger\hat{\mathbf{J}} \neq \mathbf{I}$. With the given path-based Jacobian $\hat{\mathbf{J}}$, see (3.27) and (3.28)

$$\hat{\mathbf{J}} = \begin{bmatrix} \beta\mathbf{e}_\parallel^\mathrm{T} & \mathbf{0}_{1\times3} \\ \mathbf{e}_\perp^\mathrm{T} & \mathbf{0}_{1\times3} \\ \mathbf{e}_\pitchfork^\mathrm{T} & \mathbf{0}_{1\times3} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{J} \,, \tag{A.1}$$

it can be shown that

$$\hat{\mathbf{J}}^\dagger = \mathbf{J}^\dagger \begin{bmatrix} \frac{1}{\beta}\mathbf{e}_\parallel & \mathbf{e}_\perp & \mathbf{e}_\pitchfork & \mathbf{0} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I} \end{bmatrix} \tag{A.2}$$

holds. Multiplying (A.1) with (A.2), gives

$$\begin{aligned} \hat{\mathbf{J}}\hat{\mathbf{J}}^\dagger &= \begin{bmatrix} \beta\mathbf{e}_\parallel^\mathrm{T} & \mathbf{0}_{1\times3} \\ \mathbf{e}_\perp^\mathrm{T} & \mathbf{0}_{1\times3} \\ \mathbf{e}_\pitchfork^\mathrm{T} & \mathbf{0}_{1\times3} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \underbrace{\mathbf{J}\mathbf{J}^\dagger}_{\mathbf{I}} \begin{bmatrix} \frac{1}{\beta}\mathbf{e}_\parallel & \mathbf{e}_\perp & \mathbf{e}_\pitchfork & \mathbf{0} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \beta\mathbf{e}_\parallel^\mathrm{T} & \mathbf{0}_{1\times3} \\ \mathbf{e}_\perp^\mathrm{T} & \mathbf{0}_{1\times3} \\ \mathbf{e}_\pitchfork^\mathrm{T} & \mathbf{0}_{1\times3} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \frac{1}{\beta}\mathbf{e}_\parallel & \mathbf{e}_\perp & \mathbf{e}_\pitchfork & \mathbf{0} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{e}_\parallel^\mathrm{T}\mathbf{e}_\parallel & \beta\mathbf{e}_\parallel^\mathrm{T}\mathbf{e}_\perp & \beta\mathbf{e}_\parallel^\mathrm{T}\mathbf{e}_\pitchfork & \mathbf{0}_{1\times3} \\ \frac{1}{\beta}\mathbf{e}_\perp^\mathrm{T}\mathbf{e}_\parallel & \mathbf{e}_\perp^\mathrm{T}\mathbf{e}_\perp & \mathbf{e}_\perp^\mathrm{T}\mathbf{e}_\pitchfork & \mathbf{0}_{1\times3} \\ \frac{1}{\beta}\mathbf{e}_\pitchfork^\mathrm{T}\mathbf{e}_\parallel & \mathbf{e}_\pitchfork^\mathrm{T}\mathbf{e}_\perp & \mathbf{e}_\pitchfork^\mathrm{T}\mathbf{e}_\pitchfork & \mathbf{0}_{1\times3} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & \mathbf{0}_{1\times3} \\ 0 & 1 & 0 & \mathbf{0}_{1\times3} \\ 0 & 0 & 1 & \mathbf{0}_{1\times3} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{I} \end{bmatrix} = \mathbf{I} \,, \end{aligned} \tag{A.3}$$

which shows that (A.2) is indeed the pseudoinverse.

A.1.2 Transformation from path-based velocity to cartesian velocity

Analogous to the explanations in Appendix A.1.1, the equation

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix} \beta \mathbf{e}_{\parallel}^{\mathrm{T}} \\ \mathbf{e}_{\perp}^{\mathrm{T}} \\ \mathbf{e}_{\pitchfork}^{\mathrm{T}} \end{bmatrix} \dot{\mathbf{y}}_{\mathrm{t}} \ , \tag{A.4}$$

given in (3.27), is solved for $\dot{\mathbf{y}}_{\mathrm{t}}$, yielding

$$\dot{\mathbf{y}}_{\mathrm{t}} = \begin{bmatrix} \frac{1}{\beta}\mathbf{e}_{\parallel} & \mathbf{e}_{\perp} & \mathbf{e}_{\pitchfork} \end{bmatrix} \dot{\boldsymbol{\xi}} \ . \tag{A.5}$$

To prove (A.5), reinserting (A.5) in (A.4) gives

$$\begin{aligned} \dot{\boldsymbol{\xi}} &= \begin{bmatrix} \beta \mathbf{e}_{\parallel}^{\mathrm{T}} \\ \mathbf{e}_{\perp}^{\mathrm{T}} \\ \mathbf{e}_{\pitchfork}^{\mathrm{T}} \end{bmatrix} \dot{\mathbf{y}}_{\mathrm{t}} \\ &= \begin{bmatrix} \beta \mathbf{e}_{\parallel}^{\mathrm{T}} \\ \mathbf{e}_{\perp}^{\mathrm{T}} \\ \mathbf{e}_{\pitchfork}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \frac{1}{\beta}\mathbf{e}_{\parallel} & \mathbf{e}_{\perp} & \mathbf{e}_{\pitchfork} \end{bmatrix} \dot{\boldsymbol{\xi}} \\ &= \begin{bmatrix} \mathbf{e}_{\parallel}^{\mathrm{T}}\mathbf{e}_{\parallel} & \beta \mathbf{e}_{\parallel}^{\mathrm{T}}\mathbf{e}_{\perp} & \beta \mathbf{e}_{\parallel}^{\mathrm{T}}\mathbf{e}_{\pitchfork} \\ \frac{1}{\beta}\mathbf{e}_{\perp}^{\mathrm{T}}\mathbf{e}_{\parallel} & \mathbf{e}_{\perp}^{\mathrm{T}}\mathbf{e}_{\perp} & \mathbf{e}_{\perp}^{\mathrm{T}}\mathbf{e}_{\pitchfork} \\ \frac{1}{\beta}\mathbf{e}_{\pitchfork}^{\mathrm{T}}\mathbf{e}_{\parallel} & \mathbf{e}_{\pitchfork}^{\mathrm{T}}\mathbf{e}_{\perp} & \mathbf{e}_{\pitchfork}^{\mathrm{T}}\mathbf{e}_{\pitchfork} \end{bmatrix} \dot{\boldsymbol{\xi}} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \dot{\boldsymbol{\xi}} = \dot{\boldsymbol{\xi}} \ . \end{aligned} \tag{A.6}$$

This shows that (A.5) holds.

A.2 Parameters

A.2.1 System Parameters

i	$d_{i,\mathrm{y}}$ [mm]	$d_{i,\mathrm{z}}$ [mm]	α_i [rad]
1	0	152.5	0
2	0	207.5	$-\frac{\pi}{2}$
3	−232.5	0	$\frac{\pi}{2}$
4	0	187.5	$\frac{\pi}{2}$
5	212.5	0	$-\frac{\pi}{2}$
6	0	187.5	$-\frac{\pi}{2}$
7	−79.6	0	$\frac{\pi}{2}$
8	–	72.4	–

Table A.1: Kinematic parameters of the robot KUKA LBR iiwa 14 R820.

Mass	m [kg]	1.96
Center of Gravity (cog)	$d_{\mathrm{cog,x}}$ [mm]	54
	$d_{\mathrm{cog,y}}$ [mm]	0
	$d_{\mathrm{cog,z}}$ [mm]	119
Origin position $\mathbf{d}_{\mathcal{E}}^{\mathcal{T}}$ of the TCP $\mathcal{T}$ w.r.t. the endeffector frame $\mathcal{E}$	$d_{\mathcal{E},\mathrm{x}}^{\mathcal{T}}$ [mm]	0
	$d_{\mathcal{E},\mathrm{y}}^{\mathcal{T}}$ [mm]	0
	$d_{\mathcal{E},\mathrm{z}}^{\mathcal{T}}$ [mm]	130

Table A.2: Parameters of the cordless drill MAKITA DHP446 used as the tool for the experiments.

A.2.2 Controller Parameters

Symbol	Value	Unit
$\mathbf{M}^{\mathrm{d}}$	$\mathrm{diag}([0,0,0,1,1,1])$	kg
$\mathbf{D}^{\mathrm{d}}$	$\mathrm{diag}([0,0,0,30,30,30])$	N s/m
$\mathbf{K}^{\mathrm{d}}$	$\mathrm{diag}([0,0,0,200,200,200])$	N/m
$d_{0,\mathrm{t}}$	0.04	m
c_{t}	2.5	
$d_{0,\mathrm{r}}$	0.1	m
c_{r}	2.5	
$\mathbf{D}_{\mathrm{dam,t}}$	$\mathrm{diag}([10,10,10])$	N s/m
$c_{\mathrm{dam,t}}$	16	
$\mathbf{D}_{\mathrm{dam,r}}$	$\mathrm{diag}([10,10,10])$	N s/m
$c_{\mathrm{dam,r}}$	16	
$\mathbf{K}_{\mathrm{n}}$	$0.1 \cdot \mathbf{I}_7$	N/m
k_{n}	10	N/m
$\mathbf{D}_{\mathrm{n}}$	$40 \cdot \mathbf{I}_7$	N s/m
Δq_i	$0.2\,(q_{\max,i} - q_{\min,i})$	rad
N_φ	4	
N_ϑ	5	
N_ψ	4	

Table A.3: Controller parameters for the Teach-in experiment.

Symbol	Value	Unit
$\mathbf{M}^{\mathrm{d}}$	$\mathrm{diag}([1,1,1,1,1,1])$	kg
$\mathbf{D}^{\mathrm{d}}$	$\mathrm{diag}([10,60,60,60,60,60])$	N s/m
$\mathbf{K}^{\mathrm{d}}$	$\mathrm{diag}([0,500,500,500,500,500])$	N/m
$\xi_{\parallel}^{\mathrm{d}}$	0 (arbitrary, since $k_{\parallel}^{\mathrm{d}} = 0$)	m
$\dot{\xi}_{\parallel}^{\mathrm{d}}$	0	m/s
$\ddot{\xi}_{\parallel}^{\mathrm{d}}$	0	m/s^2
$d_{0,\mathrm{t}}$	0.04	m
c_{t}	2.5	
$d_{0,\mathrm{r}}$	0.1	m
c_{r}	2.5	
$\mathbf{D}_{\mathrm{dam,t}}$	$\mathrm{diag}([10,10,10])$	N s/m
$c_{\mathrm{dam,t}}$	16	
$\mathbf{D}_{\mathrm{dam,r}}$	$\mathrm{diag}([10,10,10])$	N s/m
$c_{\mathrm{dam,r}}$	16	
$\mathbf{K}_{\mathrm{n}}$	$0.1 \cdot \mathbf{I}_7$	N/m
k_{n}	10	N/m
$\mathbf{D}_{\mathrm{n}}$	$40 \cdot \mathbf{I}_7$	N s/m
Δq_i	$0.2\,(q_{\max,i} - q_{\min,i})$	rad

Table A.4: Controller parameters for the Path Snap-in experiment.

Symbol	Value	Unit
$\mathbf{M}^{\mathrm{d}}$	$\mathrm{diag}([1,1,1,1,1,1])$	kg
$\mathbf{D}^{\mathrm{d}}$	$\mathrm{diag}([5,100,100,100,100,100])$	N s/m
$\mathbf{K}^{\mathrm{d}}$	$\mathrm{diag}([0,1000,1000,1000,1000,1000])$	N/m
$\mathbf{M}_{\kappa}^{\mathrm{d}}$	$\mathrm{diag}([1,1,1,1,1,1])$	kg
$\mathbf{D}_{\kappa}^{\mathrm{d}}$	$\mathrm{diag}([50,50,50,50,50,50])$	N s/m
$\mathbf{K}_{\kappa}^{\mathrm{d}}$	$\mathrm{diag}([1000,1000,1000,1000,1000,1000])$	N/m
$\xi_{\parallel}^{\mathrm{d}}$	0 (arbitrary, since $k_{\parallel}^{\mathrm{d}} = 0$)	m
$\dot{\xi}_{\parallel}^{\mathrm{d}}$	0	m/s
$\ddot{\xi}_{\parallel}^{\mathrm{d}}$	0	m/s^2
ε_{d}	1	m^{-1} s^{-1}
ε_{f}	4	N^{-1} s^{-1}
b_{th}	0.8	
$f_{\mathrm{h,t}}$	80	N
ϕ	45	°
c_{t}	20	s/m
c_{r}	20	s
$\mathbf{K}_{\mathrm{n}}$	$0.01 \cdot \mathbf{I}_7$	N/m
k_{n}	40	N/m
$\mathbf{D}_{\mathrm{n}}$	$30 \cdot \mathbf{I}_7$	N s/m
Δq_i	$0.2\,(q_{\max,i} - q_{\min,i})$	rad

Table A.5: Controller parameters for the Path Switch experiment.

A.3 Algorithms

A.3.1 Winner Take All Algorithm

Algorithm 1: Winner Take All Algorithm $\dot{\mathbf{b}} = \Omega\left(\mathbf{b}, \dot{\hat{\mathbf{b}}}\right)$

Input: $\mathbf{b}^{\mathrm{T}} = \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix}$, current belief vector

Input: $\dot{\hat{\mathbf{b}}}^{\mathrm{T}} = \begin{bmatrix} \dot{\hat{b}}_1 & \dot{\hat{b}}_2 & \dots & \dot{\hat{b}}_n \end{bmatrix}$, vector of desired belief updates

Output: $\dot{\mathbf{b}}^{\mathrm{T}} = \begin{bmatrix} \dot{b}_1 & \dot{b}_2 & \cdots & \dot{b}_n \end{bmatrix}$, vector of computed belief updates that keeps the belief system consistent

/* Find the index w of largest belief increase $\dot{\hat{b}}_i$ */

$w = \underset{i \in \{1,\dots,n\}}{\arg\max}\, \dot{\hat{b}}_i$

/* If the belief at the largest belief increase b_w is already 1, nothing should change */

if $b_w = 1$ **then**
  $\dot{\mathbf{b}} \leftarrow \mathbf{0}$
  return $\dot{\mathbf{b}}$
end

/* Otherwise, find the index v of second largest belief increase $\dot{\hat{b}}_i$ */

$v = \underset{i \in \{1,\dots,n\} \setminus w}{\arg\max}\, \dot{\hat{b}}_i$

for $i \in \{1, \dots, n\}$ **do**
  /* Subtract the mean value of the two largest belief increases, such that $\dot{\hat{b}}_w \geq 0$ and $\dot{\hat{b}}_v \leq 0$ */
  $\dot{b}_i \leftarrow \dot{\hat{b}}_i - (\dot{\hat{b}}_w + \dot{\hat{b}}_v)/2$
  /* Do not decrease beliefs, that are already zero */
  if $\dot{b}_i < 0$ and $b_i = 0$ **then**
    $\dot{b}_i \leftarrow 0$
  end
end

/* Ensure that sum of belief changes is zero */

$\dot{b}_w \leftarrow \dot{b}_w - \sum_{i \in \{1,\dots,n\}} \dot{b}_i$

return $\dot{\mathbf{b}}$

A.4 Drawings

Figure A.1: Assembly drawing of the holder for the cordless drill - side.

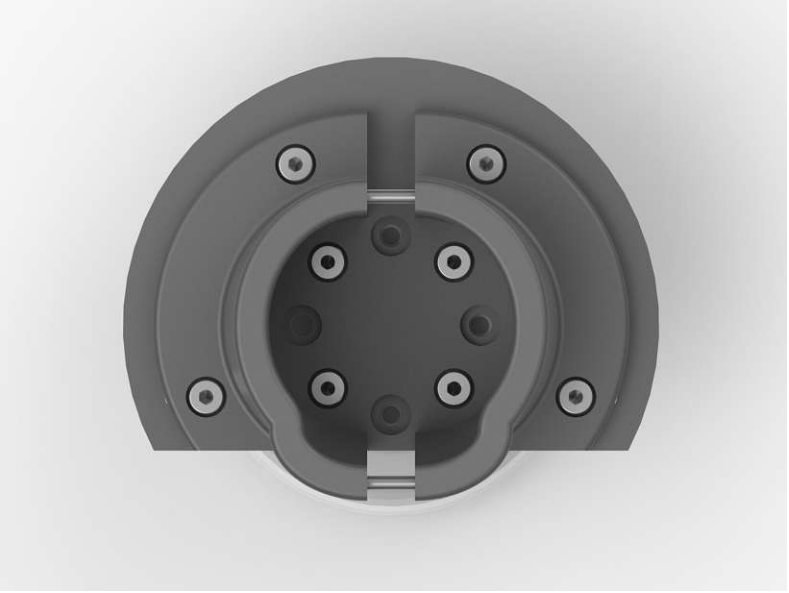


Figure A.2: Assembly drawing of the holder for the cordless drill - front.

Figure A.3: Assembly drawing of the holder for the cordless drill - top.

Bibliography

[1] International Federation of Robotics, "Executive summary world robotics 2021 industrial robots," 2021. [Online]. Available: https://ifr.org/img/worldrobotics/Executive_Summary_WR_Industrial_Robots_2021.pdf (visited on Jun. 2, 2022).

[2] L. Probst, L. Frideres, B. Pedersen, and C. Caputi, "Service innovation for smart industry: Human–robot collaboration," 2015. [Online]. Available: https://ec.europa.eu/docsroom/documents/13392/attachments/4/translations/en/renditions/native (visited on Jun. 2, 2022).

[3] J. E. Colgate, J. Edward, M. A. Peshkin, and W. Wannasuphoprasit, "Cobots: Robots for collaboration with human operators," in *Proceedings of the ASME International Mechanical Engineering Congress and Exposition*, 1996, pp. 433–439.

[4] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018. [Online]. Available: https://doi.org/10.1016/j.mechatronics.2018.02.009 (visited on Jun. 2, 2022).

[5] T. Ende, S. Haddadin, S. Parusel, T. Wüsthoff, M. Hassenzahl, and A. Albu-Schäffer, "A human-centered approach to robot gesture based communication within collaborative working processes," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3367–3374. [Online]. Available: https://doi.org/10.1109/IROS.2011.6094592 (visited on Jul. 4, 2022).

[6] A. Pettersson, T. Ohlsson, S. Davis, J. O. Gray, and T. J. Dodd, "A hygienically designed force gripper for flexible handling of variable and easily damaged natural food products," *Innovative Food Science & Emerging Technologies*, vol. 12, no. 3, pp. 344–351, 2011. [Online]. Available: https://doi.org/10.1016/j.ifset.2011.03.002 (visited on Jun. 7, 2022).

[7] J. Krüger, R. Bernhardt, D. Surdilovic, and G. Spur, "Intelligent Assist Systems for Flexible Assembly," *CIRP Annals*, vol. 55, no. 1, pp. 29–32, 2006. [Online]. Available: https://doi.org/10.1016/S0007-8506(07)60359-X (visited on Jun. 7, 2022).

[8] A. Hentout, M. Aouache, A. Maoudj, and I. Akli, "Human–robot interaction in industrial collaborative robotics: A literature review of the decade 2008–2017," *Advanced Robotics*, vol. 33, no. 15-16, pp. 764–799, 2019. [Online]. Available: https://doi.org/10.1080/01691864.2019.1636714 (visited on Jun. 3, 2022).

[9] L. Peternel, T. Petric, E. Oztop, and J. Babic, "Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach," *Autonomous Robots*, vol. 36, no. 1, pp. 123–136, 2014. [Online]. Available: https://doi.org/10.1007/s10514-013-9361-0 (visited on Jun. 3, 2022).

[10] S. Chen, T. Qiu, T. Lin, L. Wu, J. Tian, W. Lv, and Y. Zhang, "Intelligent technologies for robotic welding," in *Robotic welding, intelligence and automation*, Berlin, Heidelberg: Springer, 2004, pp. 123–143. [Online]. Available: https://doi.org/10.1007/978-3-540-44415-2_8 (visited on Aug. 1, 2022).

[11] M. V. Andulkar and S. S. Chiddarwar, "Incremental approach for trajectory generation of spray painting robot," *Industrial Robot: An International Journal*, vol. 42, no. 3, pp. 228–241, 2015. [Online]. Available: https://doi.org/10.1108/IR-10-2014-0405 (visited on Jun. 3, 2022).

[12] H. N. Huynh, H. Assadi, E. Rivière-Lorphèvre, O. Verlinden, and K. Ahmadi, "Modelling the dynamics of industrial robots for milling operations," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101 852, 2020. [Online]. Available: https://doi.org/10.1016/j.rcim.2019.101852 (visited on Jun. 3, 2022).

[13] I. Iglesias, M. A. Sebastián, and J. E. Ares, "Overview of the State of Robotic Machining: Current Situation and Future Potential," *Procedia Engineering*, vol. 132, pp. 911–917, 2015. [Online]. Available: https://doi.org/10.1016/j.proeng.2015.12.577 (visited on Jun. 3, 2022).

[14] T. Faulwasser, T. Weber, P. Zometa, and R. Findeisen, "Implementation of Nonlinear Model Predictive Path-Following Control for an Industrial Robot," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1505–1511, 2017. [Online]. Available: https://doi.org/10.1109/TCST.2016.2601624 (visited on Aug. 1, 2022).

[15] C. Nielsen, C. Fulford, and M. Maggiore, "Path following using transverse feedback linearization: Application to a maglev positioning system," *Automatica*, vol. 46, no. 3, pp. 585–590, 2010. [Online]. Available: https://doi.org/10.1016/j.automatica.2010.01.009 (visited on Jun. 3, 2022).

[16] N. Hogan, "Impedance control of industrial robots," *Robotics and Computer-Integrated Manufacturing*, vol. 1, no. 1, pp. 97–113, 1984. [Online]. Available: https://doi.org/10.1016/0736-5845(84)90084-X (visited on Jun. 3, 2022).

[17] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, 2018. [Online]. Available: https://doi.org/10.1007/s10514-017-9677-2 (visited on Jun. 2, 2022).

[18] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical human–robot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2008. [Online]. Available: https://doi.org/10.1016/j.mechmachtheory.2007.03.003 (visited on Jun. 7, 2022).

[19] F. Vicentini, "Collaborative Robotics: A Survey," *Journal of Mechanical Design*, vol. 143, no. 4, 2020, 040802. [Online]. Available: https://doi.org/10.1115/1.4046238 (visited on Aug. 30, 2022).

[20] M. Khoramshahi and A. Billard, "A dynamical system approach to task-adaptation in physical human-robot interaction," *Autonomous Robots*, vol. 43, no. 4, pp. 927–946, 2019. [Online]. Available: https://doi.org/10.1007/s10514-018-9764-z (visited on Jan. 13, 2022).

[21] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013. [Online]. Available: https://doi.org/10.1162/NECO_a_00393 (visited on Aug. 1, 2022).

[22] B. Nemec, N. Likar, A. Gams, and A. Ude, "Human robot cooperation with compliance adaptation along the motion trajectory," *Autonomous Robots*, vol. 42, no. 5, pp. 1023–1035, 2018. [Online]. Available: https://doi.org/10.1007/s10514-017-9676-3 (visited on Apr. 7, 2022).

[23] R. J. Ansari and Y. Karayiannidis, "Task-Based Role Adaptation for Human-Robot Cooperative Object Handling," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3592–3598, 2021. [Online]. Available: https://doi.org/10.1109/LRA.2021.3064498 (visited on Aug. 1, 2022).

[24] D. P. Losey and M. K. O'Malley, "Trajectory Deformations From Physical Human–Robot Interaction," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 126–138, 2018. [Online]. Available: https://doi.org/10.1109/TRO.2017.2765335 (visited on Aug. 1, 2022).

[25] B. Siciliano, O. Khatib, and T. Kröger, *Springer Handbook of Robotics*. Berlin: Springer, 2008, vol. 200. [Online]. Available: https://doi.org/10.1007/978-3-540-30301-5 (visited on Aug. 1, 2022).

[26] C. Hartl-Nesic, *Surface-Based Path Following Control on Freeform 3D Objects*. Shaker Verlag, 2020.

[27] A. Albu-Schaffer, C. Ott, U. Frese, and G. Hirzinger, "Cartesian impedance control of redundant robots: Recent results with the DLR-light-weight-arms," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, 2003, 3704–3709 vol.3. [Online]. Available: https://doi.org/10.1109/ROBOT.2003.1242165 (visited on Aug. 1, 2022).

[28] C. Ott, *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, 1st ed. Berlin: Springer, 2008. (visited on Aug. 1, 2022).

[29] B. Bischof, T. Glück, and A. Kugi, "Combined Path Following and Compliance Control for Fully Actuated Rigid Body Systems in 3-D Space," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 5, pp. 1750–1760, 2017. [Online]. Available: https://doi.org/10.1109/TCST.2016.2630599 (visited on Aug. 1, 2022).

[30] C. Hartl-Nesic, B. Bischof, T. Glück, and A. Kugi, "Pfadfolgeregelung mit Konzepten für den Pfadfortschritt: Ein Assemblierungsszenario," *at - Automatisierungstechnik*, vol. 68, no. 1, pp. 44–57, 2020. [Online]. Available: https://doi.org/10.1515/auto-2019-0114 (visited on Aug. 1, 2022).

[31] R. L. Bishop, "There is More than One Way to Frame a Curve," *The American Mathematical Monthly*, vol. 82, no. 3, pp. 246–251, 1975. [Online]. Available: https://doi.org/10.2307/2319846 (visited on Jan. 18, 2022).

[32] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, "Six-DOF impedance control based on angle/axis representations," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 289–300, 1999. [Online]. Available: https://doi.org/10.1109/70.760350 (visited on Aug. 1, 2022).

[33] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot Collisions: A Survey on Detection, Isolation, and Identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017. [Online]. Available: https://doi.org/10.1109/TRO.2017.2723903 (visited on Aug. 1, 2022).

[34] T. Flash and N. Hogan, "The coordination of arm movements: An experimentally confirmed mathematical model," *Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985. [Online]. Available: https://doi.org/10.1523/JNEUROSCI.05-07-01688.1985 (visited on Jan. 19, 2022).

[35] K. Shoemake, "Animating rotation with quaternion curves," in *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, 1985, pp. 245–254. [Online]. Available: https://doi.org/10.1145/325334.325242 (visited on Jul. 4, 2022).

Eidesstattliche Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct & Regeln zur Sicherung guter wissenschaftlicher Praxis (in der aktuellen Fassung des jeweiligen Mitteilungsblattes der TU Wien), insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, im September 2022

Elias Pritzi