TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Informationstechnische Regelung

Model-based Control and Learning in Physical Human-Robot Interaction

José Ramón Medina Hernández

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Jun.-Prof. Dr. rer. nat. Martin Kleinsteuber

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Sandra Hirche

2. Prof. Dr. Aude Billard,
École Polytechnique Fédérale de Lausanne / Schweiz

Die Dissertation wurde am 15.01.2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 29.06.2015 angenommen.

Abstract

Physical cooperation between humans and robots has high potential impact in many critical application areas. Utilization domains reach from manufacturing via mobility aids to service/medical robotics comprising rehabilitation, education, training and elderly care. Among all robot control approaches for physical cooperation, anticipatory robotic assistance has demonstrated superior performance in terms of human effort minimization. However, the deployment of these control schemes for arbitrary tasks entails great difficulty. The required behavior models enabling human behavior predictions are rarely available a priori and must be retrieved as the interaction occurs. In addition, robot decisions must cope with the inherent uncertainty of predictions. Besides expected human actions, their level of uncertainty is a decisive indicator for potential prediction errors yielding undesired disagreements. This urges the need for novel methods providing intuitive assistance in such uncertain and challenging conditions.

The present thesis explores control and learning aspects of anticipatory physical assistants. We focus on four relevant open issues. First, the design of fast and online-capable behavior learning algorithms providing smooth predictions; the continuous physical coupling between agents requires non-abrupt and immediate adaptation. Second, the problem of control based on learned probabilistic/statistical predictive models where uncertainty itself is a crucial factor that should influence decisions. Third, a deeper understanding of interaction wrenches that facilitate a proper estimation of human intentions through force/torque signals. Fourth, the design of control schemes for physical assistance that seamlessly adapt to both human behavior uncertainty and disagreements.

The main contributions of this thesis address the above-mentioned issues. Regarding learning methods we introduce (i) impedance-based Gaussian Processes, which incorporate a priori human arm impedance characteristics and (ii) time-based Hidden Markov Models, which provide smooth predictions despite its discrete latent state space. To enable decision-making with uncertain models we propose a novel systematic stochastic optimal control approach where uncertainty itself is explicitly considered. In contrast to classical methods not only the expected cost but also higher order cost statistics determine the optimization criterion, increasing flexibility and robustness. We further introduce fundamental insights on the relation between interaction wrenches, load distributions and disagreements with the human. This analysis provides the means for the design of novel anticipatory control schemes adapting to both human behavior uncertainty and disagreements. Results demonstrate superior performance in terms of both implicit and subjective measures in evaluations with human users.

Contents

Notations

Abbreviations

pHRI	physical human-robot interaction
PbD	programming by demonstration
GMM	Gaussian mixture model
GMR	Gaussian mixture regression
HMM	hidden Markov model
PHMM	parametric hidden Markov model
GP	Gaussian Process
LWPR	locally-weighted projection regression
DMP	dynamical movement primitive
DoF	degree-of-freedom
PD	proportional-derivative
i.i.d	independent identically distributed
LQ	linear quadratic
LQR	linear quadratic regulator
LEQR	linear exponential quadratic regulator
iLQR	iterative linear quadratic regulator
iLQG	iterative linear quadratic Gaussian
iKCCR	iterative K cost cumulant regulator

Symbols

General

$\boldsymbol{p}$	position
$\boldsymbol{w}$	rotational velocity
$\boldsymbol{x}$	system configuration
$\boldsymbol{u}$	wrenches/system control input
A	system matrix
B	input matrix
T	time horizon
ξ	input/state variable
t	time

Spaces

$\mathbb{R}$	real line
$\mathbb{R}^n$	n-dimensional Euclidean space
$\mathbb{R}^{n\times m}$	space of $n \times m$-dimensional matrices
$\mathbb{Z}$	set of integers
$\mathbb{Z}^+$	set of non-negative integers

Operators

A^{T}	transpose of matrix A
A^{-1}	inverse of matrix A
$A \geq 0$	positive semi-definite matrix A
$A > 0$	positive definite matrix A
tr	trace
I_n	$n \times n$-dimensional identity matrix
0_n	$n \times n$-dimensional matrix with all entries 0
$\|\|\boldsymbol{x}\|\|$	Euclidean norm of vector $\boldsymbol{x}$
$\|\|\boldsymbol{x}\|\|_Q$	quadratic construct $\boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x}$

Probability Theory

Ω	abstract sample space
$\mathscr{F}$	σ-algebra on Ω
P	probability measure on $\mathscr{F}$
$\mathsf{E}[\cdot]$	expectation
$\mathsf{E}_{\mathsf{P}}[\cdot]$	expectation over probability measure P
$\mathscr{N}(\boldsymbol{x}\|\boldsymbol{\mu},\Sigma)$	Gaussian random variable defined over $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance Σ

List of Figures

List of Tables

1

Introduction

Physical cooperation between humans and robots has high impact in many critical application areas. Manufacturing is significantly facilitated by cooperative robots in object transportation. Accounting for most of the necessary effort, human physical stress is significantly reduced thereby expediting production. The impact of physical robotic partners in medical settings can not be overstated. Applications such as rehabilitation or physical training are greatly enhanced by robotic partners regulating the appropriate human effort. Service robots for domestic use also require intuitive physical interaction with human users. In this context, education and learning by means of kinesthetic teaching are promising utilization examples enabling skill transfer between humans and robots. Similarly, elderly care benefits from efficient and comfortable mobility aids from robot companions. The high impact of physical human-robot cooperation in such a broad spectrum of application domains highlights the need for intuitive and effective physical robotic partners.

Among all aspects involved in the realization of physical helpers, robot control is a crucial component; it determines robot behavior while physically coupled to a human partner. Although challenging due to human actions, a proper robot control scheme has the potential to enable seamless physical interaction between humans and robots. The first endeavors to synthesize robot behavior for physical assistance focus on compensating human force/torque signals [1]. These *passive* robots intuitively react to human inputs but do not efficiently reduce human effort. In contrast, a robot partner pursuing its own goal in a *proactive* manner is a more general, desirable and sometimes even necessary alternative [2]. As a prototypical example consider the the cooperative transport of a rigid object. A passive robot behaves accommodating for sensed wrenches through the object, thereby rendering an additional mass in the overall system. In contrast, a proactive partner do not only reacts to the human input but also pursues its own goal, a priori accounting for all required effort. It also provides increased cooperative manipulability, a requirement for fulfilling task involving navigation through narrow passages. In fact, proactive robots are the only partners capable of achieving successful cooperation, which require working jointly towards a common known goal [3].

Within proactive approaches, robot assistance based on human behavior prediction has demonstrated superior performance in terms of human effort minimization [4]. By anticipating human intentions, robot actions gather all necessary redundant effort. As a payoff, its design becomes significantly more challenging: the necessary human *behavior models* enabling predictions are rarely available a priori for arbitrary tasks. Due to the complexity of human decision-making, probabilistic/statistical data-driven approaches are a especially compelling model acquisition alternative. These machine learning and statistical methods enable the incremental retrieval of human behavior models as the interaction occurs. In fact, although a physically coupled human partner is an arduous issue for robot control, it is also an opportunity to exploit human cognitive capabilities and learn from them. However, learning during interaction for later control entails further challenges. The physical coupling between human and robot requires non-abrupt and constantly updated predictions; any discontinuities are later rendered as sudden uncomfortable wrenches. In addition, potentially unsafe predictions possibly involving risks for the human must be avoided. The design of learning algorithms providing smooth, fast and safe human behavior predictions is a challenging but necessary step towards intuitive assistance.

Based on human behavior predictions, the control task for robot assistance consists of determining the robot applied wrenches. A convenient alternative for modulating robot behavior is optimal control, where robot actions respond to the minimization of a performance criterion. Suitable optimization criteria in this context are the minimization of human effort or, similarly, following/tracking the expected human desired trajectory. However, optimal control strategies relying only on expected human actions are likely to perform poorly as they neglect valuable information encoded in prediction variance. Besides expected values, probabilistic/statistical models also provide confidence levels reflecting potential variability. This level of uncertainty is a crucial indicator of potential prediction errors that should influence robot actions. This issue is also applicable to many other robotic settings which rely on learned models. Consider as an example the problem of grasping based on uncertain object models. Some grasp configurations may seem a priori non-optimal options if only the expected cost is considered. However, if they deliver low variability they might be preferable than optimal solutions in expectation but yielding high uncertainty. This urges the need for systematic optimal decision-making methods which go beyond classical control approaches by explicitly accounting for uncertainty.

Regardless of the control method used, the synthesis of goal-oriented actions for physical assistance imply the possibility of disagreements with the human. When robot's expectations mismatch human intentions, undesired interaction forces appear incurring safety risks and discomfort. These force components result from a divergence in terms of desired accelerations, thereby representing a fundamental cue regarding human intentions. There is no consensus in the literature about how to quantify such disagreements. However, their characterization is instrumental for the design of robotic helpers that intuitively react in case of disagreement.

The present thesis studies learning and control aspects of physical assistants acquiring behavior models during interaction. We focus on four relevant open challenges related to their design:

- **Learning during interaction**. The design of fast and online-capable learning algorithms providing smooth and safe/conservative human behavior predictions.
- **Uncertainty-dependent optimal control**. The problem of optimal control based on probabilistic/statistical predictive models where uncertainty itself is a crucial factor that should influence robot actions.
- **Disagreement and load share characterization**. The identification internal/disagreement wrenches and effective load shares among agents is an instrumental step in order to interpret human intentions through force signals.
- **Assistance considering human behavior uncertainty and disagreements**. The design of control schemes for physical assistance that seamlessly adapt to both human behavior uncertainty and disagreements with the human.

Addressing these issues are key steps for the deployment of intuitive and effective physical assistants.

1.1 Outline and Contributions

This thesis aims for fundamental issues in the design of physical anticipatory robotic partners. Chapter 2 reviews related background on physical Human-Robot Interaction (pHRI), learning from demonstrations and stochastic optimal control theory. The following Chapters address the aforementioned problems of (i) behavior model learning, (ii) uncertainty-dependent control, (iii) disagreement characterization and (iv) control design for efficient anticipatory assistance. Regarding learning methods, Chapter 3 introduces novel time-series and dynamical systems models supporting incremental updates and considering a priori human behavior properties. To enable flexible decision-making with uncertain models, Chapter 4 presents a novel systematic stochastic optimal control approach where uncertainty itself is explicitly considered yielding uncertainty-dependent actions. Chapter 5 introduces fundamental insights on the relation between interaction wrenches, load distributions and disagreements with the human. In addition, novel anticipatory control schemes adapting to both model uncertainty and disagreements are presented, demonstrating superior performance in terms of both implicit and subjective measures in experiments with human users. In the following, the major contributions within each chapter are outlined in more detail.

Chapter 3: Learning Behavior Models during Interaction

This chapter explores learning methods for acquiring behavior models during interaction. Focusing on fast, safe and smooth predictions, we first present time-based Hidden Markov Models (HMMs). Besides human motion and force data, time information is also learned explicitly providing smoother predictions. Efficient regression techniques together with the application of an incremental learning framework enable learning during interaction. A user study validates the approach showing increasing prediction performance during interaction. We further extend time-based HMMs with additional hyperparameters. The resulting parametric time-based HMMs represent environmental information and natural language as latent variables that determine variations on the encoded motions. Experiments show increased generalization and abstraction capabilities.

Although time-series models provide smooth and safe predictions, their explicit time-dependency hinders the acquisition of state-dependent constraints. As an alternative, we study the applicability of dynamical systems by means of Gaussian Processes (GPs) in order to learn observed task dynamics in state space. We present a novel impedance-based GP model which incorporates the a priori impedance characteristics of the human arm. This model ensures desirable compliance properties in unvisited regions of the state space. Additionally, the human-desired trajectory is modeled as a latent variable, tracked by means of a proportional-derivative (PD) controller. Experiments reveal superior prediction performance w.r.t. naive GP models. The results of this chapter have been partly published previously in [5–7].

Chapter 4: Uncertainty-dependent Optimal Control

In this chapter we introduce a novel systematic stochastic optimal control approach yielding uncertainty-dependent actions. In order to account for uncertainty in optimal decision-making problems we study optimization criteria considering high-order cost statistics. Classical stochastic optimal control approaches consider the expected cost and thereby neglect the influence of uncertainty. In contrast, high-order cost moments or cumulants properly capture its effect on performance. The proposed approach yields a flexible and general family of decision-makers: depending on design parameters, cost variability is interpreted as an additive or discounting measure for the overall performance. We present risk-sensitive and cost-cumulant solutions for this problem for non-linear dynamics and non-quadratic costs. Locally optimal solutions are found by iteratively performing a linear quadratic approximation around a nominal trajectory, solving the local problem and updating the trajectory until convergence. To illustrate its application in different robotics settings, we study solutions for prototypical problems. Examples considering multiple uncertainty sources affecting both plant dynamics and cost parameters such as a desired goal or an obstacle are presented. Simulation results validate the benefits and the systematic applicability of the proposed approach. The results of this chapter have been partly published previously in [8].

Chapter 5: Model-based Anticipatory Control for Human-Robot Cooperative Manipulation

This chapter explores open challenges in the design of model-based anticipatory physical assistants with special emphasis on learned models. We introduce fundamental results on the characterization of internal and effective wrench components arising between goal-oriented agents. This analysis follows from understanding possibly non-uniform wrench allocation policies and load shares. Internal wrench are proven to indicate errors in terms of desired acceleration directions thereby showing its relevance for efficient anticipatory control design. Exploiting these insights and by means of the methods from Chapter 3 and Chapter 4, we introduce novel control approaches for model-based physical assistance in three different scenarios: navigation tasks, tasks comprising environmental contacts and multi-task settings.

We first present control schemes for navigation adapting to both internal wrench components and disagreements. Formulated as an optimal control problem targeting human force minimization, prediction uncertainty and expected disagreements become decisive components of robot assistance. The application of uncertainty-dependent optimal control and the adaptation to disagreements are novel concepts not addressed in any previous works concerning physical assistance. The proposed approach is objectively and subjectively evaluated in an experiment with human users. Results indicate superior performance in terms of perceived helpfulness and human effort minimization.

We further introduce a novel optimal control scheme for manipulation tasks. Depending on expected environmental wrenches, the robot motion control not only adapts its reference but also takes uncertain environmental force deviations into account. A risk-sensitive optimization yields time-varying compliance characteristics depending on uncertainty. Experiments with a robotic manipulator show increased task success in variable environmental conditions.

Envisaging robotic partners fulfilling multiple tasks in cooperation with a human, we present a dynamic strategy selection method that decides either to retain a currently selected model or switch to another one depending on recent prediction performance. In addition, a predictive model relying on observed human wrench is also introduced, becoming a suitable alternative when no available model coincides with human intentions. An experiment with human users highlights superior performance of a dynamic selection strategy when multiple models are available. The results of this Chapter have been partly published previously in [5, 9–13].

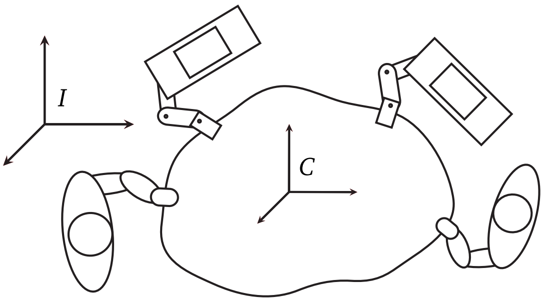
2

Background

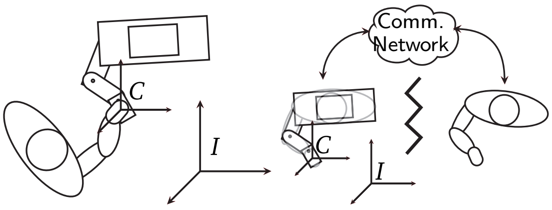
The design of physical assistants based on learned behavior models involves a wide variety of fields. This chapter provides the necessary background with special emphasis on machine learning and control aspects. An overview of pHRI covering basic control concepts and related works is first explained in Section 2.1. With regard to model acquisition methods, data-driven approaches for learning human behavior are summarized in Section 2.2. The main theoretical contribution of this thesis is the design of optimal decision-makers influenced by the inherent uncertainty of learned models. We review basic concepts of stochastic optimal control theory, detailed in Section 2.3, including discrete-time solutions of prototypical problems of interest for this thesis.

2.1 Physical Human-Robot Interaction

Physical human-robot interaction has wide and relevant application domains covering, among other areas, interactive robotics and virtual/augmented realities, medicine, domestic/service/industrial robotics and education/training. From mobility, rehabilitation or physical training assistants to occasional physical contacts with humans, *intuitive*, *safe* and *dependable* pHRI are essential requirements for artificial cognitive systems sharing their workspace with humans. Among all design aspects involved in the realization of physical helpers [14], this thesis focus on control and learning issues arising in cooperative manipulation tasks where a human and a group of robots are physically coupled by tightly grasping a rigid object. As an illustrative example and without lack of generality, in this thesis we consider the cooperative transport of an object, as shown in Fig. 2.1(a). Other scenarios, such as kinesthetic teaching Fig. 2.1(b) or teleoperation Fig. 2.1(c) are straightforwardly transformed into this setting by considering their respective kinematic constraints. This section is structured as



(a)



(b) (c)

Figure 2.1: Physical multi-agent interaction in a cooperative transportation (a), kinesthetic teaching (b) and teleoperation settings (c).

follows. In Section 2.1.1, we first define the problem of cooperative manipulation for a multi-robot system. The additional challenges arising when one of the agents is a human are then analyzed in Section 2.1.2 yielding the main challenges considered in this thesis. An overview of previous works on physical robotic assistants only reacting to human forces (*reactive* assistants) together with anticipatory goal-oriented implementations (*proactive* assistants) are presented in Section 2.1.3 and Section 2.1.4 respectively. Finally, both implicit and explicit performance measures assessing physical assistance quality with human users are explained in Section 2.1.5.

2.1.1 Cooperative Manipulation

Consider $N \in \mathbb{Z}^+$ manipulators tightly grasping a rigid object. Each manipulator is equipped with force/torque sensors mounted either on the wrist or at each joint. Let $\boldsymbol{x} \in SE(3)$ be the state of the manipulated object composed by its Cartesian position and orientation in the inertial frame I as depicted in Fig. 2.1(a). The object's twist is given by

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix} , \tag{2.1}$$

where $\dot{\boldsymbol{p}} \in \mathbb{R}^3$ is the translational velocity and $\boldsymbol{\omega} \in \mathbb{R}^3$ the rotational velocity. The dynamics of the object are given by

$$M_o \ddot{\boldsymbol{x}}(t) + \boldsymbol{h_o}(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t)) = \boldsymbol{u}(t) + \boldsymbol{u}_{\text{env}}(t) , \tag{2.2}$$

where $M_o \in \mathbb{R}^{6\times 6}$ represent the object mass and inertia and $\boldsymbol{h_o}$ is the sum of friction and gravitation,

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\tau} \end{bmatrix}, \tag{2.3}$$

is the applied wrench, where $\boldsymbol{f} \in \mathbb{R}^3$ and $\boldsymbol{\tau} \in \mathbb{R}^3$ denote the force and torque components in frame C respectively and $\boldsymbol{u}_{\text{env}}(t) \in \mathbb{R}^6$ is the wrench resulting from contacts with the environment, which is measurable separately with an additional force/torque sensor.

The wrench $\boldsymbol{u}$ results from the addition of each agent's individual wrench contribution. Let $\tilde{\boldsymbol{u}}_i$ be the wrench applied by agent i in frame C and let $\boldsymbol{p}_i \in \mathbb{R}^3$ the vector from the i-th grasp position to the object frame. The resulting wrench is given by

$$\boldsymbol{u}(t) = \sum_{i=1}^{N} G_i \tilde{\boldsymbol{u}}_i(t), \tag{2.4}$$

where G_i is the Jacobian of the kinematics constraints of agent i, usually referred to as partial grasp matrix in this context [15]. The matrix G_i takes the form

$$G_i = \begin{bmatrix} I_3 & 0_3 \\ P_i & I_3 \end{bmatrix}, \tag{2.5}$$

where $P_i = [\boldsymbol{p}_i]_\times$ is the cross-product matrix for $\boldsymbol{p}_i$ that renders the torques generated by applying forces on the grasping position.

For mathematical convenience we adopt the stacked form

$$G = \begin{bmatrix} G_1 \; G_2 \; \cdots \; G_L \end{bmatrix}$$

and

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} \tilde{\boldsymbol{u}}_1^{\mathrm{T}} \; \tilde{\boldsymbol{u}}_2^{\mathrm{T}} \; \cdots \; \tilde{\boldsymbol{u}}_L^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}},$$

such that

$$\boldsymbol{u}(t) = G\tilde{\boldsymbol{u}}(t) .$$

Given this setting, the control problem of cooperative manipulation consists of the design of the agents' exerted wrenches $\tilde{\boldsymbol{u}}$ fulfilling the following objective [16]:

Definition 2.1. The objective of a cooperative manipulation task is given by

$$\lim_{t\to\infty} \boldsymbol{x}(t) \to \boldsymbol{x}_d(t) , \tag{2.6}$$

where $\boldsymbol{x}_d(t) \in \mathbb{R}^6$ the desired object trajectory,

$$\lim_{t\to\infty} \tilde{\boldsymbol{u}}(t) \to \tilde{\boldsymbol{u}}_d(t) , \tag{2.7}$$

where $\tilde{\boldsymbol{u}}_d(t) \in \mathbb{R}^{6N}$ are the desired manipulator wrench trajectories and

$$\lim_{t\to\infty} \boldsymbol{u}_{\text{env}}(t) \to \boldsymbol{u}_{\text{env}d}(t) , \tag{2.8}$$

where $\boldsymbol{u}_{\text{env}d}(t) \in \mathbb{R}^6$ is the desired environmental wrench trajectory.

Definition 2.1 is conform with a wide variety of tasks. Any tasks requiring object motion rely on (2.6). Tasks which require exerting a certain wrench, such as in a window washing or pushing a button require the regulation of environmental wrenches from (2.8). Otherwise, when only free space motions are considered, environmental wrenches are avoided, i.e. $\boldsymbol{u}_{\mathrm{env}d} = \mathbf{0}$. Objective (2.7) is necessary to avoid unnecessary wrench components between agents that only produce undesired squeeze or internal object stress.

Remark 2.1. The teleoperation and kinesthetic teaching scenarios are specific instances of a cooperative manipulation scenario. In both cases the human is directly coupled to the robot at the end-effector and the Jacobian of the kinematic constraints is given by the identity matrix, i.e. $G_i = I_6$. The dynamics of the coupled system are then given by the operational space's manipulator compliance, either passive or actively rendered.

Impedance and Admittance Control

Further than a position control scheme, in order to fulfill objectives (2.7) and (2.8), the regulation of wrenches is instrumental. Among all force control methods, due to its convenient properties, the most recurrent approach in the literature is impedance/admittance control [1]. It regulates the apparent inertia, damping and stiffness of the system by means of force feedback enabling *compliant* behavior. The simplest example is given by a second-order system rendering a mass-spring-damper system. In this case, the impedance control law is given by

$$M_r\ddot{\boldsymbol{e}}(t) + D_r\dot{\boldsymbol{e}}(t) + K_r\boldsymbol{e}(t) = \boldsymbol{u}_{\mathrm{env}}(t)\,, \tag{2.9}$$

where $\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}$ and $M_r, D_r, K_r \in \mathbb{R}^{6\times 6}$ are the desired virtual mass and inertia, damping and stiffness [1]. Both impedance and admittance control follow (2.9), but their underlying assumptions are inverse. Impedance control assumes the environment behaves as an admittance and given the observed displacement $\boldsymbol{e}$ and its derivatives, the corresponding force/torque $\boldsymbol{u}$ is computed. In contrast, admittance control, also called position-based impedance control, assumes an environmental impedance and therefore the plant itself behaves as an admittance. Consequently, observed force/torques $\boldsymbol{u}$ yield a desired displacement $\boldsymbol{e}$ as dictated by (2.9). In terms of performance, their characteristics are also complementary: impedance control performs better in high stiffness environments while admittance control is more accurate when the environment is soft and vice versa. The majority of approaches in the literature implement admittance-type force controllers, as most of manipulators offer only a position interface [18].

A more general expression of (2.9) is

$$M_r\ddot{\boldsymbol{x}}(t) + D_r\dot{\boldsymbol{x}}(t) = \boldsymbol{u}_{\mathrm{imp}}(t) + \boldsymbol{u}_{\mathrm{env}}(t)\,, \tag{2.10}$$

where $\boldsymbol{u}_{\mathrm{imp}}(t)$ is a virtual force rendering a desired task-dependent behavior, for instance

[1] Note that for orientation degrees of freedom, this expression holds only if $\boldsymbol{w} \times \boldsymbol{w} J_d$ holds, where J_d represents the desired inertia. For simplicity of illustration we stick to this particular case. A more rigorous analysis can be found in [17]

following a desired configuration trajectory $\boldsymbol{x}_d$ or a desired environmental wrench trajectory $\boldsymbol{u}_{\mathrm{env}d}$. A valid example tracks $\boldsymbol{x}_d$ with a proportional derivative controller such that

$$\boldsymbol{u}_{\mathrm{imp}}(t) = K_{\dot{x}}\dot{\boldsymbol{e}}(t) + K_{x}\boldsymbol{e}(t) , \tag{2.11}$$

where $K_{\dot{x}}, K_x \in \mathbb{R}^{6\times 6}$ are stiffness and damping gains respectively. Following this idea, in [17] two robotic manipulators are controlled such that the resulting object dynamics follow impedance control law (2.10) and (2.11). This approach renders a virtual object that accommodates for external environmental forces and enables the regulation of wrenches. Synthesizing the desired system behavior (2.10) considering the real object dynamics (2.2) requires exerting a desired resulting wrench on the object

$$\boldsymbol{u}_d(t) = \boldsymbol{u}_{\mathrm{dyn}}(t) + \overline{\boldsymbol{u}}_{\mathrm{imp}}(t) - \boldsymbol{u}_{\mathrm{env}}(t) , \tag{2.12}$$

where

$$\boldsymbol{u}_{\mathrm{dyn}}(t) = M_o M_r^{-1}(\boldsymbol{u}_{\mathrm{env}}(t) - D_r\dot{\boldsymbol{x}}(\boldsymbol{t})) + \boldsymbol{h}_o(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t))$$

compensates for dynamics (2.2) and renders desired behavior (2.10) and

$$\overline{\boldsymbol{u}}_{\mathrm{imp}}(t) = M_o M_r^{-1}\boldsymbol{u}_{\mathrm{imp}}(t) .$$

The computation of the corresponding wrench trajectories for each manipulator $\tilde{\boldsymbol{u}}(t)$ yielding $\boldsymbol{u}_d(t)$ is an ill-posed problem. The possible solutions depend on a decomposition matrix $A \in \mathbb{R}^{6N\times 6}$ which distributes the load among agents such that

$$\tilde{\boldsymbol{u}}(t) = A\boldsymbol{u}_d(t) \qquad \text{and} \qquad \boldsymbol{u}_d(t) = GA\boldsymbol{u}_d(t) \tag{2.13}$$

hold. A usual approach in the literature is given by the Moore-Penrose pseudoinverse of G, denoted G^+, which yields the solution minimizing $||\boldsymbol{u_d} - A\tilde{\boldsymbol{u}}||$ with minimum norm $\tilde{\boldsymbol{u}}$ and thereby assigning a uniform load allocation. The wrench of each manipulator is then given by

$$\tilde{\boldsymbol{u}}_d(t) = G^+\boldsymbol{u}_d(t) . \tag{2.14}$$

Under ideal conditions, the controls resulting from (2.10), (2.11) and (2.14) fulfill objectives (2.6) and (2.7) [19]. However, the deployment of real cooperative manipulation systems is significantly more challenging due to sensorimotor uncertainties, which give rise to unexpected internal stress [20]. High interaction forces and torques do not just hinder tracking performance but also significantly compromise safety.

2.1.2 Control Challenges in Physical Human-Robot Interaction

Although the main challenge of control design for multi-robot cooperative manipulation tasks are the effects of sensorimotor uncertainties, when one of the agents is a human the control task acquires a different scope.

Consider now the setting from Section 2.1.1 with an additional human agent. Let $\widetilde{\boldsymbol{u}}_h \in \mathbb{R}^6$ be the applied wrench of the human in frame C and $\boldsymbol{p}_h \in \mathbb{R}^3$ the vector from the human grasp position to the object frame. The corresponding human partial grasp matrix defined as in (2.5) is denoted G_h. Object dynamics from (2.2) are now given by

$$M_o\ddot{\boldsymbol{x}}(t) + \boldsymbol{h_o}(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t)) = \boldsymbol{u}(t) + \boldsymbol{u}_h(t) + \boldsymbol{u}_{\text{env}}(t) \,, \tag{2.15}$$

where $\boldsymbol{u}_h = G_h\widetilde{\boldsymbol{u}}_h(t)$. Although task definition 2.1 holds for the pHRI case, control design based on (2.10) and (2.14) is in general not attainable any more. The presence of a physically coupled human partner implies two additional challenging issues:

- The desired object configuration trajectory $\boldsymbol{x}_d(t)$ and environmental wrench trajectory $\boldsymbol{u}_{\text{env}d}(t)$ from (2.6) and (2.8) are unknown due to unpredictable human actions. Even when a final goal configuration is known to both the human and the robot, any assumption on the human-in-the-loop desired object trajectory or environmental wrench trajectory at every time instant is, in most settings, unrealistic.

- The human-desired load share or, equivalently, the human-desired input $\widetilde{\boldsymbol{u}}_{d,h}$ concerning (2.7) is unknown. This fact hinders the design of matrix A which distributes the load among agents as given in (2.13).

In order to cope with these issues there are two options in the literature. On one side, *reactive* assistants consider the human an exogenous environmental agent. The human applied wrench is modeled as an external wrench component and is accommodated by means of a force control scheme. On the other side, *proactive* assistants estimate or assume feed-forward models λ of the human-in-the-loop desired object trajectory $\boldsymbol{x}_d(t)$, desired individual wrench inputs $\widetilde{\boldsymbol{u}}_d(t)$ (including $\widetilde{\boldsymbol{u}}_{d,h}$) and desired environmental wrench $\boldsymbol{u}_{\text{env}d}(t)$. These approaches yield model-based control schemes that anticipate human actions instead of just compensating for them. The following subsections review in detail related works following these two different approaches.

2.1.3 Reactive assistants

Reactive assistants consider the human wrench input as an additional environmental disturbance and compensate for it by means of force control. Following the approach presented in [17] and explicitly modeling the human operator as an exogenous input, comfortable and intuitive interaction of humans with multiple robots is deployed in [21] by rendering a mass-damper system

$$M_r\ddot{\boldsymbol{x}}(t) + D_r\dot{\boldsymbol{x}}(t) = \boldsymbol{u}_{\text{ext}}(t) = \boldsymbol{u}_h(t) + \boldsymbol{u}_{\text{env}}(t) \,. \tag{2.16}$$

The external human wrench or any environmental contacts are compensated in an intuitive manner enabling the firsts *reactive* physical assistants. This approach is further extended in a decentralized manner in [22]. It is remarkable that, assuming the human hand behaves as a passive element plus an independent exogenous signal [23], the stability of this system is guaranteed [24].

In terms of definition (2.1), no robot-desired trajectory nor environmental contacts are considered, i.e. $\boldsymbol{u}_{envd}(t) = \boldsymbol{u}_h(t) = \mathbf{0}$ is the only control goal to be fulfilled. The agents' load distribution A from (2.13) is also given by the Moore-Penrose pseudoinverse but the human operator is not considered an agent of the system and thereby is ignored in the decomposition.

Reactive assistants, although effective and intuitive, do not reduce significantly human work but rather serve as an physical human-robot interface. In fact, reactive helpers increase the necessary load as the robot acts as an additional impedance in the overall dynamics.

2.1.4 Proactive assistants

In contrast to reactive assistants, proactive approaches solve the cooperative manipulation problem considering the human as a constitutive part of the multi-agent system and estimating its preferences. From a psychological perspective, this approach is necessary for successful *cooperation*. Psychologists emphasize that cooperation allows multiple agents to jointly work towards a shared goal but implies awareness of other interacting agents in terms of reasoning and knowing about their intentions and capabilities. It further imposes the requirement of implicitly or explicitly communicating the goal [25]. In this context, in [26] a reactive control scheme was successfully transferred to a humanoid robot and combined with a predefined movement generation routine based on directions from human speech inputs. However, wherever explicit communication between robot and human is infeasible, predicting outcomes of the partner's actions is a requirement for successful interaction during joint action [3]. Following this idea, proactive assistants consider *behavior models* λ comprising motion, force or both of them, estimating the human-desired trajectory $\boldsymbol{x}_d$, individual wrench inputs $\widetilde{\boldsymbol{u}}_d(t)$ (including $\widetilde{\boldsymbol{u}}_{d,h}$) and/or environmental wrench trajectory $\boldsymbol{u}_{\mathrm{env}d}$.

Definition 2.2. A *behavior model* λ is given by the function $\boldsymbol{f} : \mathbb{R}^n \rightarrow \mathbb{R}^{12+6N}$ such that

$$\begin{bmatrix} \boldsymbol{x}_d(t) \\ \widetilde{\boldsymbol{u}}_d(t) \\ \boldsymbol{u}_{\mathrm{env}d}(t) \end{bmatrix} = \boldsymbol{f}(\xi(t)) ,$$

where $\xi(t) = \begin{bmatrix} \boldsymbol{x}(t) & \dot{\boldsymbol{x}}(t) & \boldsymbol{u}_{\mathrm{env}}(t) & \widetilde{\boldsymbol{u}}(t) & \xi_{\mathrm{env}}(t) \end{bmatrix}$ is the input space and ξ_{env} are environment or task-related features.

These models serve as a predictive reference for the robot in order to effectively cooperate with its partner and urge towards the human-desired goal. Proactive control schemes achieve higher transparency than reactive ones, i.e. they are able to follow/anticipate human movements with lower human-perceptible resistive forces [4]. Intuitively speaking, a correct prediction of a human partner's desired motion can yield better results than a technical implementation of zero-force control. Furthermore, goal-oriented behavior is even necessary to solve a manipulation task if parts of the task state space are not reachable for a single agent or difficult to control by the human, for example when transporting a voluminous object thorough a cluttered environment [2].

In general, proactive assistants render an overall system behavior given by

$$M_r\ddot{\boldsymbol{x}}(t) + D_r\dot{\boldsymbol{x}}(t) = \boldsymbol{u}_r(t) + \boldsymbol{u}_h(t) + \boldsymbol{u}_{\mathrm{env}}(t) , \tag{2.17}$$

where $\boldsymbol{u}_r \in \mathbb{R}^6$ is the resulting proactive control input of the robots. Instead of only acting as a human-robot interface as in (2.16), the presence of the additional input $\boldsymbol{u}_r$ enables anticipation by exploiting the feed-forward knowledge provided by λ.

The characterization of behavior models λ for pHRI is very diverse. Some behavior models present a discrete latent state space. These models yield hybrid control schemes that switch between different admittance parameters triggered by the recognition of predefined sets of haptic and motion patterns in [27,28]. Predicting human intention from the force derivative provides a way to adapt the rendered damping D_r [29] while ensuring stability [30]. While these approaches synthesize indirectly a proactive input by modifying the rendered admittance, the simplicity of the behavior model suggest that they should be classified as adaptive reactive assistants rather than proactive ones.

A different line of work rely on *analytical models* of human behavior in order to estimate $\boldsymbol{x}_d$. In this case predictions $\boldsymbol{f}(\xi(t))$ from definition 2.2 are given by an a priori assumed analytical function of the inputs. The well-known minimum jerk principle [31] is exploited either on a position level in [32] or on a velocity level in [33]. However, this principle does not hold for cooperative manipulation tasks, where a polynomial extrapolation becomes more suitable [34]. When tasks of higher complexity involving sequences of multiple, previously unknown primitive actions are considered these works can serve only as guidelines as they just consider local predictive models. As an alternative, planning approaches provide suitable task solving behavior models when both the common goal and the environment are known. Especially compelling are feedback motion planning methods [35, 36], which continuously adapt their predictions when human deviations occur [37]. However, besides requiring accurate environment models, considering human dynamic preferences in motion planners imply prohibitive computation times due to increased complexity.

A recent trend in haptic assistance follow the application of *learning techniques* and more precisely the Programming by Demonstration (PbD) paradigm [38] to generate models of human motion behavior. In this context, the cooperation behavior synthesis presented by [39] represents the first steps of strategies involving learning and recognizing human force patterns in a large-scale setup to generate an appropriate object position trajectory. The recognition performance of left-to-right HMMs on Fourier-transformed force data as well as the generation of hand-crafted robot actions serve as a starting point for developments directed towards more reliable intention recognition and and intuitive robotic actions in the recognized task phases. Acquiring demonstrations through teleoperation, human-like interactive behavior is reproduced considering learned motions and interaction forces [40–42]. Interpreting the interaction as a continuous teaching process, the robot may evolve from a passive follower into an active contributor using iterative learning techniques [43]. Although these approaches rely on probabilistic models, their decision-making schemes do not exploit the probabilistic nature of predictions. In addition, no methods offering online learning capabilities are explored in the literature. In fact, most of PbD-based approaches are implemented considering separated learning and reproduction phases.

Regardless of the method used, the goal-oriented behavior that all above-mentioned proactive approaches synthesize implies the possibility of conflicts with the interacting partner. An insight on the consequences of this problem is acquired by observing haptic interaction between physically coupled human-human dyads during point to point tasks. Humans behave in a goal-oriented fashion and, while they do cooperate towards a final common goal, they have to continuously negotiate a trajectory to follow. Studies highlight the importance of haptic feedback for the continuous negotiation process between partners [44, 45] and indicate the existence of different roles [46]. These results motivate the synthesis of interaction roles, i.e. the load share that each partner is responsible for, defined by matrix A in (2.13). Following this idea, a possible framework is presented in [47], where different roles are achieved using a controller blending scheme. This concept is exploited by adjusting the robot role depending on the level of prediction confidence in [48]. Other options smoothly adapt the role of the robot depending on the human force magnitude [49] or a measure of disagreement between partners [50], while following a predefined trajectory. The large scale human user study presented in [49] shows the advantages of a dynamic role allocation scheme. Similar results are outlined in [51], where the scope of roles is explored using haptic shared control in a driver supporting scenario. These works rely on the robot role adaptation based on some measure of disagreement. However, there is no consensus in the literature about how to evaluate physical disagreement between N goal-oriented agents or how should the robotic role be adapted.

This thesis explores the following open issues of proactive physical assistants:

- **Learning during interaction** . Online learning and assistive control based on a constantly updated model of $\boldsymbol{x}_d$ and $\boldsymbol{u}_{\mathrm{env}d}$ impose additional challenges both in the model learning algorithm and the decision-making process that synthesizes robotic assistance. This process can be interpreted as a traditional PbD setting where learning and reproduction take place at the same time. We explore both task-space trajectory learning and dynamics learning models.

- **Disagreement and load share characterization** . We explore possible measures of disagreement between goal-oriented agents with six-dimensional force/torque inputs. We put special emphasis on a physically consistent and orientation independent decomposition A from (2.13) that yields effective and internal or disagreement wrench components for arbitrary load distributions.

- **Uncertainty- and disagreement-dependent assistance** . We exploit the probabilistic nature of predictions in order to avoid in advance potential disagreements with the human partner. Modeling robotic assistance as a stochastic optimal control problem, we explore possibilities that adapt the robotic contribution depending on prediction uncertainty, model confidence and disagreements. In addition, we also study control schemes for tasks comprising environmental contacts.

2.1.5 Performance Measures for pHRI

In order to assess the assistive performance of physical helpers, studies exploring human-human dyads provide adequate guidelines. A good overview of possible *implicit* measures is given in Chapter 3 of [52]. The evaluation of physical human-robot experiments is similar. In the following we describe measures related to a specific task or to the physical interaction itself concerning applied wrenches. Note that, additionally, *explicit* or subjective measures reveal human preferences considering non-physical aspects. Possible measures are human comfort or perceived helpfulness and are evaluated by means of questionnaires after trials.

Task-related measures

Concerning tasks with a predefined known trajectory $\boldsymbol{x}_d(t)$, the mean-squared tracking performance is a measure of tracking accuracy given by

- Root-mean-square-error $RMSE = \sqrt{\frac{1}{T} \int\limits_{t=0}^{T} ||\boldsymbol{x}_d(t) - \boldsymbol{x}(t)||dt}$,

where T is the task execution time.

When the desired trajectory is not fixed, the execution time may vary and is a valid performance measure. Considering $E \in \mathbb{Z}^+$ trials of the same task, the average execution time is given by

- Mean execution time $\frac{1}{E} \sum\limits_{e=1}^{E} T_e$,

where T_e is the execution time of the e-th trial.

If a cluttered environment is considered, the number of collisions or the average environmental force per collision are also indicators of human safety and comfort.

Wrench-related measures

An evaluation in terms of human applied force or torque reveals additional aspects of the interaction. Especially illustrative is the human power, an indicator for the capability of the robotic assistant to take over the overall work. The respective measures are given by

- Mean absolute human force $\sqrt{\frac{1}{T} \int\limits_{t=0}^{T} ||\boldsymbol{f}_h(t)||dt.}$

- Mean absolute human torque $\sqrt{\frac{1}{T} \int\limits_{t=0}^{T} ||\boldsymbol{\tau}_h(t)||dt.}$

- Mean human power $\sqrt{\frac{1}{T} \int\limits_{t=0}^{T} \boldsymbol{u}_h^{\mathrm{T}}(t)\dot{\boldsymbol{x}}(t)dt.}$

An important measure of efficiency of the interaction is also given by the *disagreement* between agents. There is no consensus in the literature about its definition. This issue will be considered in detail in Chapter 5.

2.2 Behavior Learning from Demonstrations

A significant body of work has been dedicated to modeling human behavior in the field of programming by demonstration [53]. In the PbD paradigm a human teacher performs a task one or several times while the robot observes the execution by means of vision or any other sensor modality. Observed data is the basis for building a task behavior model in a statistical or probabilistic manner, thereby capturing potential task constraints. The resulting model is later used by the robot in order to perform the task autonomously. Depending on the abstraction level, the behavior model represents the task either on a *symbolic* level or a *trajectory* level. In this thesis we put special emphasis on behavior models encoded in a trajectory level. For further details on behavior encoding at a symbolic level, where data is abstracted in a high-level into a set of predefined symbols, we refer the interested reader to [38]. Among all trajectory-based methods, in this section we consider three methodologies with different underlying assumptions:

- Assuming the task is described by a trajectory with a given duration, some approaches exploit statistical analysis in order to build a generalized *time-series* model of the task.
- Due to its convenient properties for later application in control, other works aim for a *dynamical system* task representation.
- Assuming the principle of rational action, learning *cost functions* that best explain observed behavior enable the modulation of rich motions.

These are valid options for the acquisition of the necessary behavior models λ from definition 2.2 that proactive physical helpers require, as explained in the previous Section. In order to learn the human *desired* configuration and wrench trajectories, observations from previous demonstrations are assumed to be close to the human intentions, i.e. observed $\boldsymbol{x} \approx \boldsymbol{x}_d$, observed $\widetilde{\boldsymbol{u}} \approx \widetilde{\boldsymbol{u}}_d$ and observed $\boldsymbol{u}_{\text{env}} \approx \boldsymbol{u}_{\text{env}d}$. The rest of this section explores each of these three possibilities in detail. Section 2.2.1 reviews time-series models, Section 2.2.2 dynamical system models and Section 2.2.3 inverse optimal control methods.

2.2.1 Learning Time-series Models

The original works representing the initial PbD steps are found in [54], [55] and [56] considering time-series task representations. These initial efforts focus on the control problem of reproducing recorded task or joint space trajectories. Further than just replaying previous trials, an emphasis in the exploitation of variability observed among several trials of the same task is first found in [57]. Observed variance is related to the necessary precision of the control task in [58]. This fact highlights the importance of statistical time-series analysis in order to acquire richer models. Considering definition 2.2, time-series models restrict their input space to

$$\begin{bmatrix} \boldsymbol{x}_d(t) \\ \widetilde{\boldsymbol{u}}_d(t) \\ \boldsymbol{u}_{\text{env}d}(t) \end{bmatrix} = \boldsymbol{f}(\xi_t(t)) \,, \tag{2.18}$$

where $\xi_t \in \mathbb{R}$ is a temporal value, usually given by $\xi_t = t$. In this context, the robustness and flexibility of Hidden Markov Models (HMMs) have been widely applied [59]. Their ability to recognize and generate time-series is similar to the biological behavior of mirror neurons, neural regions of the brain which activate both observing a certain motion and performing it [60]. When continuous observation spaces are considered, the observation densities of each HMM state are usually modeled by means of a Gaussian Mixture Model (GMM) [61]. Due to their relevance in time-series modeling, we now briefly define formally the GMM and its applications in this context followed by a brief explanation of the HMM.

Gaussian Mixture Models

A GMM with $C \in \mathbb{Z}^+$ Gaussian components assumes that observations $\boldsymbol{\xi} \in \mathbb{R}^n$ are distributed as

$$\mathsf{P}(\boldsymbol{\xi}|\boldsymbol{w},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{c=1}^{C} w_c \mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu}_c, \Sigma_c) , \tag{2.19}$$

where

- $\boldsymbol{w} = \begin{Bmatrix} w_1 & w_2 & \cdots & w_C \end{Bmatrix}$ with $w_c \in \mathbb{R}$ are the weighting factors or prior probabilities,
- $\boldsymbol{\mu} = \begin{Bmatrix} \boldsymbol{\mu}_1 & \boldsymbol{\mu}_2 & \cdots & \boldsymbol{\mu}_C \end{Bmatrix}$ with $\boldsymbol{\mu}_c \in \mathbb{R}^n$ are the Gaussian means and
- $\boldsymbol{\Sigma} = \begin{Bmatrix} \Sigma_1 & \Sigma_2 & \cdots & \Sigma_C \end{Bmatrix}$ with $\Sigma_c \in \mathbb{R}^{n\times n}$ are the Gaussian covariance matrices.

Given a set of E observations $\Xi = \{\boldsymbol{\xi}_e\}_{e=1}^{E}$, the expectation-maximization (EM) algorithm [62] computes iteratively the GMM parameters $\left\{\widetilde{\boldsymbol{w}}, \widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\Sigma}}\right\}$ that maximize the likelihood (2.19), i.e. $\prod_{e=1}^{E} \mathsf{P}(\boldsymbol{\xi}_e|\boldsymbol{w},\boldsymbol{\mu},\boldsymbol{\Sigma})$. As a result, the trained parameters capture the statistical properties of the training dataset.

In order to represent time-series with GMMs, the observation space is modeled as $\boldsymbol{\xi} = \left[\boldsymbol{\xi}_s^{\mathrm{T}} \ \xi_t\right]^{\mathrm{T}}$, where $\boldsymbol{\xi}_s \in \mathbb{R}^{n-1}$ is the spatial vector and $\xi_t \in \mathbb{R}$ is its corresponding temporal value [63]. A GMM with these characteristics trained for a given training set Ξ captures the statistical dependencies between time and spatial data, which, for the c-th Gaussian are given by

$$\boldsymbol{\mu}_c = \begin{bmatrix} \boldsymbol{\mu_s}_c^{\mathrm{T}} & \mu_{t_c} \end{bmatrix}^{\mathrm{T}} \quad \Sigma_c = \begin{bmatrix} \Sigma_{ss_c} & \Sigma_{st_c} \\ \Sigma_{ts_c} & \Sigma_{tt_c} \end{bmatrix} .$$

By means of Gaussian conditioning, the expected value of spatial data for the c-th Gaussian is given by

$$\mathsf{P}(\boldsymbol{\xi}_s|\xi_t, \boldsymbol{\mu}_c, \Sigma_c) = \mathcal{N}(\boldsymbol{\xi}_s|\hat{\boldsymbol{\mu}}_c, \hat{\Sigma}_c) , \tag{2.20}$$

where $\hat{\boldsymbol{\mu}}_c = \boldsymbol{\mu_s}_c + \Sigma_{st_c}\Sigma_{tt_c}^{-1}(\xi_t - \mu_{t_c})$ and $\hat{\Sigma}_c = \Sigma_{ss_c} + \Sigma_{st_c}\Sigma_{tt_c}^{-1}\Sigma_{ts_c}$. The expected value of the spatial vector for a given temporal value ξ_t considering all components and its mixture is then given by

$$\mathsf{P}(\boldsymbol{\xi}_s|\xi_t, \boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{c=1}^{C} \gamma_c(\xi_t) \mathcal{N}(\boldsymbol{\xi}_s|\hat{\boldsymbol{\mu}}_c, \hat{\Sigma}_c) , \tag{2.21}$$

where $\gamma_c(\xi_t)$ is the *responsibility* of the c-th Gaussian for observation ξ_t and is given by

$$\gamma_c(\xi_t) = \frac{w_c \mathcal{N}(\xi_t | \mu_{t_c}, \Sigma_{tt_c})}{\sum\limits_{i=1}^{C} w_i \mathcal{N}(\xi_t | \mu_{t_i}, \Sigma_{tt_i})} \,.$$

Note that a regression task is usually written in the form $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{\xi})$, where $\boldsymbol{y}$ is the output and $\boldsymbol{\xi}$ the input. In this particular case, (2.21) corresponds to

$$\xi_s = \boldsymbol{f}(\xi_t)\,, \tag{2.22}$$

i.e. this particular application performs regression in time domain considering time information as an input and the spatial data as an output. This expression matches definition 2.2 and (2.18), which characterize time-series behavior models. In the context of GMMs, the regression task exploiting the conditional distributions between dimensions of the observation space is denoted *Gaussian Mixture Regression* (GMR) [64].

Hidden Markov Models

An HMM with continuous observation densities is a statistical model composed of $S \in \mathbb{Z}^+$ discrete states where each state is represented by a GMM with $C \in \mathbb{Z}^+$ components. The dynamics of the latent discrete state space are assumed to be Markovian. In the following, let $\boldsymbol{\xi} \in \mathbb{R}^n$ be an observation at discrete time k and $q_k \in \mathbb{S}$ with $\mathbb{S} = \{1, 2 \cdots S\}$ its corresponding latent state.

More formally, an HMM is given by the tuple $\lambda = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{W}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, where

- $\boldsymbol{\pi} = \{\pi_i\}$ with $\pi_i \in [0, 1]$ are the initial state probabilities, such that $\forall i \in \mathbb{S} \quad \mathsf{P}(q_0 = i) = \pi_i$.
- $A = \{a_{ij}\}$ with $a_{ij} \in [0, 1]$ is the transition probability matrix describing the latent space dynamics such that $\forall i, j \in \mathbb{S} \quad \mathsf{P}(q_{k+1} = j | q_k = i) = a_{ij}$.
- $\boldsymbol{W} = \{w_{ic}\}, \boldsymbol{\mu} = \{\boldsymbol{\mu}_{ic}\}, \boldsymbol{\Sigma} = \{\Sigma_{ic}\}$ are the weight, mean and covariance matrix of the c-th mixture component of i-th state such that $\forall i \in \mathbb{S} \quad \mathsf{P}(\boldsymbol{\xi} | q_k = i) = \sum\limits_{c=1}^{C} w_{ic} \mathcal{N}(\boldsymbol{\xi} | \boldsymbol{\mu}_{ic}, \Sigma_{ic})$.

Given a set of E observations *sequences* $\overline{\Xi} = \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots \boldsymbol{\xi}_{T_e}\}_{e=1}^{E}$, the Baum-Welch algorithm computes iteratively and efficiently the HMM parameters $\widetilde{\lambda}$ that maximize the likelihood $P(\overline{\Xi}|\widetilde{\lambda})$ by means of dynamic programming [62]. Similarly, the Viterbi algorithm works as a state estimator by computing the most probable latent space sequence given an observation sequence. Detailed explanations of both procedures are given in [61].

Note that due to the latent space dynamics, the HMM is a generative model: for a given initial state q_0 an arbitrary sequence of expected observations can be simulated by applying Markovian dynamics A and their corresponding observation densities.

Left-to-right HMMs have been proved efficient for encoding time-series data. In this case the transition matrix is constrained to the form

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1S} \\ 0 & \ddots & & \vdots \\ \vdots & & & \\ 0 & \cdots & 0 & a_{SS} \end{bmatrix} ,$$

where $a_{SS} = 1$ neglects any dynamics at the *final state*. This constrained form suits well time-series segments; intuitively speaking, the latent state acts as a discretized temporal value. In this case, following a similar notation as in definition 2.2 and (2.18) predictions are given by

$$\xi = f(q_k) = \mathsf{E}[\mathsf{P}(\xi | q_k)] \, . \tag{2.23}$$

In contrast to (2.22) the temporal value is now represented by the latent state and predictions are limited to the expected value. In combination with hierarchical clustering, left-to-right HMMs enable continuous incremental improvement over a large training data set, as successfully shown by [65]. Similar to [65], [66] and [67] proposed discrete incremental learning frameworks for symbols and gestures respectively, using HMMs. A promising approach towards an efficient incremental learning and representation of human motion is presented by [68]. *Growing HMMs* allow the state space of an HMM to grow depending on the granularity of observations. This technique is used to overcome the typical limitation of HMMs to adapt to the actual, often underestimated complexity of the modeled system. However, the topological state space representation in their work inhibits a separation of overlapping tasks in contrast to primitive-based representations.

The main impediment of HMMs for the reconstruction of time-series is the discrete nature of its state space, as shown in (2.23). An interesting alternative is the reformulation of the HMM as a trajectory model [69] at the cost of loosing the recognition and estimation flexibility of its initially discrete state space. Similar to [63] with GMMs, in this thesis we exploit the applicability of GMR to HMMs by explicitly learning time information. We will explore this idea, first introduced in [70], in order to enable smooth time regression together with flexible state estimation capabilities.

2.2.2 Learning Dynamical Systems

The application of dynamical systems theory to robot learning in the PbD framework has gained remarkable attention in recent years. In general and considering definition 2.2, dynamical systems express human desired behavior as a differential equation

$$\xi^{(n)} = f(\xi, \xi^{(1)}, \cdots, \xi^{(n-1)}, t) \, , \tag{2.24}$$

where ξ is denoted the state of the system and $\xi^{(n)} = \frac{d^n \xi}{dt^n}$ represents the n-th time derivative of ξ. The vector field f is assumed to be a non-linear continuous and continuously differentiable mapping.

The first efforts emphasizing the convenience and robustness of dynamical system-based approaches are Dynamical Movement Primitives (DMPs) [71], where motions are assumed to be goal-oriented with second-order characteristics and perturbed by a non-linear term. Another alternative assumes a constant vector field $\boldsymbol{f}$ in (2.24), i.e. no time dependency is considered, yielding

$$\xi^{(n)} = \boldsymbol{f}(\xi, \xi^{(1)}, \cdots, \xi^{(n-1)}) , \tag{2.25}$$

This idea is first exploited in the PbD setting in [72] for first-order dynamics and more in depth in [73] and [74] for second-order systems. Similarly but not explicitly in this context, the vector field describing the second order dynamics of complex manipulators is also learned in an incremental manner in joint [75] or task space [76]. In this case, $\boldsymbol{f}$ takes the form

$$\ddot{\xi} = \boldsymbol{f}(\xi, \dot{\xi}, \boldsymbol{u}) , \tag{2.26}$$

where $\boldsymbol{u} \in \mathbb{R}^m$ is the manipulator's control input. These approaches enable the autonomous acquisition of dynamic or inverse kinematic models by exploration.

The above-mentioned approaches rely on regression techniques in order to characterize the non-linear vector field $\boldsymbol{f}$ in (2.25) and (2.26) or the additive non-linear forcing term in the case of DMPs, enabling the encoding of complex dynamics. In the following we review the most recurrent regression methods in the literature and their respective applications in robotics as well as DMPs.

GMMs and HMMs

The regression capabilities of both GMMs and HMMs for learning the vector field $\boldsymbol{f}$ as in (2.25) are first applied in [73]. By means of GMR as explained in Section 2.2.1, the second order dynamics of observed motions are encoded as a unique static function when using a GMM. In the case of an HMM, the function $\boldsymbol{f}$ also reflects the latent space dynamics of the HMM and 'navigates' along the GMMs of the different states. Although effective, both approximations lack stability guarantees. The issue is first addressed in [74] where local asymptotic stability is guaranteed calculating safe numerically approximated boundaries. Asymptotic stability is guaranteed in [77] assuming a final known attractor.

Gaussian Processes

A Gaussian process (GP) $f(\xi)$, with $\xi \in \mathbb{R}^n$ and $f(\xi) : \mathbb{R}^n \to \mathbb{R}$, is a stochastic process where any finite collection of samples $\boldsymbol{f}_E = \{f_1 \cdots f_E\}$, where $f_e = f(\xi_e)$, forms a multivariate Gaussian random variable. A GP is completely defined by its mean function

$$m(\xi) = \mathsf{E}[f(\xi)]$$

and covariance function

$$k(\xi, \xi') = \mathsf{E}[(f(\xi) - m(\xi))(f(\xi') - m(\xi'))^{\mathrm{T}})] .$$

The definition of $f(\xi)$ as a GP is compactly formulated as $f(\xi) \sim \mathcal{GP}(m(\xi), k(\xi, \xi'))$. Choosing the most appropriate covariance function $k(\xi, \xi')$, which defines the smoothness and overall properties of the GP is key for acquiring a successful model.

Assuming a GP prior over the vector field of (2.25) or (2.26) implies that, a priori,

$$\boldsymbol{f}_E \sim \mathcal{N}(\boldsymbol{f}_E|\mathbf{0}, K) . \tag{2.27}$$

When applying GPs for regression [78], observations are usually assumed to be noisy, such that

$$y_e = f(\xi_e) + \epsilon , \tag{2.28}$$

where $y_e \in \mathbb{R}$ is the e-th one-dimensional output and $\epsilon \sim \mathcal{N}(\epsilon|0, \sigma_n^2)$. Due to the homogeneity of noise, i.e. all observations y_e have the same finite prior, this model is referred as *homoscedastic*. $\mathcal{N}(\epsilon|0, \sigma_n^2)$ represents the a priori *likelihood* of the model.

In the following, a set of training inputs is denoted by $\Xi = \{\xi_e\}_{e=1}^E$ and its corresponding outputs $\boldsymbol{y} = \{y_e\}_{e=1}^E$. By convention, the covariance vector of a test input ξ_* and the training inputs is denoted $k(\Xi, \xi_*) = \boldsymbol{k}_*$, the autocovariance of a test input $k(\xi_*, \xi_*) = k_{**}$ and the covariance matrix of the training inputs is given by $k(\Xi, \Xi) = K$. Without lack of generality, we further assume a 0-mean prior, i.e. $m(\xi) = 0$. Note that for multivariable targets $\boldsymbol{y}$ as $\xi^{(n)}$ in (2.25) or $\ddot{\xi}$ in (2.26), for simplicity, independent GPs are considered for each output dimension. If necessary, multi-output GPs offer output correlation but at the cost of higher computational complexity [79].

From (2.28) it follows that

$$\mathsf{P}(y_e|f_e) = \mathcal{N}(y_e|f_e, \sigma_n^2) . \tag{2.29}$$

Consequently, from (2.29) and (2.27) and given a test input ξ_* and its assotiated function value f_* the joint density of all involved quantities is given by

$$\mathsf{P}(\boldsymbol{f}_E, f_*, \boldsymbol{y}|\Xi, \xi_*) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{f}_E \\ \boldsymbol{y} \\ f_* \end{bmatrix} \middle| \mathbf{0}, \begin{bmatrix} K & K & \boldsymbol{k}_* \\ K & K + \sigma_n^2 I & \boldsymbol{k}_* \\ \boldsymbol{k}_*^{\mathrm{T}} & \boldsymbol{k}_*^{\mathrm{T}} & k_{**} \end{bmatrix}\right) . \tag{2.30}$$

With this joint distribution and using Gaussian conditioning as in (2.20), the posterior distribution for f_* is given by

$$\begin{aligned} \mathsf{P}(f_*|\boldsymbol{f}_E, \boldsymbol{y}, \Xi, \xi_*) =& \mathcal{N}(f_*|\mu_{f_*}, \sigma_{f_*}^2) = \\ & \mathcal{N}(f_*|\boldsymbol{k}_*^{\mathrm{T}}(K + \sigma_n^2 I)^{-1}\boldsymbol{y}, k_{**} - \boldsymbol{k}_*^{\mathrm{T}}(K + \sigma_n^2 I)^{-1}\boldsymbol{k}_*) . \end{aligned} \tag{2.31}$$

The posterior predictive distribution of y_* follows immediately by recalling prior (2.29), yielding

$$\mathsf{P}(y_*|f_*, \boldsymbol{f}_E, \boldsymbol{y}, \Xi, \xi_*) = \mathcal{N}(y_*|\mu_{f_*}, \sigma_{f_*}^2 + \sigma_n^2) . \tag{2.32}$$

Note that the complexity of this computation is governed by the matrix inversion $(K + \sigma_n^2 I)^{-1}$ and thereby is $\mathcal{O}(E^3)$. To apply GPs for regression in realistic scenarios, sparse or local approximations are usually necessary.

GPs have been used for many applications in robotics. Some examples of its efficacy are [75] or [76] where the joint or task space dynamics are learned. The controls to perform a task are also learning in an online manner using sparse approximations in [80]. Other applications include the error modeling of non-accurate analytic descriptors [81], [82]. GPs have also been used to improve the efficiency of reinforcement learning algorithms by learning the initially unknown plant dynamics of an manipulator [83]. In order to learn complex high-dimensional functionals, an additional latent space is also considered yielding Gaussian Process Latent Variable Models (GPLVM) [84]. A especially compelling application of this model are Gaussian Process Dynamical Models (GPDM) which model the latent space as the system state and the observed data space as the observation space [85] at the cost of high computational complexity. In this thesis we explore the application of GPs in order to learn the dynamics of a human performing a task in the form (2.25). We explore solutions assuming well-known impedance properties of the human arm as well as sparse approximations yielding applicable models for the anticipatory control of physical helpers.

Locally Weighted Projection Regression

Locally Weighted Projection Regression (LWPR) is a regression method applied in many robotics settings. It assumes that outputs are distributed following a weighted combination of N local models

$$y_* = \frac{\sum_{i=1}^{N} w_i(\xi) y_{*i}(\xi)}{\sum_{i=1}^{N} w_i(\xi)} \,, \tag{2.33}$$

where $y_*(\xi) \in \mathbb{R}$ is the one-dimensional output, $y_{*i}(\xi)$ is the one-dimensional local output and $\xi \in \mathbb{R}^n$ is the input. The local weighting factors $w_i(\xi)$ are given by an unnormalized Gaussian density, such that

$$w_i(\xi) = \exp\left\{-\frac{1}{2}(\xi - \boldsymbol{\mu}_i)^{\mathrm{T}} D_i (\xi - \boldsymbol{\mu}_i)\right\} \,.$$

Each of the local output predictions $y_{*i}(\xi)$ are given by a sum of weighted projection computed with an online formulation of the partial least squares method (PLS) [86]. PLS projects jointly input and output spaces in order to compute the projection of the input which explains maximum output variance. The local outputs are given by

$$y_{*i}(\xi) = \beta_0 + \sum_{k=1}^{R} \beta_k s_k \,,$$

where $\boldsymbol{\beta} = \left\{\beta_0, \beta_1 \cdots \beta_R\right\}$ are weighting factors and $s_k = \xi^{\mathrm{T}} \boldsymbol{z}_k$ are the projections of the input on projection vector $\boldsymbol{z}_k$. Note that each projection vector is updated in an online fashion.

Parameters $\boldsymbol{\beta}$ and D_i are learned from data maximizing the cross correlation and updated online using stochastic gradient descent. For later interest, it is also remarkable that a Bayesian approach to LWPR results from assuming a prior for each local model such that

$$\tilde{y}_{*_i}(\xi) = y_{*i}(\xi) + \epsilon_1 + \epsilon_{2,i} \,,$$

where $\epsilon_1 \sim \mathcal{N}(\epsilon|0, \sigma^2/w_i)$ is global noise process and $\epsilon_{2,i} \sim \mathcal{N}(\epsilon|0, \sigma^2_{pred,i}/w_i)$ is a local noise process and with $y_{*i}(\boldsymbol{\xi}) = \tilde{y}_{*_i}(\boldsymbol{\xi})$ in (2.33). An expression for the predictive variance can be found in [87]. Note that, as GPs, LWPR assume statistically independent outputs modeling each output dimension independently.

LWPR has been the principal regression technique for modeling the external disturbance present in DMPs [71], explained in the next subsection. However, its online regression capabilities and low computational complexity have also found room for applications in general robotics regression problems. For example, the dynamics of manipulators have been learned with LWPR models in [88]. Additional applications for control are summarized in [89].

Dynamical Movement Primitives

Dynamical Movement Primitives, originally proposed in [71] for learning non-linear attractor dynamics, assume a stable one-dimensional dynamical model given by damped spring model

$$\tau \ddot{x}(t) = k_1(k_2(g - x(t)) - \dot{x}(t)) + f(\alpha)\,, \tag{2.34}$$

where $x \in \mathbb{R}$ is the state, $g \in \mathbb{R}$ is the goal, $f \in \mathbb{R}$ is a non-linear forcing term, $k_1, k_2 \in \mathbb{R}^+$ are positive constants and α is a phase variable with dynamics

$$\tau \dot{\alpha}(t) = -k_\alpha \alpha(t)\,, \tag{2.35}$$

where $k_\alpha \in \mathbb{R}^+$ is a positive constant and $\alpha(0) = 1$. In order to represent arbitrary complex motions, the non-linear forcing term $f(\alpha)$ is usually modeled by a LWPR model as in (2.33), such that

$$f(\alpha) = \frac{\sum_{i=1}^{N} w_i(\boldsymbol{\alpha}) y_{*i}(\boldsymbol{\alpha})}{\sum_{i=1}^{N} w_i(\boldsymbol{\alpha})}\, \alpha\, (g - x(0))\,. \tag{2.36}$$

This formulation ensures stability of the overall dynamics (2.34) as the decay of the phase variable (2.35) monotonically converges to 0, thereby nullifying the only potentially unstable term (2.36). This non-linearity can be also modeled by any other analytical model or regression technique. Note that, although expressed as a differential equation, the non-linear term depends on a phase variable that acts as a temporal value. Once k_α is set in (2.35), the non-linear forcing term follows from a regression step in time domain.

DMPs have many desirable properties. The parameters of the LWPR term (2.36) are adjustable in a data-driven way so that the modulated dynamics are similar to a set of demonstrations [90]. Additionally, the parametric formulation of the goal g and the execution velocity given by τ enable goal-independent and velocity-independent encoding. This fact enhances the recognition and generalization capabilities of DMPs, which find inspiration in biological principles [91]. It is also remarkable that DMPs also modulate limit cycle attractors by defining the phase variable as a phase oscillator.

DMPs is a widely spread method in robotics, covering motor skill learning [92] with online adaptation [93] as well as periodic tasks learning [94] A recent exposition of DMPs and its applications is presented in [95].

2.2.3 Learning Cost Functions

Another alternative for characterizing human behavior assumes that human decisions follow the principle of rational action [96]. This implies efficiency in their decisions or, similarly, optimality w.r.t some performance measure or cost. If the *features* that determine the cost are known and observable, human demonstrations enable the autonomous extraction of the *relevance* of each of the features. This is the idea behind *inverse optimal control* also referred to as *inverse reinforcement learning* in the literature and initially proposed in [97]. Concerning definition 2.2, in this case the vector field $\boldsymbol{f}$ describes desired human behavior as the result of an optimization problem, i.e.

$$\min_{\boldsymbol{x}_d(t),\boldsymbol{u}_{\mathrm{env}d}(t)} J(\boldsymbol{z}) \,,$$

constrained to the system dynamics and where $J \in \mathbb{R}$ represents the learned cost function. Note that ξ contains the considered features, such as some dynamic properties or environmental or task-related parameters present in ξ_{env}.

Consider a discrete-time optimal control problem with known dynamics

$$\xi_{k+1} = \boldsymbol{f}(\xi_k, \boldsymbol{u}_k) \,, \tag{2.37}$$

where $\xi_k \in \mathbb{R}^n$ is the system's state and $\boldsymbol{u}_k \in \mathbb{R}^m$ is the control input. Let the vector of observed features at time step k be $\boldsymbol{z}(\xi_k, \boldsymbol{u}_k) \in \mathbb{R}^\nu$, compactly written $\boldsymbol{z}_k$. For simplicity of illustration, assume first a cost functional linear in the features in the form

$$J(\boldsymbol{\theta}) = \sum_{k=0}^{\infty} \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{z}_k \,, \tag{2.38}$$

where $\boldsymbol{\theta} \in \mathbb{R}^\nu$ are unknown weighting factors. Given a set of E optimal expert demonstrations in term of feature trajectories, i.e. $\overline{\Xi} = \left\{\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots \boldsymbol{z}_{T_e}\right\}_{e=1}^{E}$, the empirical estimate of feature counts is

$$J_E = \frac{1}{E} \sum_{e=1}^{E} \sum_{k=1}^{T_e} \boldsymbol{z}_k \,. \tag{2.39}$$

The inverse optimal control problem aims for the weights $\hat{\boldsymbol{\theta}}$ for which the solution to the optimal control constrained to dynamics (2.37) yield similar number of feature counts as the expert demonstrations, i.e. $J(\hat{\boldsymbol{\theta}}) \approx J_E$. This initial problem formulation from [97] is further extended increasing efficiency in [98] and considering probabilistic policies in [99]. In terms of definition 2.2, the predicted human desired behavior for a finite horizon of T steps consist of the trajectories $\boldsymbol{x}_d$ and $\boldsymbol{u}_{\mathrm{env}d}$ resulting from optimization

$$\begin{aligned} &\min_{\boldsymbol{u}_{0\cdots T-1}} J(\hat{\boldsymbol{\theta}}) \\ &s.t.\ \xi_{k+1} = \boldsymbol{f}(\xi_k, \boldsymbol{u}_k) \,. \end{aligned}$$

Understanding the decision-making process as a causal process of states and controls, the maximum causal entropy method [100] applied to inverse reinforcement learning [101] is a specially appealing alternative as it does not assume optimal human demonstrations. This solution aims for the controls that explain the demonstrated decisions with maximum entropy, i.e. the least committed or most fair solution that do not favor any possibility. This approach is extended in [102] to locally optimal problems with continuous states and actions spaces and in [103] adding a non-linear term to cost function (2.38) such that

$$J(\boldsymbol{g}, \boldsymbol{\theta}) = \sum_{k=0}^{\infty} \boldsymbol{g}(\xi_k) + \boldsymbol{\theta}^{\mathrm{T}} \xi_k \,, \tag{2.40}$$

where $\boldsymbol{g}(\xi_k) \sim \mathcal{GP}(m(\xi), k(\xi, \xi'))$. It is remarkable that this cost is a random variable due to the Gaussian Process representing the additional non-linear term. Although we do not explicitly explore inverse optimal control approaches in this thesis, their future application in physical assistance has great potential. In addition, the methods that we present on Chapter 4 consider uncertainty-dependent control solutions. This methods are also useful for learned cost functions such as (2.40), where the learned weights are uncertain and uncertainty has an important influence on decisions.

2.2.4 Discussion

The recent literature proposes several alternatives for encoding human behavior from demonstrations. The assumptions, generalization capabilities and the required knowledge of the methodologies reviewed in this section are presented in Table 2.1. Based on different assumptions, these models aim for an accurate representation of observed behavior, thereby acquiring spatial, temporal or environmental/task constraints.

	Assumptions	**Generalization**	**Req. Knowledge**
Time-series models (Sec. 2.2.1)	none	–	none
State-space dynamics as in (2.25) (Sec. 2.2.2)	none	+	none
Goal-oriented dynamical systems (DMPs and [77])	goal-orientation	+	goal
Inverse Optimal Control (Sec. 2.2.3)	optimality	++	features

Tabular 2.1: Properties of human behavior modeling approaches in terms of the underlying assumptions, their generalization capabilities and the a priori required knowledge.

Dynamical systems in the form (2.25) and time-series models present no underlying assumptions further than their inherent dynamic properties; these models only describe the evolution of state variables w.r.t time. DMPs assume that behavior is goal-oriented, i.e. any encoded behavior aims for a specific configuration. In contrast, Inverse Optimal Control assumes the principle of rational action or the principle of maximum entropy, i.e. actions are explained as the maximization of a performance measure comprising specific features. This strategy has been proven efficient representing human behavior [104]. However, its application in arbitrary settings must be studied in depth a priori: modeled behavior can always be represented as the minimization of some convex functional but its correspondence to the real inherent properties of the system might be vague.

The generalization capabilities of time-series models are rather limited due to their representation as a trajectory of statistical means and variances. Any prediction of a time-series model is limited to a sample of a single trajectory. As a result, temporal constraints are well represented but the acquisition of spatial constraints is unsatisfactory. This becomes evident in navigation tasks comprising multiple homotopy classes, i.e. bifurcations. In contrast, as dynamical systems model the state-dependent vector field $\boldsymbol{f}$, their predictions are well-suited for encoding such spatial constraints. It is however uncertain how the system behaves in state-space regions with missing or few observations. This issue is critical for later application in control. DMPs generalize motions in a goal- and velocity-independent way as a function of time. As a result, spatial constraints further than the desired goal are again not well represented. Learned cost functions generalize observed behavior to situations that can be described well in terms of the considered features. If features are selected properly they achieve superior performance. Note that goal-oriented behaviors are similarly encoded in cost functions by considering the goal one of the features.

Concerning the a priori required knowledge, besides observed configuration trajectories, goal-oriented dynamical systems require knowing the goal. Although given a demonstration the goal can be assumed to be the last state sample, in some settings this might not hold. The main impediment for learning cost functions is the selection of the required features that determine the performance measure. This is in fact a critical issue, which requires selecting potential relevant features that determine the optimization criterion of humans. This challenging task requires expert knowledge.

In this thesis we restrict our study to naive models with no a priori premises concerning the task. We will present time-series and dynamical systems models for learning human behavior models during interaction. No assumptions requiring additional a priori knowledge, such as a knowing a final goal or relevant environment-related features are made.

2.3 Stochastic Optimal Control Theory

Optimal control plays a key role in modern control design of cyber-physical systems. The flexibility of decision-makers that respond to the minimization of a cost functional for modulating complex dynamics is especially appealing for robotic systems. For later convenience, we expose this problem for fully observed discrete-time settings. Consider a system governed by the following stochastic difference equation

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{f}(\boldsymbol{\xi}_k, \boldsymbol{u}_k, \boldsymbol{\epsilon}_k) \;, \tag{2.41}$$

where $\boldsymbol{\xi} \in \mathbb{R}^n$ is the system's state, $\boldsymbol{u} \in \mathbb{R}^m$ the control input and k the sample time; $\boldsymbol{\epsilon} \in \mathbb{R}^q$ is an independent identically distributed (i.i.d.) Gaussian random variable with zero mean defined in the probability space given by the triple $(\Omega, \mathscr{F}, \mathsf{P})$ where $\Omega = \mathbb{R}^q$ is the sample space, $\mathscr{F}$ the σ-algebra and P the q-dimensional Gaussian measure. It follows that

$$\mathsf{E}[\boldsymbol{\epsilon}_k] = \mathbf{0} \qquad \mathsf{E}[\boldsymbol{\epsilon}_k \boldsymbol{\epsilon}_k^{\mathrm{T}}] = \Sigma_k \;,$$

where $\Sigma_k \in \mathbb{R}^{q \times q}$ is the covariance matrix and $\mathsf{E}_{\mathsf{P}}[f(\boldsymbol{\epsilon})]$ is the expectation operation, given by

$$\mathsf{E}[f(\boldsymbol{\epsilon})] = \int_{\Omega} f(\boldsymbol{\epsilon}) d\, \mathsf{P}(\boldsymbol{\epsilon}) \; .$$

To evaluate a a candidate control input sequence or *policy*, $\boldsymbol{u}_{0 \cdots T-1}$ for a finite horizon of T steps and initial state ξ_0, the performance measure is given by the cost functional

$$J(\xi_0, \boldsymbol{u}_{0 \cdots T-1}) = h_T(\xi_T) + \sum_{k=0}^{T-1} h_k(\xi_k, \boldsymbol{u}_k) \; , \tag{2.42}$$

where $h_T(\xi_T)$ is the *final cost* and $h_k(\xi_k, \boldsymbol{u}_k)$ is the *running cost*.

Due to the Gaussian input $\boldsymbol{\epsilon}_k$ present in system dynamics (2.41), $\xi_{1 \cdots T}$ is Gaussian and therefore performance measure (2.42) is also a random variable. In order to evaluate J, a deterministic measure is necessary. A straightforward solution is the expected value, i.e. $\mathsf{E}[J]$. In this case the stochastic optimal control problem is defined as

$$\begin{aligned} &\min_{\boldsymbol{u}_{0 \cdots T-1}} \mathsf{E}[J(\xi_0, \boldsymbol{u}_{0 \cdots T-1})] \\ &\text{s.t.} \quad \xi_{k+1} = \boldsymbol{f}(\xi_k, \boldsymbol{u}_k, \boldsymbol{\epsilon}_k) \; . \end{aligned} \tag{2.43}$$

The following section exposes the method of dynamic programming in order to solve this problem. Section 2.3.2 present alternatives to problem (2.43) that consider different statistical measures instead of the expectation. The convenient and special case of linear plant and quadratic cost is exposed in Section 2.3.3.

2.3.1 Dynamic Programming

Dynamic programming is an efficient method for solving discrete-time optimal control problems such as (2.43). It follows from the principle of optimality [105], which states

> An optimal policy has the property that whatever the initial state and decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

The application of this principle in our setting (2.43), yields the *Bellman equation*, which solves the problem recursively, backwards in time, for $k = T - 1 \cdots 0$ as

$$\begin{aligned} J(\xi_k, \boldsymbol{u}_k) &= \mathsf{E}[h_k(\xi_k, \boldsymbol{u}_k) + V(\xi_{k+1})] \\ V(\xi_k) &= \min_{\boldsymbol{u}_k} J(\xi_k, \boldsymbol{u}_k) \; , \end{aligned} \tag{2.44}$$

where $V(\xi_k)$ is the *value function* representing the optimal cost of the remaining decisions.

Throughout this thesis we will rely on the Bellman equation in order to solve discrete-time optimal control problems. However, in contrast to (2.44) which considers the expectation we will analyze different measures of the random cost as exposed in the following Subsection.

Note that optimal control problems in continuous-time systems are solved by means of Pontryagin's minimum principle, yielding a nonlinear two-point boundary value problem or by means of the Hamilton-Jacobi-Bellman equations, which are the equivalent to (2.44) for continuous-time systems. Another alternative is to compute a discrete-time approximation of the continuous-time problem and apply (2.44) directly. This thesis relies on this approach, approximating continuous non-linear systems with arbitrary costs to linear quadratic settings as later exposed in Section 2.3.3.

2.3.2 Considering High-Order Cost Statistics

In some applications, problem (2.43) neglects valuable information present on the random cost J but ignored in its expected value $\mathsf{E}[J]$. As an example consider a navigation problem where, on expectation, following two different paths yields the same cost. However, the cost variance is significantly different. A solution from problem (2.43) shows no preferences for any path. However, choosing the path with lower variance ensures more robust performance. Accounting for higher level statistical measures significantly increases robustness in uncertain settings. Here, we review risk-sensitive control and cost-cumulant control, two closely-related optimal control methods that account for high-order cost statistics. These methods are instrumental in optimal robot control problems that rely on learned models; the inherent uncertainty of probabilistic or statistical models exposed in Section 2.2 can be a decisive performance factor for successful task execution. Following this idea, we present novel control design approaches based on learned models in Chapter 4.

Risk-sensitive Optimal Control

Risk-sensitive control is an suitable alternative to (2.43), as it considers the problem

$$\begin{aligned} &\min_{\boldsymbol{u}_{0\cdots T-1}} \; \theta^{-1}\Psi(\theta) \\ &s.t. \quad \xi_{k+1} = \boldsymbol{f}(\xi_k, \boldsymbol{u}_k, \boldsymbol{\epsilon}_k)\,, \end{aligned} \tag{2.45}$$

where $\Psi(\theta) = \log \mathsf{E}[\exp\{\theta J(\xi_0, \boldsymbol{u}_{0\cdots T-1})\}]$ and θ is the risk-sensitivity parameter. The Taylor series expansion of the functional $\theta^{-1}\Psi(\theta)$ around $\theta = 0$ is

$$\sum_{r=1}^{\infty} \frac{\theta^r}{r!} \frac{\partial^r \Psi(\theta)}{\partial \theta^r}\bigg|_{\theta=0} = \mathsf{E}[J] - \frac{1}{2}\theta\, \mathsf{Var}[J] + \ldots\,, \tag{2.46}$$

which shows the influence of high-order statistical measures of the cost. For $\theta \approx 0$ any higher order terms of the expansion can be neglected yielding a mean-variance optimizer. The influence of the value of θ becomes apparent. A positive value, $\theta > 0$ yield *risk-averse* solutions, i.e. cost variability is interpreted as an additional penalty. In contrast,a negative value $\theta < 0$ corresponds to a *risk-seeking* solution, which considers cost variability as a discounting value. The *risk-neutral* case, $\theta = 0$ neglects higher order statistical cost terms and recovers the expected value problem (2.43).

Risk-sensitive optimal control is first introduced by Jacobson [106] and later further studied by Whittle [107] in discrete-time settings with partial state information. Continuous-time solutions are studied first in [108]. It is remarkable that risk-sensitive control is closely related to many other control paradigms. An overview of its relationships with differential game theory, H-infinity control, entropy minimization among others can be found in [109]. Very few applications of risk-sensitive control are found in robotics. Recent works applied this concept to stochastic cost functions learned with Gaussian Processes [110] as well as in the context of approximate inference control [111, 112].

Cost-cumulant Control

Cost-cumulant control is closely related risk-sensitive control. Cumulants are computed by means of the *cumulant generating function*

$$\Psi(\theta) = \log \mathsf{E}[\exp\{\theta J(\boldsymbol{x}_0, \boldsymbol{u}_{0\cdots T-1})\}] .$$

The r-th cumulant is given by

$$\kappa^{(r)}(J) = \left.\frac{\partial^r \Psi(\theta)}{\partial \theta^r}\right|_{\theta=0} .$$

It is interesting to observe the correspondence of cumulants with common statistical measures. The first and second cumulants represent the expected value and the variance respectively, i.e. $\kappa^{(1)}(J) = \mathsf{E}[J]$ and $\kappa^{(2)}(J) = \mathsf{Var}[J]$. The third and fourth cumulants are related to the *skewness* and *kurtosis* of J respectively.

The cost-cumulant control problem minimizes an arbitrary weighted sum of $K \in \mathbb{Z}^+$ cumulants, i.e.

$$\begin{aligned} &\min_{\boldsymbol{u}_{0\cdots T-1}} \sum_{r=1}^{K} \theta_r \kappa^{(r)}(J) \\ &s.t. \quad \xi_{k+1} = \boldsymbol{f}(\xi_k, \boldsymbol{u}_k, \boldsymbol{\epsilon}_k) , \end{aligned} \tag{2.47}$$

The relation with risk-sensitive control is now evident observing (2.46) and (2.47): problem (2.45) considers an infinite sum of cost cumulants with specific weighting factors given by the McLaurin coefficients divided by θ. Hence, cost-cumulant control is a generalization of risk-sensitive control as it enables the consideration of arbitrary weighting factors.

Cost-cumulant control was first presented in [113], studying mean-variance decision-makers. Further results are given in [114], referring to its relationship with risk-sensitive control. State-feedback solutions are further presented in [115]. An overview of current results for continuous-time linear quadratic settings is found in [116] and for the discrete-time case in [117].

2.3.3 Linear Quadratic Problems

Optimal control solutions do not usually have closed form and must be computed by means of numerical solutions. However, a specially convenient exception is the linear quadratic setting, which yield closed-form state-feedback solutions. In this case dynamics are given by the stochastic difference equation

$$\xi_{k+1} = A_k \xi_k + B_k \boldsymbol{u}_k + \Gamma_k \boldsymbol{\epsilon}_k \quad ,$$

where $A_k \in \mathbb{R}^{n\times n}$, $B_k \in \mathbb{R}^{n\times m}$ and $\Gamma_k \in \mathbb{R}^{n\times q}$ is a semi-orthogonal matrix.

Given an initial state $\boldsymbol{\xi}_0$ the cost follows quadratic form

$$J(\boldsymbol{\xi}_{1\cdots T}, \boldsymbol{u}_{1\cdots T-1}) = \frac{1}{2}\boldsymbol{\xi}_T^{\mathrm{T}} Q_T \boldsymbol{\xi}_T + \sum_{k=0}^{T-1} \frac{1}{2}\boldsymbol{\xi}_k^{\mathrm{T}} Q_k \boldsymbol{\xi}_k + \frac{1}{2}\boldsymbol{u}_k^{\mathrm{T}} R_k \boldsymbol{u}_k \ ,$$

where $Q_k \in \mathbb{R}^{n\times n}$ and $R_k \in \mathbb{R}^{m\times m}$ are positive semi-definite and positive definite matrices respectively.

This particular regulation problem has closed-form solutions yielding optimal state-feedback policies in the form

$$\boldsymbol{u}_k = L_k \boldsymbol{x}_k \ ,$$

where $L_k \in \mathbb{R}^{m\times n}$ is the optimal feedback matrix.

The solution to these problems for the risk-sensitive and cost-cumulant control cases are found in [107] and [118] respectively. However, the application to non-linear systems with arbitrary costs is still a widely unexplored issue, especially in robotic settings. In this thesis we explore a more general family of LQ problems that enable the computation of locally optimal risk-sensitive and cost-cumulant solutions for non-linear systems with arbitrary costs. Following the idea behind differential dynamic programming [119], in Chapter 4, we discretize and locally approximate an initial nominal trajectory as an LQ problem. Solutions of the local problem serve to iteratively update the nominal solution until convergence. Unlike many other approximation methods, the presented approach yields state-feedback control laws increasing robustness under disturbances during runtime.

3

Learning Behavior Models during Interaction

The acquisition of human behavior models based on motion and force/torque signals is an instrumental step in the design of physical robotic partners. Due to the complexity of human decision-making, observed behavior often represent complex nonlinear dynamics. Regarding its application to control in pHRI, it is desirable that a human behavior model exhibits the following properties:

- *Smooth predictions*. Prediction discontinuities are undesirable for later control as they may render abrupt robotic behavior and disturb the human partner. Traditional HMMs for time series are limited in this concern as their latent state space is discrete. Dynamical systems offer better performance due to the inherent smoothness of continuously differentiable differential equations.

- *Fast computation time*. As humans may act unexpectedly any time, fast re-computation of predictions is necessary for intuitive interaction. Models requiring high computational complexity are undesirable.

- *Conservative or 'safe' predictions*. When unexpected actions occur, it is desirable that predictions remain close to training data, even in previously unvisited regions of the observation space. Learned state-space dynamical systems models are especially unsuitable concerning this issue as predictions in unknown regions may yield unstable behavior [74].

This chapter presents model acquisition algorithms for learning both time-series and dynamical system models during interaction accounting for these properties. In Section 3.1 we present efficient and smooth prediction techniques for time-based HMMs. In order to enable incremental learning with time-series models, we explore the application of automated segmentation and clustering techniques. Evaluations demonstrate increasing prediction performance during interaction. Section 3.2 presents parametric time-based HMMs, where environmental information and natural language are modeled as latent variables that determine variations on learned behaviors. As shown by the results, speech commands or environmental configurations modify the shape and/or duration of encoded motions. As an alternative to time-series models, the online update capabilities of GPs in order to incrementally learn state-space models of observed tasks are studied in Section 3.3. We introduce impedance-based covariance functions in order to exploit a priori knowledge of human behavior. Experiments show increased prediction performance with respect to naive models.

3.1 Modeling Human Behavior with Time-based HMMs

As introduced in Section 2.2, HMMs provide low computational complexity and flexible recognition capabilities due to their discrete latent space. Modeling time series, left-to-right HMM are the most appropriate choice as they simulate the inherent flow of time constraining the transition matrix to an upper diagonal form. In such cases, each state intuitively represents a number of consecutive samples of the time series for which the corresponding probability distribution remains constant. Predictions are inherently conservative as they are limited to the expected training data average corresponding to a latent state. However, regression over time is limited due to the lack of dynamics during each particular latent state yielding non-smooth predictions. To cope with this issue, the trajectory-HMM [69] reformulates the HMM as a trajectory model, increasing its state space to each of the samples of a trajectory of means and covariances. As a trade-off, this extended formulation lacks the recognition flexibility of a standard HMM's latent state space, especially effective for observations sequences with different execution speed. In this section, we study the extension of the classical model explicitly learning the time correlation [70] in a similar manner to [63] with GMMs. The proposed model provides smoother predictions while maintaining the low computational complexity and high recognition flexibility of the standard HMM.

The rest of this Section is structured as follows. Section 3.1.1 introduces time-based HMMs and efficient regression techniques. A valid incremental learning framework is presented in Section 3.1.2. The proposed method is evaluated in an experiment with human users in Section 3.1.3.

3.1.1 Time-based HMMs

As presented in Section 2.2.1, an HMM with $S \in \mathbb{Z}^+$ states is given by the tuple $\lambda = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{W}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, where

- $\boldsymbol{\pi} = \{\pi_i\}$ with $\pi_i \in [0,1]$ are the initial state probabilities, such that $\forall i \in \mathbb{S} \quad \mathsf{P}(q_0 = i) = \pi_i$, where $\mathbb{S} = \{1, 2 \cdots S\}$ is the set of possible states.
- $A = \{a_{ij}\}$ with $a_{ij} \in [0,1]$ is the transition probability matrix describing the latent space dynamics such that $\forall i, j \in \mathbb{S} \quad \mathsf{P}(q_{t+1} = j | q_t = i) = a_{ij}$.

- $\boldsymbol{W} = \{w_{ik}\}, \boldsymbol{\mu} = \{\boldsymbol{\mu}_{ik}\}, \boldsymbol{\Sigma} = \{\Sigma_{ik}\}$ are the weight, mean and covariance matrix of the k-th mixture component of i-th state such that $\forall i \in \mathbb{S} \quad \mathsf{P}(\boldsymbol{\xi}|q_t = i) = \sum_{k=1}^{K} w_{ik}\mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu}_{ik}, \Sigma_{ik})$.

A time-based HMM consists of a left-to-right HMM defined over the spatio-temporal observation space $\boldsymbol{\xi} = \left[\boldsymbol{\xi}_s^{\mathrm{T}}\ \xi_t\right]^{\mathrm{T}}$, where $\boldsymbol{\xi}_s \in \mathbb{R}^{m-1}$ is the spatial vector given in this case by motion and wrench components and $\xi_t \in \mathbb{R}$ is its corresponding temporal value. The transition matrix is constrained to the form

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1S} \\ 0 & \ddots & & \vdots \\ \vdots & & & \\ 0 & \cdots & 0 & a_{SS} \end{bmatrix}, \tag{3.1}$$

where $a_{SS} = 1$ neglects any dynamics at the *final state*. For later convenience, the k-th component of the i-th state is normally distributed $\mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu}_{ik}, \Sigma_{ik})$ and the corresponding spatial an temporal statistics are denoted

$$\mathcal{N}\left(\begin{bmatrix} \boldsymbol{\xi}_s \\ \xi_t \end{bmatrix} \middle| \begin{bmatrix} \boldsymbol{\mu}_{sik} \\ \mu_{tik} \end{bmatrix}, \begin{bmatrix} \Sigma_{ssik} & \Sigma_{stik} \\ \Sigma_{tsik} & \Sigma_{ttik} \end{bmatrix}\right).$$

Training

Given a set of $E \in \mathbb{Z}^+$ observations *sequences* where the e-th observation is given by $O_e = \{\boldsymbol{\xi}(1), \boldsymbol{\xi}(2), \cdots \boldsymbol{\xi}(T_e)\}$, the Baum-Welch algorithm [62] computes iteratively the HMM parameters λ that maximize the sequences' likelihood. However, this standard training procedure overestimates temporal statistics, which strictly depend on the length of each observation sequence T_e. To cope with this issue, we modify the maximization step of the Baum-Welch algorithm regarding temporal statistics as

$$\mu_{tik} = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T^e} \gamma_{ik}^e(t)\xi_t(t)}{\sum_{e=1}^{E}\sum_{t=1}^{T^e} \gamma_{ik}^e(t)}$$

$$\Sigma_{ttik} = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T^e} \gamma_{ik}^e(t)(\xi_t(t) - \mu_{t\,ik}^{\ e})^2}{\sum_{e=1}^{E}\sum_{t=1}^{T^e} \gamma_{ik}^e(t)} \tag{3.2}$$

$$\Sigma_{tsik} = \Sigma_{st\,ik}^{\ \mathrm{T}} = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T^e} \gamma_{ik}^e(t)(\xi_t(t) - \mu_{t\,ik}^{\ e})(\boldsymbol{\xi}_s^e(t) - \boldsymbol{\mu}_{sik})}{\sum_{e=1}^{E}\sum_{t=1}^{T^e} \gamma_{ik}^e(t)}, \tag{3.3}$$

where $\gamma_{ik}^{e}(t)$ is the probability of being at state i at time t with the k-th mixture component accounting only for the e-th spatial observation sequence and

$$\mu_{t\,ik}^{\;e} = \frac{\sum_{t=1}^{T^e} \gamma_{ik}^{e}(t)\xi_t(t)}{\sum_{t=1}^{T^e} \gamma_{ik}^{e}(t)} \quad .$$

It is remarkable that both (3.2) and (3.3) define covariances with respect to the auxiliary time mean $\mu_{t\,ik}^{\;e}$, which considers only the e-th observation sequence. Temporal statistics are computed depending on each observation's length thereby enabling learning in datasets with demonstrations of variable length. As a result no preprocessing step in order to align observations such as Dynamic Time Warping (DTW) is needed.

Prediction

Standard HMMs perform poorly as generative models for time-series in continuous observation spaces. The causes of their limited predictive capability become apparent observing their state-space-like form given by

$$\begin{aligned} q_{t+1} &= f_A(q_t) \\ \xi &= \mathsf{P}(\xi | q_t) \, , \end{aligned}$$

where $q_t \in \mathbb{S}$ is the latent state at time t and f_A represents the discrete latent space Markovian dynamics from (3.1). Predictions relying on this model are given by

$$\mathsf{E}[\mathsf{P}(\xi | q_t = i)] = \sum_{k=1}^{K} w_{ik} \boldsymbol{\mu}_{ik} \, ,$$

where $q_t = i$ is the estimated latent state. Hence, predicted values are limited to the expected observation of one of the latent states and accuracy is determined by the number of states.

As an alternative, we exploit the additional time information considered in time-based HMMs and aim for a generative model in the form

$$\begin{aligned} q_{t+1} &= f_A(q_t) \\ \xi_s &= \mathsf{P}(\xi_s | \, \mathsf{E}[\xi_t | q_t, q_{t-1}, \cdots, q_{t-l}]) \, , \end{aligned} \tag{3.4}$$

where $l \in \mathbb{Z}^+$. Predictions depend now on the temporal value ξ_t, which acts as an alternative latent space providing more accurate predictions than the standard model. For a given ξ_t, expression (3.4) corresponds to a GMR algorithm introduced in Section 2.2.1 yielding $\mathcal{N}(\xi_s | \hat{\boldsymbol{\mu}}_s, \hat{\Sigma}_{ss})$, where

$$\hat{\boldsymbol{\mu}}_s = \sum_{i=1}^{S} \sum_{k=1}^{K} \gamma_{ik}(\xi_t) \left(\boldsymbol{\mu}_{sik} + \frac{\Sigma_{tsik}}{\Sigma_{ttik}} (\xi_t - \mu_{tik}) \right). \tag{3.5}$$

and

$$\hat{\Sigma}_{ss} = \sum_{i=1}^{S} \sum_{k=1}^{K} \gamma_{ik}(\xi_t) \left(\Sigma_{ssik} - \frac{\Sigma_{stik}\Sigma_{tsik}}{\Sigma_{ttik}} \right), \tag{3.6}$$

where

$$\gamma_{ik}(\xi_t) = \frac{\mathcal{N}(\xi_t|\mu_{tik}, \Sigma_{ttik})}{\sum\limits_{n=1}^{S}\sum\limits_{j=1}^{K} \mathcal{N}(\xi_t|\mu_{tnj}, \Sigma_{ttnj})} \; . \tag{3.7}$$

If ξ_t is not available a priori, its expected value is estimated relying on the sequence of the last l latent states as

$$\mathsf{E}[\xi_t|q_t = i, q_{t-1}, \cdots, q_{t-l}] = \mathsf{E}[\xi_t|q_t = i] + \Delta n_{q_t} = \sum_{k=1}^{K} w_{ik}\mu_{tik} + \Delta n_{q_t} \; ,$$

where Δ is the sampling time representing the inherent time dynamics and n_{q_t} is the number of time steps spent of state i, i.e. the number of consecutive repetitions of state i in the latent state sequence. Note that this estimation procedure is straightforwardly extended to a given spatial observation sequences as

$$\mathsf{E}[\xi_t|\xi_s(t), \xi_s(t-1), \cdots, \xi_s(t-l)] = \mathsf{E}[\xi_t|q_t^*, q_{t-1}^*, \cdots, q_{t-l}^*] \; , \tag{3.8}$$

where $q_t^*, q_{t-1}^*, \cdots, q_{t-l}^*$ is the most likely latent state sequence, which can be efficiently computed by means of the Viterbi algorithm [61].

Similar to [63], the presented method enables also the computation of a generalized normal distribution trajectory representing the encoded time-series. This *generalized output* O_λ results from applying (3.5) and (3.6) for temporal values in the range $[\xi_t(0) = 0, \xi_t(T) = \Delta T]$, where T is the a priori unknown *expected length* in samples of the model. In order to approximate it, classical HMMs estimate the duration d_i of the i-th state relying on the implicit time information of transition probabilities by means of predictive distribution

$$d_i = \frac{1}{1 - a_{ii}} \; . \tag{3.9}$$

However, constraint (3.1) prevents from applying (3.9) at the final state due to the lack of dynamics, i.e. $a_{SS} = 1$. As an alternative, regarding the additional time information, the expected length of the model fulfills

$$T\Delta = \mu_{tS} + \Delta d_S/2 \; , \tag{3.10}$$

where $\mu_{tS} = \mathsf{E}[\xi_t|q_t = S] = \sum\limits_{k=1}^{K} w_{Sk}\mu_{tSk}$ and d_S is the unknown expected length of the final state. From (3.2) and assuming a discrete assignment of $\gamma_{ik}(t)$, i.e. 0 or 1, the temporal variance of state i is determined by the state duration d_i, fulfilling

$$\Delta^{-2}\mathsf{Var}[\xi_t|q_t = i] = \frac{2\sum\limits_{n=1}^{d_i/2} n^2}{d_i} = \frac{((d_i/2 + 1)(d_i + 1))}{6} \; , \tag{3.11}$$

where $\mathsf{Var}[\xi_t|q_t = i] = \sum\limits_{k=1}^{K} w_{ik}\Sigma_{ttik}$. Isolating d_i yields

$$d_i = \frac{-3 + \sqrt{1 + 48\Delta^{-2}\mathsf{Var}[\xi_t|q_t = i]}}{4} \; . \tag{3.12}$$

Applying this result to (3.10) yields T.

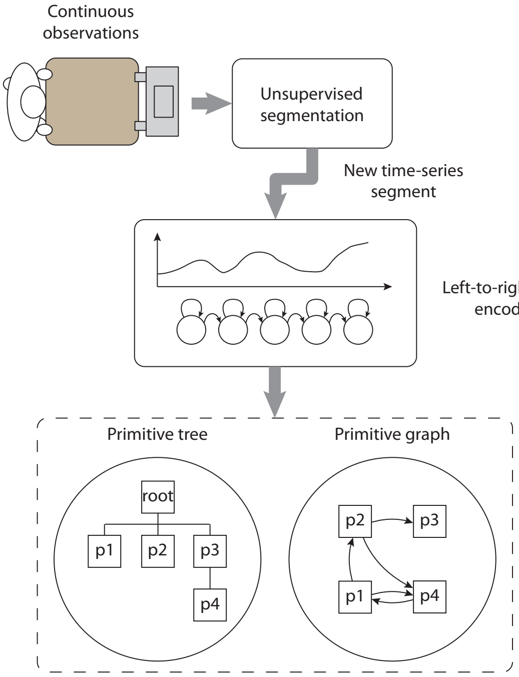


Figure 3.1: Primitive-based learning framework. The robot autonomously extracts primitives, hierarchically organized on the primitive tree and sequentially ordered in the primitive graph.

3.1.2 Incremental Learning

Learning during interaction requires a method that structures the continuous time-series observations coming from the robot sensors in a compact way. A primitive-based approach for time series provides a suitable way of organizing observations as it considers segments of time series as the minimal level of abstraction. Combining this concept with the inherent generalization capabilities of HMMs, an autonomous incremental learning process for our scenario can be developed. We base our approach on the methods explained in [65], briefly summarized in this subsection and shown in Fig. 3.1.

The autonomous extraction of behavior primitives requires first an unsupervised method that segments continuous time series into potential primitives. The segmentation algorithm tracks the probability density function assuming that data belonging to the same primitive will have the same underlying distribution. This criterion makes this method suitable for any kind of time series and not only haptic observations. For a detailed description we kindly refer the reader to [120, 121].

Each resulting segment is first encoded into a left-to right HMM λ as explained in Section 3.1.1. The encoded segments are organized forming a *primitive tree*, i.e. a hierarchical tree structure where each node of the tree represents a primitive. Having initially just a root node, each new segment is added as a member of the closest node of the tree. Once added, a clustering procedure is executed in the node. If a group of at least m members close enough is found, a new child of the node is created, being represented by an HMM trained with the observations of the selected members. Therefore, except for the root node, each node of the tree symbolizes a primitive and is represented by an HMM. The distance between two nodes,

λ_1 and λ_2 is calculated using a symmetric version of the Kullback-Leibler divergence given by

$$D(\lambda_1,\lambda_2)=\frac{1}{T}[\log P(O_{\lambda_2}|\lambda_1)-\log P(O_{\lambda_2}|\lambda_2)],$$

where O_{λ_2} is the generalized output of λ_2, T its length in number of samples and $P(O|\lambda)$ is the likelihood of observation O for model λ.

In addition to the primitive tree, the temporal relation between behavior primitives is also modeled as a directed graph, the *primitive graph*, where an edge between two nodes represents the probability to transit from the first node to the second one. This representation on a primitive level of abstraction provides a way to make long-term predictions, further away than a single primitive and a more reliable state estimation during primitive transitions.

The primitive tree and the primitive graph represent the acquired experience, constantly updated in order to provide an improving behavior model during interaction.

Prediction with sequential primitives

Let $O_c=\left\{\xi_s(t-l),\xi_s(t-l+1),\cdots\xi_s(t)\right\}$ be the last l observations of spatial data. The knowledge acquired by the robot is a structured set of time-based HMMs representing nodes of the primitive tree. In order to select the the model that better matches O_c a common approach is to select the most likely model $\lambda*$ given by

$$\lambda^*=\max_{\lambda_i}\mathrm{P}(O_c|\lambda_i)\ ,$$

$\forall\lambda_i\in$ Primitive tree. However, when O_c lies between two sequentially executed primitives, the most likely model might be the first one when the task execution has already reached the second primitive, leading to undesired model selections. To avoid such situations, we propose to use a *window HMM*: an auxiliary left-to-right HMM defined as a window over the states of sequentially executed models. This augmented model avoids the disadvantages of the segmented nature of primitives by ensuring that O_c is well represented by one of the sequentially connected models.

To define the window HMM, the Viterbi algorithm is first applied for λ^* in order to estimate the current state, yielding the center state of the window HMM. The rest of its states are symmetrically distributed along both sides of the current state. The expected predecessor and successor models are selected from the primitive graph and concatenated as shown in Fig. 3.2. The concatenation of left-to-right time-based HMM models is enabled by the availability of explicit temporal information. Let λ_1 and λ_2 be two left-to-right HMMs with N_1 and N_2 states respectively where λ_1 precedes λ_2. The concatenation of both models, λ_C, has N_1+N_2 states and is given by the parameter set of λ_1 and λ_2 adding a transition between the last state of λ_1 to the first state of λ_2 given by

$$\begin{aligned}a_{N_1,N_1+1}&=1-a_{N_1,N_1}\\a_{N_1,N_1}&=\frac{d_{N_1}-1}{d_{N_1}}\ ,\end{aligned}\qquad(3.13)$$

where d_{N_1} is the duration of the last state of λ_1 computed as in (3.12).

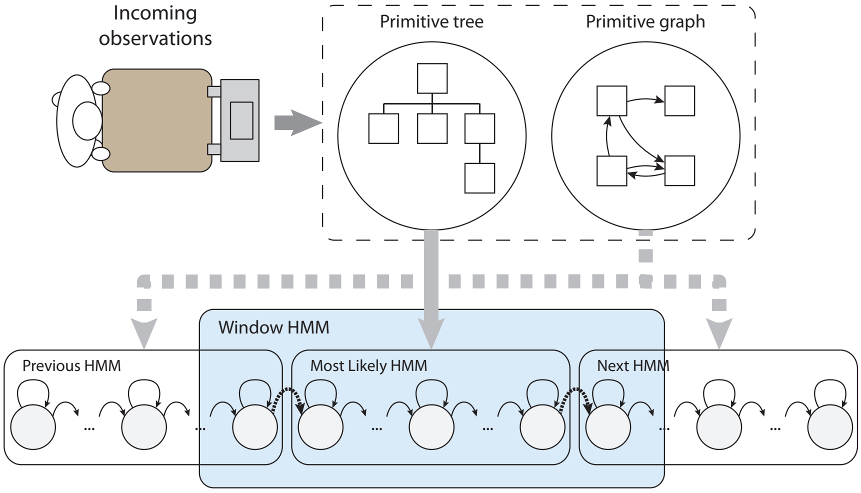


Figure 3.2: Window HMM. A window over the states of the involved models defines the window HMM for a more robust model selection.

Once the window model is defined, a prediction with an horizon of h seconds is straightforwardly computed applying (3.8), (3.5) and (3.6). Note that the definition of the window HMM implies no further computational complexity and is limited to (3.13).

3.1.3 Evaluation

In order to evaluate the applicability of the proposed framework, we designed an experiment where human users interacted physically with a simulated object in a two-dimensional scenario performing a repetitive task. Results validate the proposed approach, showing improved prediction performance as task repetitions increase.

Experimental setup

Using the two-dimensional setup from appendix A.1 implementing the simple admittance control scheme from (2.16) with mass 40 [kg] and damping 90 [Ns/m], we let 6 different subjects (all male, average age 28) perform a simple task consisting on following a path describing a flower, as shown in Fig. 3.3. The task was repeated cyclically 15 times with no pauses. The subjects had the opportunity to familiarize with the haptic device for a few minutes before the experiment and were told to follow the pre-defined trajectory for the given repetitions.

The spatial components of the time-based HMM's observation space are defined as $\boldsymbol{\xi}_s = \left[\boldsymbol{p}^{\mathrm{T}}\dot{\boldsymbol{p}}^{\mathrm{T}}\boldsymbol{f}^{\mathrm{T}}\right]^{\mathrm{T}}$ where $\boldsymbol{p}, \dot{\boldsymbol{p}}, \boldsymbol{f} \in \mathbb{R}^2$ are the position, velocity and applied force respectively. The incremental learning and prediction framework described in Section 3.1.2 is implemented in C++ and executed on a personal computer at 20 Hz. Each segment and each node is encoded in an HMM with 10 states and 1 Gaussian per state. In order to form a node in the primitive tree, a group of at least $m = 3$ members is needed and only the upper nodes of each branch are considered for the prediction. We use a window of $l = 30$ samples over the incoming observations for the primitive recognition and a window HMM with 20 latent

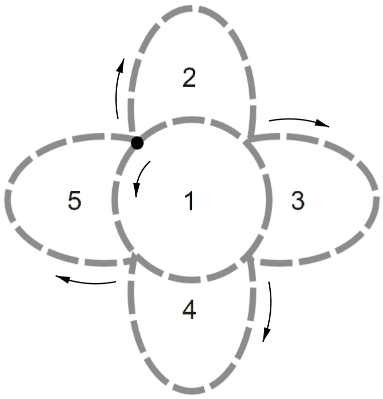


Figure 3.3: Experimental task. Starting on the black dot, each path segment is followed in the direction indicated by the arrows and in the order given by the numbers.

states. Predictions are computed for a horizon of $h = 0.3$ s. All parameters were chosen from experience.

Experimental Results

Fig. 3.4 shows the generalized output of the resulting HMM of the primitive representing the bottom petal for one of the subjects. The explicit time information used for the regression provides a smooth trajectory without applying any post processing. Once the robot learned all task primitives, the predicted future position for the given time horizon was very close to the real data after that time horizon, as shown by the dashed green line in Fig. 3.5 for one execution of the task by one of the subjects. In contrast, the red solid line shows the prediction performance without the window HMM. The five big jumps are caused by the primitive transitions, marked by the vertical dashed lines, where the most likely node is still the last one while the execution is already in the next primitive leading to wrong primitive selections. The online prediction performance is significantly improved by means of the window HMM, providing a smoother predicted trajectory over time.

Prediction errors were calculated as the difference between the predicted position for the given time horizon and the observed position after the time horizon, i.e.

$$||\boldsymbol{x}_{pred}(t) - \boldsymbol{x}(t+h)||.$$

The overall prediction performance is shown in Fig. 3.6. The mean prediction error per repetition starts decreasing after the fourth repetition of the task, reaching a minimum after 12 repetitions. As we set the minimum number of members for a node to 3, no nodes are created before the fourth execution, and therefore, no improvements are expected before. As shown by the big variance between repetitions 5 to 11, the prediction error evolved differently among the subjects. This difference relies on the performance of the unsupervised segmentation, which provides unexpected segmentation points if the user behaves abruptly and determines the learning and prediction results. The resulting tree of primitives was usually composed by as many branches as primitives the task had, in this case 5, given by the 4 petals and the center of the flower. For some users, some additional branches were the

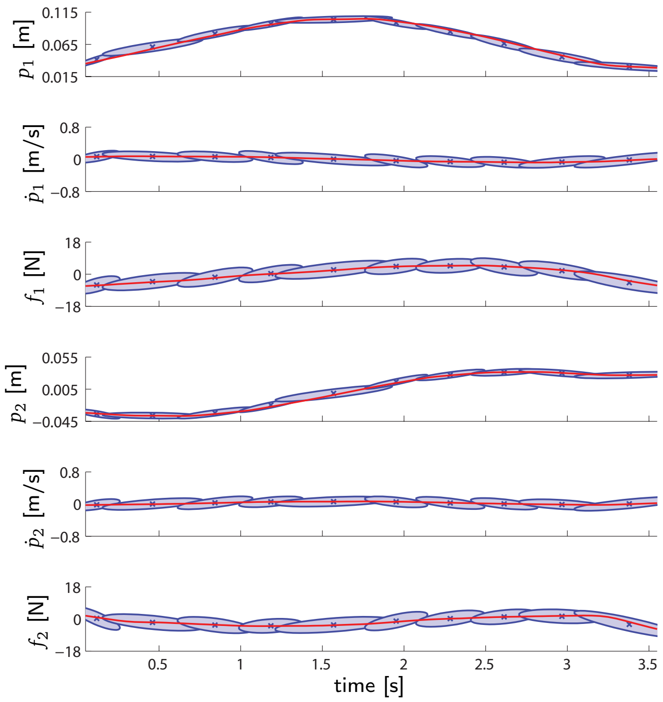


Figure 3.4: Exemplary generalized output O_λ of the time-based HMM representing the node of the bottom petal primitive for one of the subjects. The red solid line represents the expected mean trajectories. The ellipses represent the encoded spatio-temporal statistics: the eigenvalues of Σ_{stik} determine the ellipse's axis while its center is given by $(\mu_{tik}, \boldsymbol{\mu}_{ssik})$.

consequence of consecutive unexpected segmentation points. However, all task primitives were learned for every subject.

3.2 Learning Multi-modal Behavior with Parametric Time-based HMMs

In order for robots to interact with humans intuitively they must understand their partners on different levels of communication - verbally and non-verbally. This is particularly important for the cooperation of robots with humans under uncertainty where interactive planning, decision making, and control play a crucial role. Desirably a robot understands a verbal "command" by the human interaction partner and executes the corresponding motor action. In our problem setting, due to the physical coupling, both partners must agree on the path to follow as they navigate through the environment. In case of disagreement, humans can easily communicate with each other using natural language in order to find a consensus and proceed with the task execution. Synthesizing similar interactive behavior for robotic partners is however a very challenging problem as the information exchange between partners usually refers to environmental properties.

Humans easily interact with each other using motions and gestures [122, 123]. Natural language is a unique symbolic communication system to humans that provides much richer

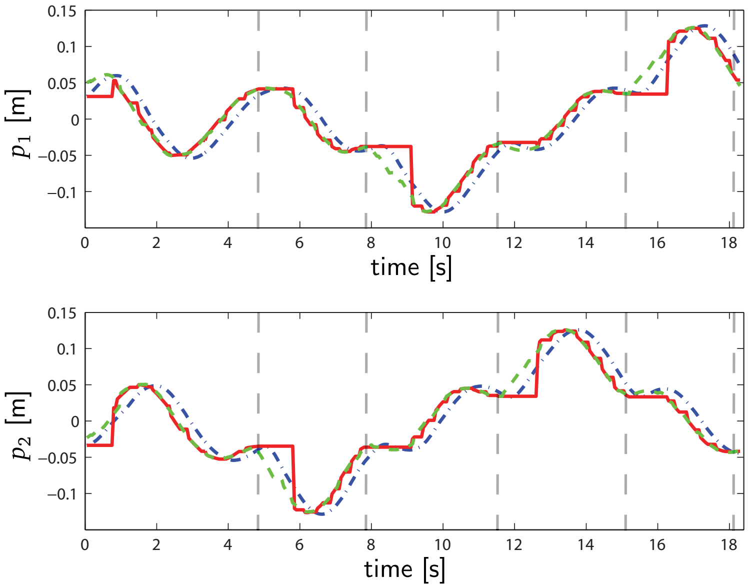


Figure 3.5: Prediction results. The blue dash dotted line represents the observed data, the green dashed line shows the prediction and the red solid line shows the prediction without both the window HMM and the velocity factor. The vertical dashed lines mark the transitions between the primitives.

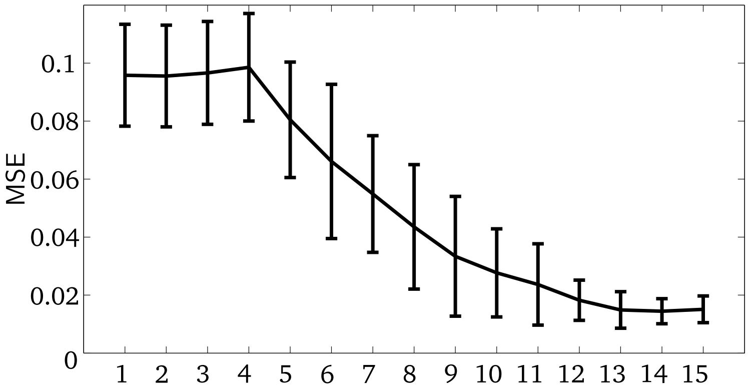


Figure 3.6: Users-average MSE of predicted positions per repetition.

descriptions even when referring to motions. More recently also the interaction through haptic signals as in physical human-robot cooperation is of increasing interest [124]. One of the major challenges is the combination of natural language with motor action in terms of haptic primitives with consideration for environmental and task constraints. Understanding the link between signifiers and their corresponding significants is a very challenging topic [125, 126], especially considering motions [127]. This problem usually implies, on one side, modeling sentence structures and word relations in a natural language model and, on the other side, considering the association between sentences and motion symbols [128, 129]. Still, this representation does not express the quality of behaviors in an explicit way. Many adverbs and adverbials specify how to parametrize and/or change motions or haptic primitives. As an example take the sentence “Turn slowly” where “turn” stands for a motion symbol and “slowly” expresses the quality of the motion primitive. Extracting such word models can deepen robots’ comprehensive capabilities when using natural language and improve their inference performance. Furthermore, motions, as well as their natural language descriptions, are usually influenced by environmental characteristics and constraints. Considering these extrinsic dependencies in conjunction with qualitative natural language descriptions will significantly expand robots' understanding capabilities and will potentially shrink the gap towards seamless HRI.

When behaviors are characterized by external variables, a natural extension of standard HMMs are Parametric HMMs (PHMMs), which include dependencies of the standard model parameters w.r.t external parameters [130]. PHMMs have been used to enrich motions representation by parameterizing its output densities [130] or by linearly interpolating exemplar parameters of the different states [131]. However, a full parameterization of all HMM parameters in a non-linear fashion is still an open issue.

In this section we address the problem of learning and executing motion primitives including quality descriptors and environmental constraints. To this end we propose a novel approach for learning relations and dependencies between motion symbols, natural language and environmental properties using parameterized left-to-right time-based HMMs. A natural language model together with a motion language model infers the link between sentences and motion symbols while the time-based HMMs parameterization represents the explicit influence on motions of both words and environmental properties. The proposed PHMM approach directly adapts both the output and the transition probabilities of time-based HMMs providing a full non-linear dependency estimation. The method is validated by learning and generating navigation primitives in a 2 DoF virtual scenario.

The remainder of this section is structured as follows. The proposed PHMM model is explained in Section 3.2.1 and the natural language model used is described in Section 3.2.2. Experimental results and discussion are presented in Section 3.2.3.

3.2.1 Parametric Time-based HMMs

A parameterized observation $O_\theta = \{O, \boldsymbol{\theta}\}$ is given by the observed spatial data sequence O and its corresponding parameters $\boldsymbol{\theta}$. Note that the parameters are constant during the whole observation, i.e., a static environment or command is assumed.

The parametric time-based HMM modifies the standard output densities and the transition probabilities from the standard model presented in Section 3.1 as a function of $\boldsymbol{\theta}$, i.e.

$$\begin{aligned} a_{ij}(\boldsymbol{\theta}) &= \overline{a_{ij}} + f_{a_{ij}}(\boldsymbol{\theta}) \\ \boldsymbol{\mu}_{ik}(\boldsymbol{\theta}) &= \overline{\boldsymbol{\mu}_{ik}} + f_{\boldsymbol{\mu}_{ik}}(\boldsymbol{\theta}) \\ \boldsymbol{\Sigma}_{\boldsymbol{ik}}(\boldsymbol{\theta}) &= \overline{\boldsymbol{\Sigma}_{ik}} + f_{\boldsymbol{\Sigma}_{ik}}(\boldsymbol{\theta}), \end{aligned} \tag{3.14}$$

where $\overline{a_{ij}}$, $\overline{\boldsymbol{\mu}_{ik}}$ and $\overline{\boldsymbol{\Sigma}_{ik}}$ are the standard transition probabilities, means and covariances, calculated using the EM algorithm [62] using only the spatial data O. Here we assume that the initial state probabilities are equal for all states and therefore no parameterization is needed, but a similar dependency could be easily added. The model's parametric dependency $\boldsymbol{f}(\boldsymbol{\theta}) = \{f_{a_{ij}}(\boldsymbol{\theta}), f_{\boldsymbol{\mu}_{ik}}(\boldsymbol{\theta}), f_{\boldsymbol{\Sigma}_{ik}}(\boldsymbol{\theta})\}$ is estimated observing the variations of the standard tHMM parameters with respect to $\boldsymbol{\theta}$.

Parametric Dependency Learning

In order to extract a non-linear approximation of the parametric dependency, we model the joint probability density of $\boldsymbol{f}(\boldsymbol{\theta})$ and the external parameters, i.e. $P(\boldsymbol{f}(\boldsymbol{\theta}), \boldsymbol{\theta})$, using a GMM. This probabilistic model expresses the probability distribution of the parameters θ in conjunction with the expected values of $\boldsymbol{f}(\boldsymbol{\theta})$. Given a set of D parametric observations $(d = 1...D)$, the joint distribution is estimated maximizing the likelihood of the GMM w.r.t the samples $\{\boldsymbol{f}(\boldsymbol{\theta_d}), \boldsymbol{\theta_d}\}$, where $\boldsymbol{f}(\boldsymbol{\theta_d})$ is given by

$$\begin{aligned} f_{a_{ij}}(\boldsymbol{\theta_d}) &= a_{ij}(\boldsymbol{\theta_d}) - \overline{a_{ij}} \\ f_{\boldsymbol{\mu}_{ik}}(\boldsymbol{\theta_d}) &= \boldsymbol{\mu}_{ik}(\boldsymbol{\theta_d}) - \overline{\boldsymbol{\mu}_{ik}} \\ f_{\boldsymbol{\Sigma}_{ik}}(\boldsymbol{\theta_d}) &= \boldsymbol{\Sigma}_{ik}(\boldsymbol{\theta_d}) - \overline{\boldsymbol{\Sigma}_{ik}}, \end{aligned} \tag{3.15}$$

and $a_{ij}(\boldsymbol{\theta_d})$, $\boldsymbol{\mu}_{ik}(\boldsymbol{\theta_d})$ and $\boldsymbol{\Sigma}_{ik}(\boldsymbol{\theta_d})$ are the transition probabilities, means and variances maximizing the likelihood only for the d-th observation.

A direct comparison of the standard model parameters with the observation specific ones requires an alignment of the states, i.e., a similar distribution of the responsibility of the states through all observations. In order to achieve this, we follow [131] initializing the values $a_{ij}(\boldsymbol{\theta_d})$, $\boldsymbol{\mu}_{ik}(\boldsymbol{\theta_d})$ and $\boldsymbol{\Sigma}_{ik}(\boldsymbol{\theta_d})$ with the overall ones $\overline{a_{ij}}$, $\overline{\boldsymbol{\mu}_{ik}}$ and $\overline{\boldsymbol{\Sigma}_{ik}}$ and then fixing the means after one iteration of the EM algorithm. This way, a similar distribution of the responsibility is extracted from the overall model validating the state comparisons from (3.15). The proposed approach is indeed similar to [131]. However, the inclusion of the GMM in order to model the parametric dependency provides better generalization properties due to its non-linearity and its regression capabilities in the whole parameter space.

Given the set of samples $\{\boldsymbol{f}(\boldsymbol{\theta_d}), \boldsymbol{\theta_d}\}$, the GMM is trained using the EM algorithm.

Generalized Parameter Model

While the influence of external parameters on a specific model is captured training a PHMM, a generalization of how those parameters affect motions can not be inferred with the parameterization of different primitives. In order to acquire a *generalized parameter model* the function $\boldsymbol{f}(\boldsymbol{\theta})$ must be learned maximizing the expected parameterization of all different motion primitives, as depicted in Fig. 3.7.

Given a set of M PHMMs $(m = 1...M)$ with D parametric observations $(d = 1...D)$ each, the generalized parameter model $\boldsymbol{f}(\boldsymbol{\theta})$ considers all $\{\boldsymbol{f}_m(\boldsymbol{\theta_d}), \boldsymbol{\theta_d}\}$ for $1 \leq m \leq M$ and $1 \leq d \leq D$, where $\boldsymbol{f}_m(\boldsymbol{\theta_d})$ is the expected value of the parametric function for the m-th PHMM and the d-th observation.

Inference and Motion Generation

Given a query with its corresponding parameters $\boldsymbol{\theta_q}$, generating the expected motion from a PHMM requires first the calculation of the expected values of $\boldsymbol{f}(\boldsymbol{\theta})$. Using Gaussian Mixture Regression (GMR) as presented in Section 2.2.1, we calculate the expected parameterization as the posterior mean estimate given by $\boldsymbol{\theta_q}$, i.e. $P(\boldsymbol{f}(\boldsymbol{\theta})|\boldsymbol{\theta_q})$. Applying then (3.14), the expected motion is generated calculating the generalized output of the resulting time-based HMM as explained in Section 3.1.

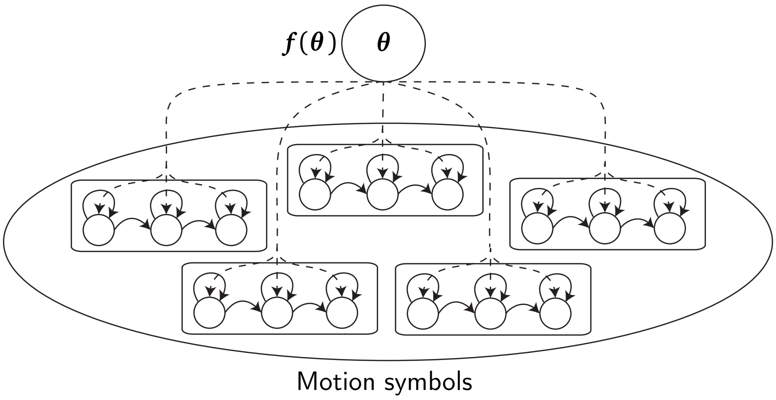


Figure 3.7: Generalized Parameter model. The non-linear dependency $f(\theta)$ is designed considering the parameterization of all possible motion symbols.

3.2.2 Motion and Natural Language Model

The connection between motions with natural language is achieved integrating two components: The motion language model and the natural language model. On one side, the natural language model represents stochastically all possible words sequences. On the other side, the motion language model is a stochastic model that captures the link between motion symbols and words. The combination of both methods provides a model that considers both the sentence structure constraints and the expected words and motion symbols links.

The natural language model stochastically represents sequences of words. A bigram model is used to represent links between words, modeling the expected sequences. Words are represented by nodes and transitions between two words, i.e. a word is followed by another one, as edges. This way, the observed sentence structures are expressed as transitions among words.

The motion language model represents the relations between motion symbols and words in a stochastic way. The model is structured in three different layers: motion symbols, latent states and words. Two different probabilities connect the motion symbols layer with the words layer. On one side the the connection between motion symbols and latent states and on the other side the link between latent states and words. The first connection represents the probability that a motion symbol generates the corresponding latent state while the second connection expresses the probability that a latent state generates a word. With this structure, the latent states layer symbolizes the association between motion symbols and words.

The motion language model and the natural language model are integrated by performing a computational search of sentences given a motion or for motions given a sentence. In this paper we focus only on the second direction. Furthermore, the usage of a bigram structure for the natural language model ensures a simple and efficient search appropriate for queries in both cases.

A more detailed explanation about both the motion language and the natural language model can be found in [132].

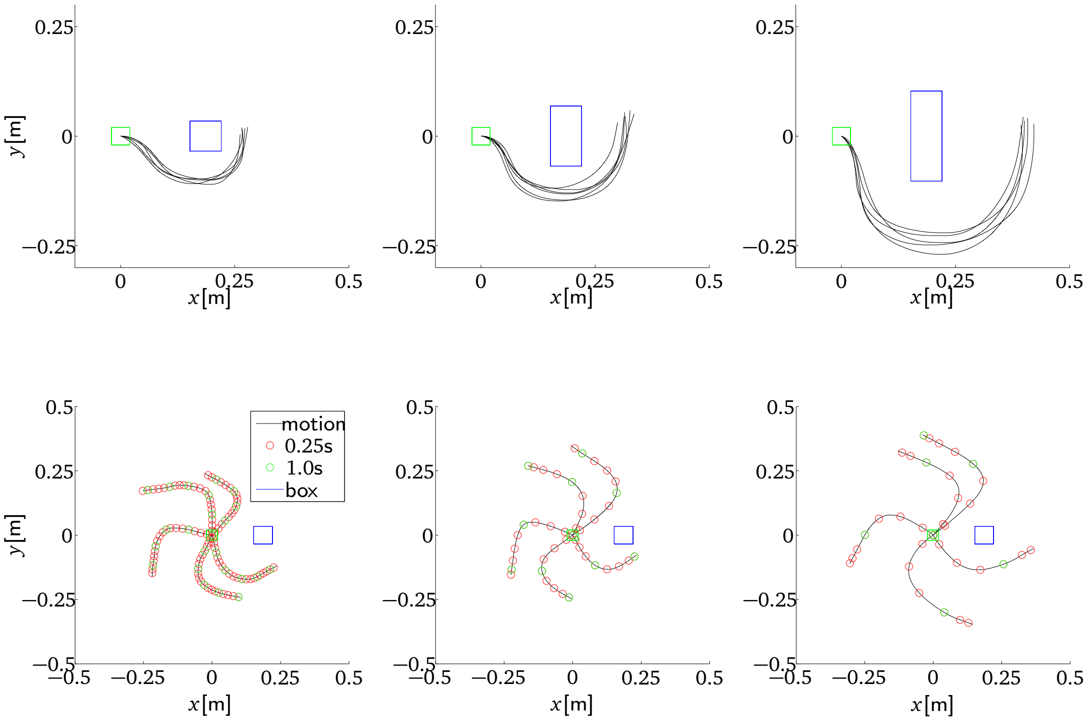


Figure 3.8: Generated dataset for the primitives "go behind the box right" for box sizes 1, 2 and 3 on top. On the bottom, generated motions for "turn left" performed in three different speeds: slow, normal and fast.

3.2.3 Evaluation

In order to test the potential applications of the proposed approach we acquired simple motion primitives labeled with adequate descriptions in a 2 DoF virtual scenario for a cooperative joint transportation task. We trained all different primitives using a single parametric dependency model, except for one primitive, used to test the applicability of the approach to unrelated motions. Using the descriptions and the learned PHMMs, we built both the corresponding natural language model and the motion language model. The presented results show the generalization properties of the system and its validity.

Experimental Setup

In a human-robot joint transportation task, due to the physical coupling, both partners must agree on the path to follow as they navigate through the environment. In case of disagreement, humans can easily communicate with each other using natural language in order to find a consensus and proceed with the task execution. However, finding an agreement implies the understanding of the description of motions through natural language. Such descriptions may also refer to environmental constraints. In this scenario, we designed a dataset in a simple environment in order to teach a robotic partner motion primitives with their respective description and environmental constraints that typically arise during this kind of interaction.

Using the two-dimensional setup from appendix A.1 implementing the simple admittance

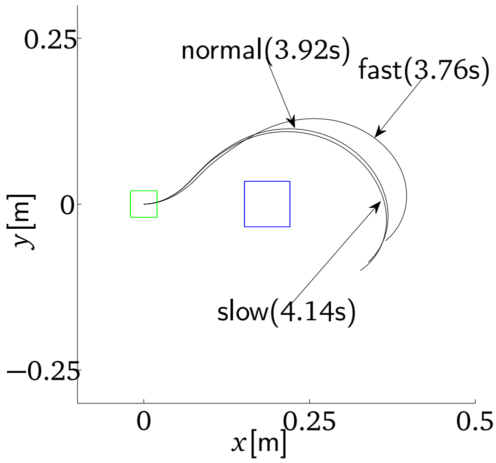


Figure 3.9: Resulting motions for the queries "go behind the box left", "go behind the box left slow" and "go behind the box fast".

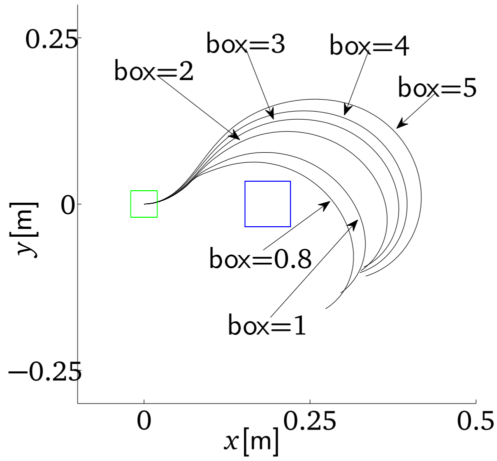


Figure 3.10: Generated motions for the queries "go behind the box left" with box sizes 0.8, 1, 2, 3, 4 and 5.

control scheme from (2.16), we generated samples for 7 different primitives: "go straight", "turn right", "turn left", "go to the right of the box", "go to the left of the box", "go behind the box left" and "go behind the box right". For each primitive we produced 25 motions, combining the standard description with the adverbs "fast" and "slow" as well as changing the size of the box displayed in the scene with three different sizes, i.e. big= 3, middle= 2 and small= 1. Several exemplary motions from the dataset for "go behind the box right" and "turn left" are shown in Fig. 3.8. The motion data captured in task space was encoded in an invariant way, which, in the case of pure translation without orientation in 2 dimensions reduces to both the curvature and the velocity profile [133].

Given the description of the motions, the dictionary used for this scenario contains 13 different words. The parameters for the observations are therefore determined by the word parameters $\boldsymbol{\theta}_l$, a vector of size 13, and the environmental ones $\boldsymbol{\theta}_e$, given by the size of the box displayed in the scene. Considering the 7 different motion symbols and the sentences

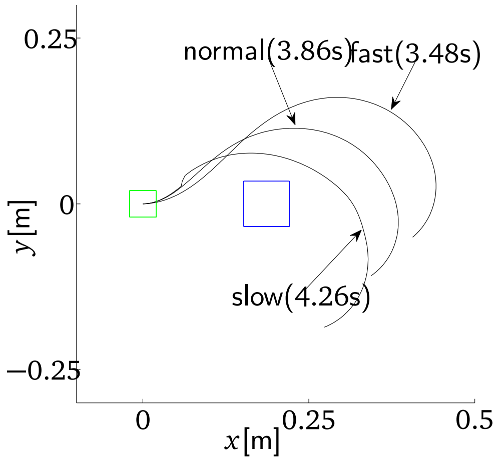


Figure 3.11: Generated motions for the generated parameterization of the marginal probabilities of "slow" and "fast" given by $P(\boldsymbol{f}(\boldsymbol{\theta})|\text{"fast"} = 1, \text{"slow"} = 0)$, $P(\boldsymbol{f}(\boldsymbol{\theta})|\text{"fast"} = 0, \text{"slow"} = 0)$ and $P(\boldsymbol{f}(\boldsymbol{\theta})|\text{"fast"} = 0, \text{"slow"} = 1)$ applied to the primitive "go behind the box left".

used for the descriptions, the natural language and the motion language model were built as explained in Section 3.2.2.

With these conditions, we trained all PHMMs sharing the same dependency function $\boldsymbol{f}(\boldsymbol{\theta})$ as explained in Section 3.2.1, except for the one representing "go behind the box left", which was trained apart. Each PHMM was trained with $N = 8$ states with $K = 1$ mixture components each. The GMM of generalized parameter function was trained with 8 mixture components. The following results show the expected motion for a query given to the system and using the generalized parameter model learnt for all primitives. The resulting motion is then reconstructed from the invariant representation back to the task coordinates.

Experimental Results

In order to show the general parametric model's generalization capabilities, the following queries given to the system are always referring to the primitive learned apart, "go behind the box left", which was not considered in $\boldsymbol{f}(\boldsymbol{\theta})$.

The resulting motions for the queries "go behind the box left", "go behind the box left slow" and "go behind the box fast" all considering a middle box size are shown in Fig. 3.9. The motion language model selected the motion symbol representing the primitive "go behind the box left" for all three cases. As shown by the execution times, the three resulting motions, while having very similar curvature profiles differ mainly on the execution time, even when the parameterization of "fast" or "slow" for this primitive was not included in the generalized parameter model.

For the queries "go behind the box left" with box sizes 0.8, 1, 2, 3 and 4, the resulting motions are shown in Fig. 3.10. As with the previous queries, the motion language model selected always the primitive "go behind the box left". In this case, the queries consider parameter values which were even not in range of the training set and a motion that was not included in the generalized parameter model. However, the resulting motions show the intuitive response of the system which can even model the influence of the box in an abstract manner for a primitive not considered previously. While this result might be initially surprising, it is mainly due to the inclusion of the primitive "go to the left of the box" in the generalized parameter model which was trained for different box sizes. As both motion descriptions share many words ("go", "the", "box", "left"), the parameterization tends to extract most of the non-linear dependency from that model, whose motion profile is indeed similar to "go behind the box left", except for the end of the motion.

While the previous experiments show a proper response of the system to fully specified queries, the marginal influence of some parameters can also be extracted generating the expected mean posterior of their marginal distribution. In order to test the influence of the words like "fast" and "slow", we can calculate the parameterization of the model as the marginal distributions $P(\boldsymbol{f}(\boldsymbol{\theta})|\text{"fast"}=1,\text{"slow"}=0)$, $P(\boldsymbol{f}(\boldsymbol{\theta})|\text{"fast"}=0,\text{"slow"}=0)$ and $P(\boldsymbol{f}(\boldsymbol{\theta})|\text{"fast"}=0, \text{"slow"} = 1)$. The results for these 3 queries applied to the primitive "go behind the box left" are shown in Fig. 3.11. As shown in the execution times, the essence of the adverbs is present as the fast motion is the fastest and the slow motion the slowest. However, it is also remarkable that in this case the motion profiles slightly differ showing an undesired response. This effect might be produced by the small amount of primitives considered in this scenario.

In summary, the proposed approach captures the influence of external parameters on motions using PHMMs. The motion language model extracts the most likely motion symbol given a natural language query and, as shown by the motion generation results, the parameterization of primitives not considered in the training using the generalized parameter model produces motions in agreement with their real meaning. Furthermore, the influence of isolated parameters is abstracted calculating their expected marginal distribution also producing similar results.

3.3 Impedance-based GPs for Modeling Human Behavior

The previous sections explore the possibilities of time-based models in order to learn observed behavior. Although effective expressing time constraints, generalizations in time-domain are especially limited when representing tasks comprising multiple homotopy classes. In contrast, dynamical systems in the form of state-space models have complementary properties: time-specific constraints are not efficiently captured, but the generalization of task-dependent constraints is significantly more effective. In this section we explore the acquisition of state-space dynamical models by assuming an impedance model of the human arm. By means of GP priors and efficient online updates, task models conform with the arm impedance characteristics are aquired during interaction. The considered problem is formally defined in Section 3.3.1. The proposed impedance-based GP model is presented in Section 3.3.2. Due to GP's computational complexity, local and sparse approximations are needed in order to enable efficient and incremental updates during interaction as explained in Section 3.3.3. The proposed model is evaluated in Section 3.3.4, with special emphasis on settings with sparse approximations.

3.3.1 Problem Definition

The problem considered in this section consists of the definition of a human arm dynamic model for point-to-point and navigation tasks. We assume that the task dynamics of the human arm are given by

$$\boldsymbol{u}_h = \boldsymbol{f}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \xi_{\text{env}}) , \tag{3.16}$$

where $\boldsymbol{u}_h \in \mathbb{R}^6$ is the human wrench at the hand, $\boldsymbol{f}$ is a continuously differentiable function, $\boldsymbol{x} \in \mathbb{R}^6$ is the human hand configuration in task space and $\xi_{\text{env}} \in \mathbb{R}^m$ comprises task- and environment-related variables. Note that $\boldsymbol{f}$ represents both the physical arm passive properties and the task-driven human intention while performing a task.

Several regression techniques are capable of modeling $\boldsymbol{f}$ from data. Among all regression methods, GPs are especially compelling due to their performance and their ability to provide confidence bounds in unvisited state space regions. As commented in Section 2.2.2, GP priors are determined by a mean function, usually set to 0 and a covariance function specifying the correlation of input space points. Some covariance functions, such as the squared exponential or Matern kernel ensure convergence to any continuous function. Exploiting this fact a straightforward approach considers a naive 0-mean GP prior on $\boldsymbol{f}$. However, if this prior does not represent the real function well, the rate of convergence may be exponential offering no prediction guarantees in previously unobserved regions [134]. A more effective alternative incorporates available knowledge in order to give GP priors a structure in accordance with the real function to approximate. Following this idea, here we assume an impedance model based on sensorimotor principles of human arm motor control as detailed in the following Section.

3.3.2 Impedance-based GP priors

We assume the human arm follows dynamics

$$\boldsymbol{u}_h = M_h(\xi)\ddot{\boldsymbol{x}} + D_h(\xi)\dot{\boldsymbol{x}} - K_{\dot{x}}(\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_d(\overline{\xi})) - K_x(\boldsymbol{x} - \boldsymbol{x}_d(\overline{\xi})) , \tag{3.17}$$

where $M_h, D_h \in \mathbb{R}^{6\times6}$ are the human arm's inertia and damping, $K_{\boldsymbol{x}}, K_{\dot{\boldsymbol{x}}} \in \mathbb{R}^{6\times6}$ are diagonal matrices representing tracking stiffness and damping respectively and $\boldsymbol{x}_d \in \mathbb{R}^6$ is the human desired trajectory, $\xi = \left[\boldsymbol{x}^{\mathrm{T}}\dot{\boldsymbol{x}}^{\mathrm{T}}\right]^{\mathrm{T}}$ and $\overline{\xi} = \left[\boldsymbol{x}^{\mathrm{T}}\dot{\boldsymbol{x}}^{\mathrm{T}}\xi_{\text{env}}^{\mathrm{T}}\right]^{\mathrm{T}}$. The inertia and damping terms are only configuration-dependent; they render the inherent passive physical properties of the human arm. In contrast, the tracking terms term encode the human intention, i.e. the human-desired task dynamics tracking a virtual desired trajectory $\boldsymbol{x}_d(\overline{\xi})$ as a proportional-derivative controller. The desired trajectory is task-dependent and therefore comprises also environmental/task variables ξ_{env}. Note that this model is conform with many neuroscience studies which explore the human arm impedance characteristics [135]. In fact, expression (3.17) is similar to the task-space arm dynamics under the equilibrium point control (EP) hypothesys [136].

In the following we exploit the flexibility of GPs in order to efficiently approximate (3.17) from data. GPs are limited to one-dimensional outputs unless significantly higher computational cost is considered [137]. For simplicity and computational efficiency we treat each

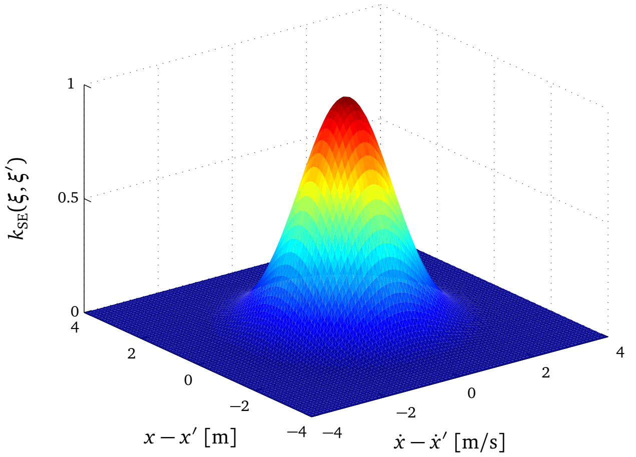


Figure 3.12: Covariances of a squared exponential kernel in a one-dimensional case for positions and velocities. Length scales and signal variance are set to $l_1 = l_2 = 1$ and $\sigma_f^2 = 1$ respectively.

dimension independently and analyze the one-dimensional case

$$u_h = m_h(\xi)\ddot{x} + d_h(\xi)\dot{x} - k_{\dot{x}}(\dot{x} - \dot{x}_d(\overline{\xi})) - k_x(\boldsymbol{x} - x_d(\overline{\xi}))\ . \tag{3.18}$$

In order to estimate unknown latent variables $m_h(\xi), d_h(\xi)$ and $x_d(\overline{\xi})$, we assume GP priors

$$\begin{aligned} m_h(\xi) &\sim \mathcal{GP}(\mu_{m_h}, k_{m_h}(\xi,\xi')) \\ d_h(\xi) &\sim \mathcal{GP}(\mu_{d_h}, k_{d_h}(\xi,\xi')) \\ e(\overline{\xi}) &\sim \mathcal{GP}(\mathbf{0}, k_e(\overline{\xi},\overline{\xi}')) \ , \end{aligned} \tag{3.19}$$

where $e(\overline{\xi}) = x - x_d(\overline{\xi})$, μ_{m_h} and μ_{d_h} are the positive expected inertia and damping. To characterize process covariances between two input points ξ and ξ' for all assumed priors (3.19), we assume Squared Exponential(SE) kernels such that

$$k_{\mathrm{SE}}(\xi,\xi') = \sigma_f^2 \exp\{-(\xi-\xi')^{\mathrm{T}}\Lambda^{-1}(\xi-\xi')\ \}\ . \tag{3.20}$$

where σ_f^2 is the signal variance and $\Lambda = \mathrm{diag}(l_1 \cdots l_6)$ comprises the length scales for each input dimension. This stationary kernel is illustrated for a one-dimensional case in Fig. 3.12. The SE kernel is the most widespread and applied kernel due to its smoothness and convergence properties. See [134] for an in-depth analysis.

From (3.19), the time derivative of the tracking error $\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_d(\overline{\xi})$ is given by

$$\dot{e}(\overline{\xi}) = \frac{\partial e(\overline{\xi})}{\partial t} = \frac{\partial e(\overline{\xi})}{\partial \overline{\xi}} \frac{\partial \overline{\xi}}{\partial t}\ .$$

As differentiation is a linear operator and considering assumptions (3.19), $\dot{e}(\overline{\xi})$ remains a GP given by

$$\dot{e}(\overline{\xi}) \sim \mathcal{GP}(\mathbf{0}, k_{\dot{e}}(\overline{\xi},\overline{\xi}'))\ ,$$

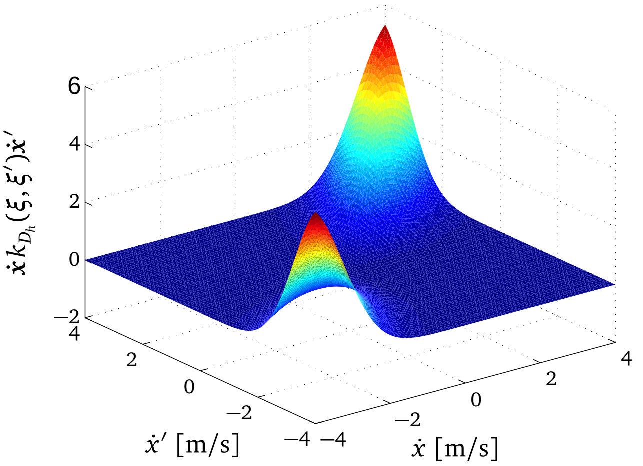


Figure 3.13: Covariances for the term $\dot{\boldsymbol{x}} k_{d_h}(\xi,\xi')\dot{\boldsymbol{x}}'$ considering only velocities for the one-dimensional case. Length scales and signal variance are set to $l_2 = 1$ and $\sigma_f^2 = 1$ respectively.

with

$$k_{\dot{e}}(\bar{\xi},\bar{\xi}') = \dot{\bar{\xi}}^{\mathrm{T}} \frac{\partial^2 k_e(\bar{\xi},\bar{\xi}')}{\partial\bar{\xi}\partial\bar{\xi}'} \dot{\bar{\xi}}' \ . \tag{3.21}$$

For a SE kernel, the gradient is

$$\frac{\partial^2 k_e(\bar{\xi},\bar{\xi}')}{\partial\bar{\xi}\partial\bar{\xi}'} = k_e(\bar{\xi},\bar{\xi}')\Lambda^{-1}(I - \Lambda^{-1}(\bar{\xi}-\bar{\xi}')(\bar{\xi}-\bar{\xi}')^{\mathrm{T}}) ,$$

Finally, as GPs are closed under addition, assumptions (3.19) imply that the human wrench is also a GP

$$\boldsymbol{u}_h \sim \mathcal{GP}(\mu_{m_h}\ddot{\boldsymbol{x}} + \mu_{d_h}\dot{\boldsymbol{x}}\ ,\ k_{u_h}(\tilde{\xi},\tilde{\xi}')) , \tag{3.22}$$

where $\tilde{\xi} = \begin{bmatrix}\bar{\xi} & \dot{\bar{\xi}}\end{bmatrix}^{\mathrm{T}}$ and

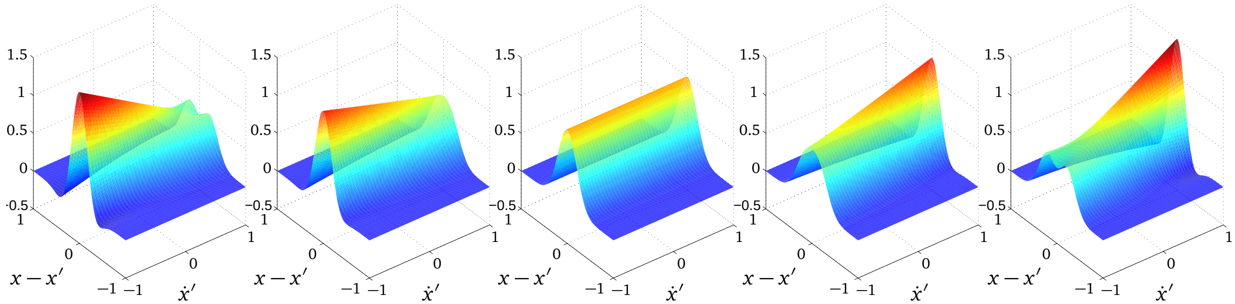
$$k_{u_h}(\tilde{\xi},\tilde{\xi}') = \ddot{x} k_{m_h}(\xi,\xi')\ddot{x}' + \dot{x} k_{d_h}(\xi,\xi')\dot{x}' + k_{\dot{x}}^2 k_{\dot{e}}(\bar{\xi},\bar{\xi}') + k_x^2 k_e(\bar{\xi},\bar{\xi}')) \ . \tag{3.23}$$

Further than the respective signal variances and length scales of each covariance function, this kernel presents the two additional parameters: the proportional k_x and derivative $k_{\dot{x}}$ gains of the tracking controller.

The resulting prior (3.22) has several interesting properties. First of all, a priori, the expected human behavior corresponds to a mass-damper system

$$\mathsf{E}[u_h] = \mu_{m_h}\ddot{x} + \mu_{d_h}\dot{x} \ ,$$

i.e., a purely passive element. This fact ensures a priori stability of the expected human dynamics. Additionally, it is remarkable that covariances terms $k_{m_h}(\xi,\xi')$ and $k_{d_h}(\xi,\xi')$



$\dot{x} = -0.1$ $\dot{x} = -0.05$ $\dot{x} = 0$ $\dot{x} = 0.05$ $\dot{x} = 0.1$

Figure 3.14: Covariances of the tracking terms $k_{\dot{e}}(\overline{\xi},\overline{\xi}') + k_e(\overline{\xi},\overline{\xi}')$ for $\overline{\xi} = x$. Length scales and signal variance are set to $l_1 = l_2 = 0.2$ and $\sigma_f^2 = 1$ respectively.

linearly depend on velocity and acceleration respectively. This fact responds to the physical constraints of (3.18). In terms of dynamics, for 0 velocity or acceleration, mass or damping uncertainty is indeed neglectable. To illustrate this, covariances of the damping term $\dot{\boldsymbol{x}} k_{D_h}(\xi,\xi')\dot{\boldsymbol{x}}'$ for different velocities are shown in Fig. 3.13. Even when velocities match, around $\dot{x} = 0$ no covariance is considered due to the $\dot{x}$ and $\dot{x}'$ factors. The same applies for the mass term w.r.t. accelerations. Note that in (3.19) we do not strictly ensure positivity of $m_h(\xi)$ and $d_h(\xi)$ in order to keep (3.22) Gaussian [1]. Instead, we assume that both mass and damping present low signal variance and remain close to their positive expected values μ_{m_h} and μ_{d_h}.

The covariance of the tracking terms $k_{\dot{e}}(\overline{\xi},\overline{\xi}') + k_e(\overline{\xi},\overline{\xi}')$ for one-dimensional simplified case with $\overline{\xi} = x$ is depicted in Fig. 3.14. From(3.21), when $\dot{x} = 0$ or $\dot{x}' = 0$, the derivative term is neglected and the covariance reduces to the SE kernel of the proportional term $k_e(\xi,\xi')$. In contrast, when both $\dot{x}$ and $\dot{x}'$ are non-zero, velocities in the direction of the gradient are proportionally more correlated due to the $k_{\dot{e}}(\xi,\xi')$ term. Velocities in the negative gradient direction yield opposite results. Proportional and derivative gains k_x and $k_{\dot{x}}$ tune the influence of both terms.

Note that the relevance of the different terms of covariance function (3.23) depend on the assumed signal variances σ_f^2 of each term. If a parameter, e.g. the mass, is expected to vary significantly and thereby has high signal variance its influence on (3.23) will be relevant. In contrast, if we assume that its mean estimate is almost deterministic having very low signal variance then its influence on (3.23) is neglectable.

3.3.3 Online Local Sparse GPs

As explained in Section 2.2.2, the computational costs of GP predictions is $\mathcal{O}(E^3)$, where E is the number of observations. This fact makes GPs prohibitive for large datasets, or, in our problem setting, for long interactions sessions. Even sparse GPs using inducing inputs [139] may not suffice when accurate and fast predictions are necessary. In this case local GPs [140], which partition the input space into local models are a useful tool. Here, we follow [141],

[1] Positivity is ensured by means of warped GPs [138]. In this case the GP priors are set as $\log(m_h(\xi)) \sim \mathcal{GP}(\log\mu_{m_h}, k_{m_h}(\xi,\xi'))$ and $\log d_h(\xi) \sim \mathcal{GP}(\log\mu_{d_h}, k_{d_h}(\xi,\xi'))$.

where predictions are computed as an average weighted sum of L local GPs similar to LWPR given by

$$y(\xi) = \frac{\sum_{l=1}^{L} k_{u_h}(\xi, \xi_l) y_l(\xi)}{\sum_{l=1}^{L} k_{u_h}(\xi, \xi_l)} ,$$

where ξ_l and y_l are the center an the prediction of the l-th local GP. In order to ensure a constant computation time, we consider that the input-space is bounded by the operational space constraints of the human arm and a partitioning is possible a priori [2]. This enables the definition of each cluster as a sparse GP, ensuring fast predictions. For each incoming input each local model is updated efficiently by means of the methods described in [142]. Assuming that each local sparse GP has B inducing inputs, the complexity of online updates and predictions is $\mathcal{O}(LB^2)$.

3.3.4 Evaluation

In order to validate the proposed model, we evaluate model performance with the dataset captured in the experimental setup from Section 3.1.

Experimental setup

Using the two-dimensional setup from appendix A.1 implementing the simple admittance control scheme from (2.16) with mass 40 [kg] and damping 90 [Ns/m], we let 6 different subjects (all male, average age 28) perform the simple task presented in Section 3.1 depicted in Fig. 3.3. The task was repeated cyclically 15 times with no pauses. The subjects had the opportunity to familiarize with the haptic device for a few minutes before the experiment and were told to follow the pre-defined trajectory for the given repetitions. The workspace of the experimental device is $(-0.15, 0.15)[m]$ for each dimension.

For simplicity we neglect terms comprising accelerations and we assume no environmental variables. We limit our evaluation model to the form

$$\boldsymbol{u}_h \sim \mathcal{GP}(\mu_{d_h} \dot{\boldsymbol{x}} \ , \ k_{u_h}(\xi, \xi')) , \tag{3.24}$$

with

$$k_{u_h}(\xi, \xi') = \dot{x} k_{d_h}(\xi, \xi') \dot{x}' + k_{\dot{x}}^2 k_{\dot{e}}(x, x') + k_x^2 k_e(x, x')) .$$

Although simplified, this model captures the influence of the assumed damping and the proportional-derivative structure of the tracking terms. Note that the desired virtual trajectory is only position-dependent,i.e. $x_d(x)$. From human arm impedance studies [135], we fix the expected human damping to $\mu_{d_h} = 2$ [Ns/m] with low signal variance $\sigma_f^2 = 0.5$. In contrast, the tracking covariance $k_e(x, x')$ is set up to $\sigma_f^2 = 4$ as it is expected to account for all human task-oriented activity. PD gains are set to $k_x = k_{\dot{x}} = 1$.

[2] If this is not the case, several methods address this issue by incrementally building data clusters, e.g. [141].

For evaluation purposes, we compare performance w.r.t. a naive GP prior model with SE kernel

$$\boldsymbol{u}_h \sim \mathcal{GP}(\boldsymbol{0}\ ,\ k_{SE}(\xi, \xi')) \tag{3.25}$$

and fixed signal variance $\sigma_f^2 = 4$. The likelihood for both model (3.24) and (3.25) is set to $\sigma_n^2 = 1$.

In our local sparse implementation, we set a total of 10 × 10 local models distributed as a grid covering the whole workspace. Each local model is composed of (4 × 4) × (4 × 4) basis vectors distributed as a grid in position and velocity space respectively. Velocities are assumed to be in the range $(-0.12, 0.12)[m]$ due to task constraints. The length-scales for both models are set to the distance between neighbour basis vectors.

Experimental Results

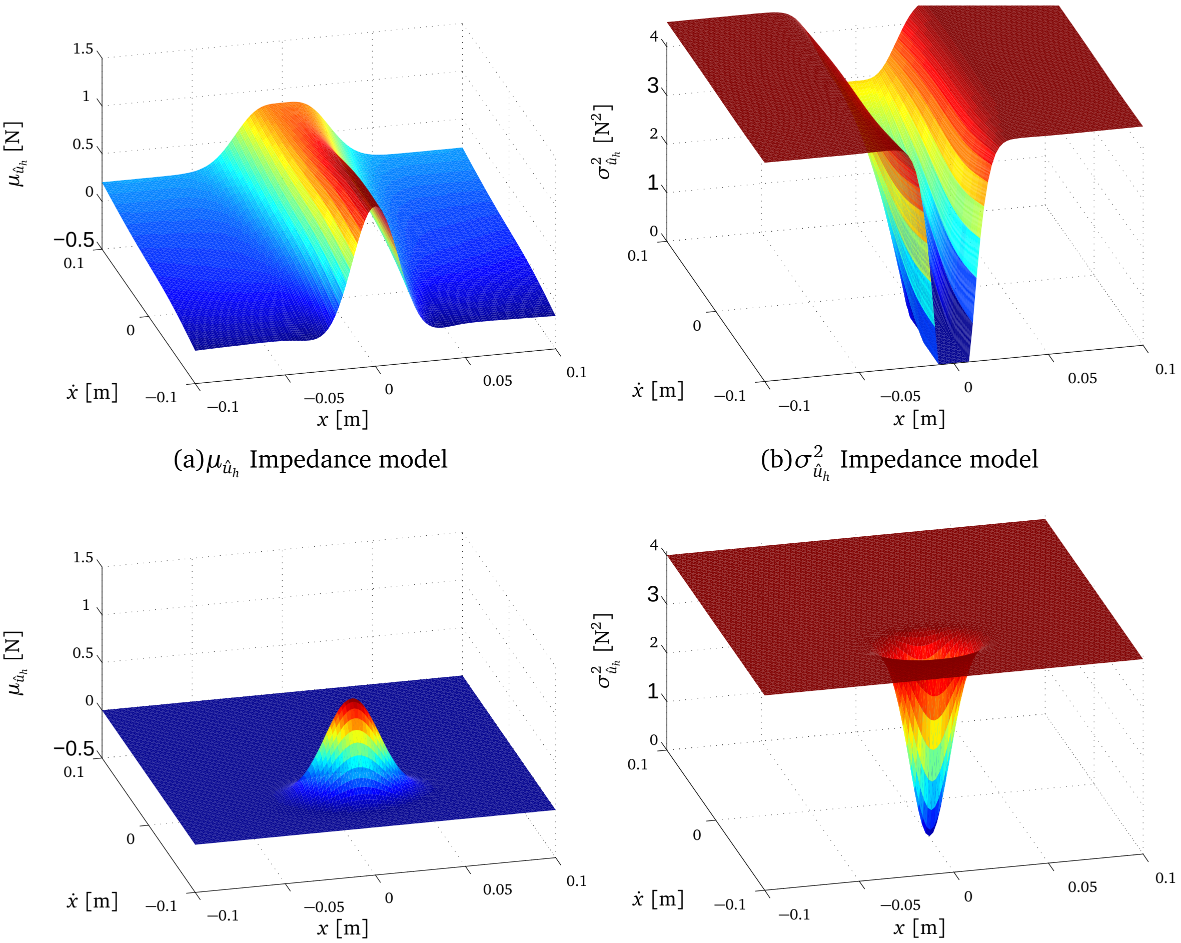
We first analyze predictions $\hat{u}_h \sim \mathcal{N}(\mu_{\hat{u}_h} | \sigma^2_{\hat{u}_h})$ for a simple one-dimensional example. Fig. 3.15 shows the predictive distribution for the proposed model (3.24) after observing a single input-output pair in comparison with model (3.25). Predictive means for model (3.24) shown in Fig. 3.15(a) present a velocity-dependent slope due to the expected damping $\overline{d}_h$, in contrast to the 0-mean of model (3.25) depicted in Fig. 3.15(c). Additionally, as model (3.24) assumes a position-dependent trajectory, expected means for all positions in the vicinity of the observed sample vary significantly in contrast to the SE model, whose expected values remain local in position-velocity space. The same applies to predictive variances depicted in Fig. 3.15(b) and (d) respectively. The SE kernel assumes only local confidence, while the proposed model provides high confidence for velocities matching the observed sample and low confidence towards the opposite direction. Note that the damping term variability hast almost no influence due to low signal variance and low velocities.

We then evaluate the model in the two-dimensional experiment from Section 3.1. The predictive variance distribution considering different input velocities is depicted for both models in Fig. 3.16 and Fig. 3.18 respectively after a single trial for one of the users. Note the difference between confidence levels. Again, the SE kernel enables only local interactions in position-velocity space while model (3.24) assumes a position-dependent tracking that enables non-local interaction in position space. The PD gains determine how strong this interaction is. This particular dependency is illustrated in Fig. 3.17 for a fixed proportional gain. As damping gains decrease the model converges to a position-only SE kernel.

	(A)	(B)
Impedance-based model (3.24)	1.87	3.11
SE kernel (3.25)	1.94	3.20

Tabular 3.1: RMSE of GP models for both experimental conditions.

To assess prediction performance, we evaluate the captured dataset for two conditions using the online local sparse approximation described in the experimental setup. For the first condition (A), we update the model online for every observation during interaction. For



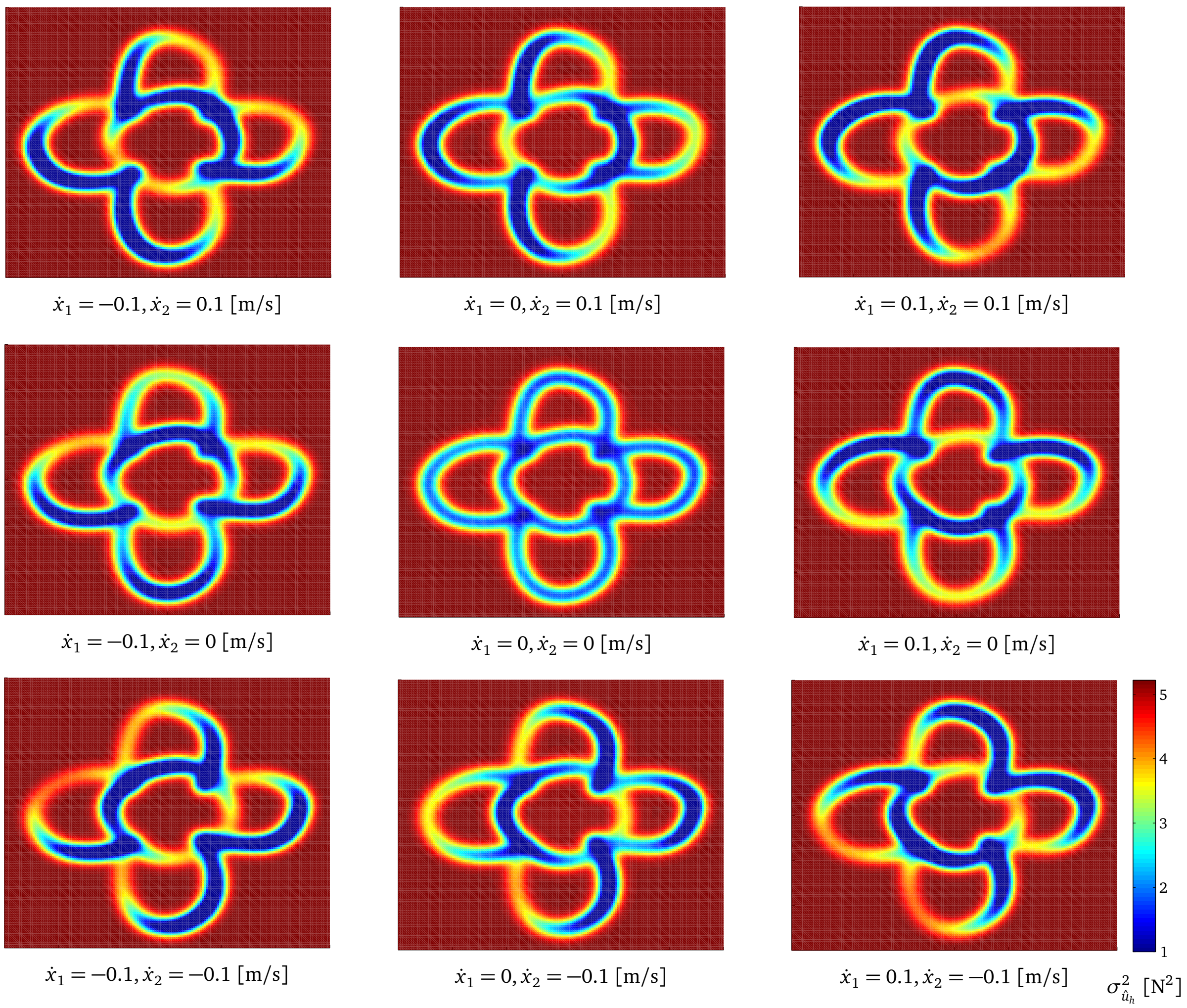
(a)$\mu_{\hat{u}_h}$ Impedance model

(b)$\sigma^2_{\hat{u}_h}$ Impedance model

(c)$\mu_{\hat{u}_h}$ SE

(d)$\sigma^2_{\hat{u}_h}$ SE

Figure 3.15: Predictive distribution after observing the input-output pair $([0, -0.05]^{\mathrm{T}}, 1)$ for the proposed impedance-based model (3.24) and a 0-mean with SE covariance GP (3.25). Length scales and signal variance are set to $l_1 = 0.01$ and $\sigma_f^2 = 4$ for $k_e(x, x')$ and $l_1 = 0.01$, $l_2 = 0.01$ and $\sigma_f^2 = 0.5$ for $k_{d_h}(\xi, \xi')$. For the SE covariance GP, length-scales and signal variance are set to $l_1 = 0.01$, $l_2 = 0.01$ and $\sigma_f^2 = 0.5$ respectively.


$\dot{x}_1 = -0.1, \dot{x}_2 = 0.1$ [m/s]

$\dot{x}_1 = 0, \dot{x}_2 = 0.1$ [m/s]

$\dot{x}_1 = 0.1, \dot{x}_2 = 0.1$ [m/s]

$\dot{x}_1 = -0.1, \dot{x}_2 = 0$ [m/s]

$\dot{x}_1 = 0, \dot{x}_2 = 0$ [m/s]

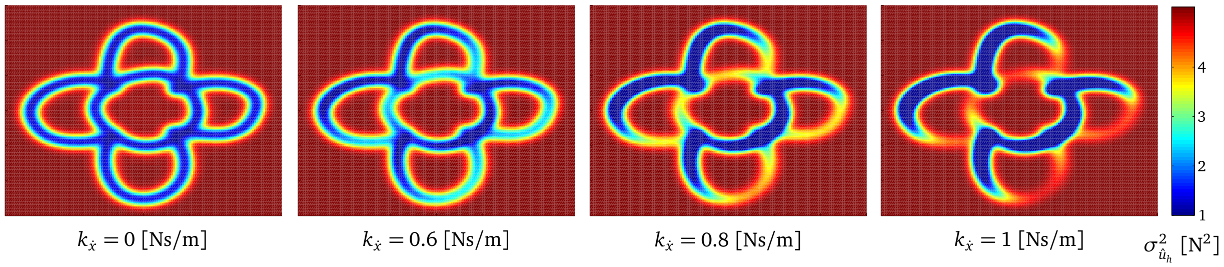
$\dot{x}_1 = 0.1, \dot{x}_2 = 0$ [m/s]

$\dot{x}_1 = -0.1, \dot{x}_2 = -0.1$ [m/s]

$\dot{x}_1 = 0, \dot{x}_2 = -0.1$ [m/s]

$\dot{x}_1 = 0.1, \dot{x}_2 = -0.1$ [m/s]

Figure 3.16: Predictive variances after one trial for model (3.24). Length scales and signal variance are set to $l_1 = 0.01$ and $\sigma_f^2 = 4$ for $k_e(x, x')$ and $l_1 = 0.01$, $l_2 = 0.1$ with damping $k_{\dot{x}} = 0.8$ [Ns/m] and $\sigma_f^2 = 0.5$ for $k_{d_h}(\xi, \xi')$.


$k_{\dot{x}} = 0$ [Ns/m]

$k_{\dot{x}} = 0.6$ [Ns/m]

$k_{\dot{x}} = 0.8$ [Ns/m]

$k_{\dot{x}} = 1$ [Ns/m]

Figure 3.17: Predictive variances after one trial for different derivative gains $k_{\dot{x}}$ for model (3.24) with $k_x = 1$ and for $\dot{x}_1 = 0, \dot{x}_2 = 0.1$ [m/s]. Length scales and signal variance are set to $l_1 = 0.01$ and $\sigma_f^2 = 4$ for $k_e(x, x')$ and $l_1 = 0.01$, $l_2 = 0.1$ and $\sigma_f^2 = 0.5$ for $k_{d_h}(\xi, \xi')$.

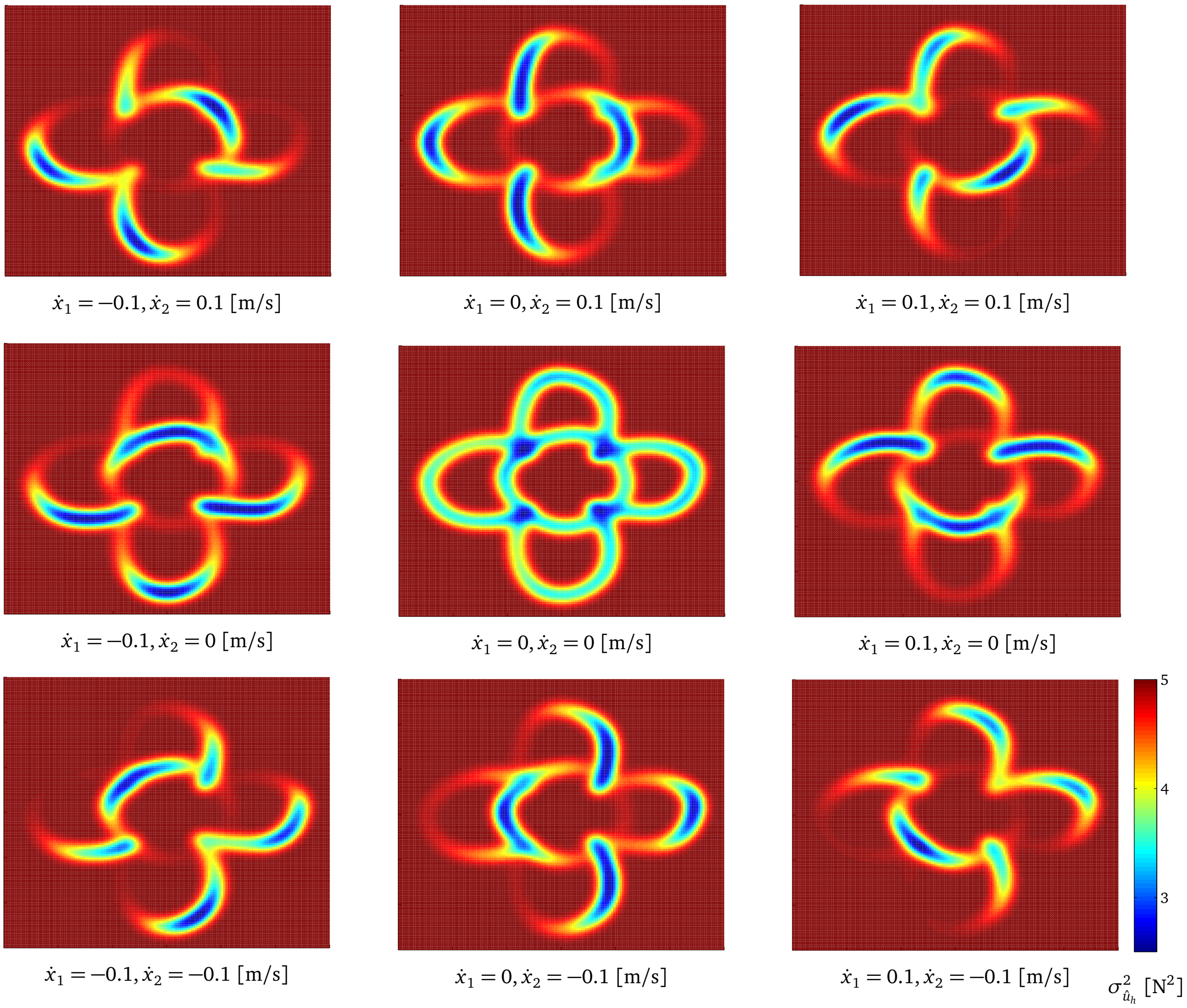


Figure 3.18: Predictive variances after one trial for model (3.25). Length-scales and signal variance are set to $l_1 = 0.01$, $l_2 = 0.1$ and $\sigma_f^2 = 0.5$ respectively.

the second condition (B), we update the model only during the first of the 15 task repetitions. Performance is evaluated in terms of the root mean squared error (RMSE). Results are shown in Table 3.1. The proposed model performs marginally better in both experimental conditions.

In summary, the proposed impedance-based model presents several desirable characteristics for modeling human task-solving behavior which also provide better performance. Of special interest is the underlying assumption of a PD tracking control scheme where the desired trajectory is modeled as a latent variable. This modeling scheme has room for further studies and extensions with potential applicability in other settings further than human behavior modeling. As shown in our user study, local and sparse GP implementations enable their applicability in online settings supporting even long-interaction sessions.

3.4 Summary

This chapter explored suitable methods for acquiring behavior models during interaction. In Section 3.1 we presented time-based HMMs where time information is learned explicitly. Efficient regression techniques together with the application of an incremental learning framework enables smoother predictions and learning during interaction, as validated in the presented user study. Section 3.2 extended this model with latent parameters, such as environmental information or natural language thereby expanding possible interaction modalities. Time-series models for learning task dynamics provide safe predictions when observations lie in the same homotopy class. However, the explicit time-dependency hinders the acquisition of motions comprising different homotopy classes and state-dependent constraints. As an alternative, in Section 3.3 we studied the applicability of dynamical systems by means of GPs in order to learn task dynamics during interaction. By incorporating an a priori impedance model, predicted human behavior has desirable compliance properties in unvisited regions of the state space. Improved overall performance was achieved assuming an underlying PD tracking structure tracking a latent virtual trajectory.

4

Uncertainty-dependent Optimal Control

One of the core capabilities of intelligent systems is the appropriate reaction to uncertainties in its environment as recognized in AI since the late 80's. While modern reasoning methods embrace stochastic uncertainties in a sophisticated way, low level robot control still largely lacks suitable approaches. State-of-the art stochastic control methods consider the expected value and ignore higher order statistics. The objective of this article is to fill the gap between probabilistic modeling approaches and robot control by providing a systematic approach to robot control in the presence of stochastic uncertainties beyond first-order statistics. As an example, consider the task of grasping an object with uncertain pose. Its expected value is only a rough estimation of its probabilistic representation. Its variance, though, may reveal more certainty in some degrees of freedom; control design targeting those more aggressively while allowing more variability in others may be key for task success. Similarly, if we consider a navigation task in a cluttered environment where obstacles have different levels of uncertainty, the possibility of collision may vary significantly depending on the obstacles' variances. These simple prototypical examples illustrate the necessity of a systematic and flexible control approach that considers uncertainty in robotic settings.

Due to the ability of humans succeeding on a wide variety of tasks, a reliable source of inspiration for robot control design is human behavior. Their capability to overcome and even benefit from the effects of noise and uncertainty bring neuroscientists to study human sensorimotor control [143]. In particular, results in this area show that human motor control follows from the minimization of a cost function with consideration of noise in the dynamics [144]. These findings motivate many stochastic optimal control approaches for robot control which minimize the *expected value* of a stochastic cost [145,146]. However, recent results accurately modulate human actions by means of *risk-sensitive* optimal control [147]. Intuitively, a risk-sensitive decision-maker deviates from the expected optimum in the face of uncertainty by considering not only the expected value of the stochastic cost but also its variance and further higher-order statistics (moments or cumulants). Interestingly, human uncertainty-dependent behavior is not limited to the effects of sensorimotor noise. Recent studies demonstrate that human actions are similarly affected by the uncertainty of task-related variables, such as a goal to reach [148] or the internal model of a cooperating partner [149]. These insights motivate us to explore risk-sensitive decision-makers for prototypical robot control problems where not only the plant but also task- and environment-dependent variables are uncertain. This problem setting naturally arises when robots are deployed in unstructured environments and must adapt to unknown surroundings, thereby estimating uncertain dynamics, goal and obstacle configurations. To the best of our knowledge, optimal control design considering high-order statistics under multiple uncertainty sources is still an open issue with potential application in many robotic scenarios.

In this chapter we study the stochastic locally optimal control problem considering high-order cost statistics for application in robotic settings. We present risk-sensitive and cost-cumulant solutions for problems with non-linear dynamics, multiple additive uncertainty sources, uncertain parameters and non-quadratic costs. Locally optimal solutions are found by iteratively performing a linear quadratic (LQ) approximation around a nominal trajectory, solving the local problem and updating the trajectory until convergence. Simulation results of a point mass robot, a two-link manipulator and a car-like robot validate the applicability of the proposed approach and illustrate its peculiarities.

The rest of this chapter is organized as follows: Section 4.1 formally defines the problem and exposes the proposed iterative locally optimal approach. Section 4.2 and Section 4.3 present the risk-sensitive and cost-cumulant solutions for the linear quadratic approximation of the problem respectively. The main iteration of the algorithm is described in Section 4.4. High-order cost statistics evaluations for exemplary static uncertain parameters are additionally presented in Section 4.5. Experimental simulations on a point mass, a two-link robotic arm and a car-like model are illustrated in Section 4.6.

Related Work

Several works explore the synthesis of uncertainty-dependent decisions in robotic settings from an optimality perspective. In order to do so, the performance measure of deterministic settings is modified in order to account for the uncertainty's variance. For instance, enhanced collision avoidance is achieved including an additive cost term in representing the probability of collision in [150]. Similarly, variance-dependent stiffness is synthesized adding prediction variance to the performance measure as in [88]. The inverse of the variance is also used as a weighting term for measuring distance to desired states in [63] or [9]. The heterogeneity of these solutions reveals the lack of a systematic approach to synthesize uncertainty-dependent actions. This work considers a linear combination of the statistics of the random cost yielding a broad family of uncertainty-dependent decision-makers.

Local linear-quadratic approximations to non-linear systems are an effective solution to robot control problems. The iterative linear quadratic Gaussian (iLQG) method presented in [151] which is a simplified version of Stochastic Differential Dynamic Programming (SDDP) [152] is a recurrent example. The resulting locally optimal feedback policies have been applied in many robot navigation problems combined with belief roadmaps [153] or adding the state variance as a part of an augmented state [145,154]. All these approaches consider the expected value of the random cost as a performance measure, neglecting cost variance. Here, we study a similar problem setting but considering higher order cumulants of the cost. We further extends the results from [117] to a more general family of linear quadratic systems that enable the application of the iterative algorithm in both risk-sensitive and cost-cumulant solutions.

The statistics of random costs have been primarily studied in the context of mathematical finance, initially exploring mean and variance solutions [113]. Further than the first two cumulants, the risk-sensitive performance criterion initially proposed in [106] and usually applied in modern portfolio theory, considers the sum of all infinite cumulants of the cost [115]. The discrete-time solution was widely studied in [155] for the linear quadratic setting and the consideration of higher order cost cumulants has recently gained interest in robotics. Recent works applied this concept to stochastic cost functions learned with Gaussian Processes [110] as well as in the context of approximate inference control [111, 112]. However, risk-sensitive control assumes a specific weighting in the infinite summation of the cumulants. When more flexibility in the weightings is needed, cost cumulant control is a suitable generalization of the risk-sensitive criterion. Main results for continuous-time linear quadratic settings are found in [116] or for the discrete-time case in [117]. In this work we extend these results to a more general family of linear quadratic systems that enable the application of the iterative algorithm.

4.1 Stochastic Optimal Control with High-Order Costs Statistics

Consider a robot with dynamics given by the completely observed controlled diffusion process

$$d\boldsymbol{x}^r(t) = \boldsymbol{f}^r(\boldsymbol{x}^r(t), \boldsymbol{u}^*(t))dt + \mathscr{G}^r dB^r(t)\,, \tag{4.1}$$

where $\boldsymbol{x}^r \in \mathbb{R}^n$, $\boldsymbol{u}^* \in \mathbb{R}^m$ represent the robot's state and control input respectively, B^r is a n-dimensional standard Brownian motion noise defined in the complete probability space $(\Omega_r, \mathscr{F}_r, \mathsf{P}_r)$, $\mathscr{G}^r \in \mathbb{R}^{n\times n}$ is its infinitesimal variance and $\boldsymbol{x}^r_0$ is the initial state. Additionally, let $\boldsymbol{x}^g \in \mathbb{R}^n$ and $\boldsymbol{x}^o \in \mathbb{R}^n$ be a desired trajectory to follow and the state of an obstacle respectively [1] with completely observed dynamics

$$\begin{aligned} d\boldsymbol{x}^g(t) &= \boldsymbol{f}^g(\boldsymbol{x}^g(t))dt + \mathscr{G}^g dB^g(t) \\ d\boldsymbol{x}^o(t) &= \boldsymbol{f}^o(\boldsymbol{x}^o(t))dt + \mathscr{G}^o dB^o(t)\,, \end{aligned} \tag{4.2}$$

where B^g and B^o are n-dimensional standard Brownian motion noises defined in complete probability spaces $(\Omega_g, \mathscr{F}_g, \mathsf{P}_g)$ and $(\Omega_o, \mathscr{F}_o, \mathsf{P}_o)$ respectively, $\mathscr{G}_g \in \mathbb{R}^{n\times n}$ and $\mathscr{G}_o \in \mathbb{R}^{n\times n}$ their infinitesimal variances and $\boldsymbol{x}^g_0$ and $\boldsymbol{x}^o_0$ their respective initial states.

The fact that all elements involved in the problem (robot, goal and obstacle) are governed by diffusion processes might appear as an extreme or rare case. However, the emergence of *probabilistic models* as an effective machine learning tool for acquiring motion models of

[1] Although for simplicity we consider here a single obstacle, a generalization to an arbitrary number is straightforward

humans or any other dynamic elements in the environment gives cause for the considered problem setting. Any decision-making process relying on such models is consequently affected by their intrinsic probabilistic nature. For instance, Bayesian regression models are usually in the form

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\epsilon}$$

where $\boldsymbol{y}$ is the output, $\boldsymbol{x}$ the input and $\boldsymbol{\epsilon}$ is normally distributed observation noise, i.e. $\mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \Sigma_\epsilon)$. Let now $\boldsymbol{x}^i$ be the state of either a desired goal, obstacle or robot. Given a set of input-output samples $(\boldsymbol{x}^i, d\boldsymbol{x}^i)$, the corresponding vector field $\boldsymbol{f}^i(\boldsymbol{x}^i) \sim \mathcal{N}(\boldsymbol{f}^i(\boldsymbol{x}^i)|\boldsymbol{f}^i_\mu(\boldsymbol{x}^i), \boldsymbol{f}^i_\Sigma(\boldsymbol{x}^i))$ is approximated in a data-driven manner. Interpreting the result as a diffusion process yields

$$d\boldsymbol{x}^i(t) = \boldsymbol{f}^i_\mu(\boldsymbol{x}^i(t))dt + (\boldsymbol{f}^i_\Sigma(\boldsymbol{x}^i(t)) + \Sigma_\epsilon)^{1/2} dB^i(t) \,.$$

Note that the Brownian motion considered here might not be present in the observed data: predicted uncertainty might be just caused by an insufficient training set, i.e. lack of knowledge. However, it reflects the potential unexpected behavior that the probabilistic model $\boldsymbol{f}^i(\boldsymbol{x}^i)$ implies [2]. Examples of this modeling approach are found in the literature for approximating plant dynamics [156], dynamic obstacle motions [157] or desired goal dynamics [73].

The measure evaluating the performance of a candidate control law $\boldsymbol{u}^*(\cdot)$ is given by cost function

$$J(\boldsymbol{u}^*(\cdot)) = h_{T_c}(\boldsymbol{x}^r(T_c), x^g(T_c), x^o(T_c)) + \int\limits_{t=0}^{T_c} h(t, \boldsymbol{x}^r(t), \boldsymbol{x}^g(t), \boldsymbol{x}^o(t), \boldsymbol{u}^*(t))dt \,, \tag{4.3}$$

where T_c is the time horizon, h is the cost rate and h_{T_c} the final cost. This performance index is usually designed penalizing both the distance to the desired goal $\boldsymbol{x}^g$ and the necessary control efforts $\boldsymbol{u}^*$ while favoring configurations distant to obstacles $\boldsymbol{x}^o$.

As (4.3) is usually defined in terms of errors between robot and obstacle and robot and goal and in order to keep a compact formulation, let ξ^* be the joint state such that

$$\xi^* = [(\boldsymbol{x}^g - \boldsymbol{x}^r) \ \ (\boldsymbol{x}^o - \boldsymbol{x}^r)]^{\mathrm{T}}. \tag{4.4}$$

The robot-obstacle-goal dynamics are then compactly formulated as

$$d\xi^*(t) = \boldsymbol{f}(\xi^*(t), \boldsymbol{u}^*(t))dt + \sum_{s=1}^{S} \mathcal{G}^s dB^s(t) \,, \tag{4.5}$$

where S is the number of independent Brownian motions (three in our particular case), B^s is the corresponding n-dimensional standard Brownian motion noise defined in the s-th probability space $(\Omega_s, \mathcal{F}_s, \mathsf{P}_s)$ for $s = 1 \cdots S$, and $\mathcal{G}^s$ and $\boldsymbol{f}$ are defined such that (4.1), (4.2) and (4.4) hold.

[2] Although for completeness we considered here state-dependent variances we limit our study to the additive case.

The optimal control solution is given by the control law $\boldsymbol{u}^*(\cdot)$ that minimizes (4.3) constrained to dynamics (4.5) and (4.4). Due to the stochastic nature of (4.5), the cost to be optimized is a random variable. Hence, prior to finding an optimal solution, an interpretation of the random cost in terms of a deterministic performance measure is necessary. A valid approach consists of evaluating a statistical measure of (4.3) usually limited to the expected value, i.e. $\mathsf{E}_{\mathsf{P}}[J]$ and where the expectation is defined in the product probability space of all uncertainty sources, i.e. the probability space $(\Omega, \mathscr{F}, \mathsf{P})$ given by

$$\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_S \quad \mathscr{F} = \mathscr{F}_1 \times \mathscr{F}_2 \times \cdots \times \mathscr{F}_S, \tag{4.6}$$

where × denotes the Cartesian product and P is the joint measure defined on the measurable space $(\Omega, \mathscr{F})$. However, this formulation has two drawbacks. First, all statistics further than the expected value are neglected even when higher-order measures enable richer uncertainty-dependent decisions. Second, the evaluation of the statistics in probability space (4.6) considers all random variables *jointly*. This limits the way cost variability influence decisions: it might be beneficial that cost fluctuations induced *marginally* by the obstacle's randomness influence the robot's decisions in a different manner as motor noise or the goal's stochasticity.

Concerning the first issue, an arbitrary linear combination of K high-order moments

$$\mathsf{E}_{\mathsf{P}}[J] + \sum_{r=2}^{K} \gamma_r \, \mathsf{E}_{\mathsf{P}}[J^r] \,, \tag{4.7}$$

is a possibility. However, optimal control design targeting cost moments yields non-linear controllers when high-order moments are considered even in linear quadratic settings [158]. As an alternative, we focus our study on cost cumulants, yielding a similar problem formulation

$$\kappa_{\mathsf{P}}^{(1)}(J) + \sum_{r=2}^{K} \gamma_r \kappa_{\mathsf{P}}^{(r)}(J) \,, \tag{4.8}$$

where $\kappa_{\mathsf{P}}^{(r)}$ denotes the r-th cumulant calculated in probability space $(\Omega, \mathscr{F}, \mathsf{P})$. [3] While the first moment and cumulant are equal and defined by the *expected cost*, in the following we will informally refer to higher order terms present in the summation of (4.8) as *cost variability* indistinctly. Note that high order statistical terms are also denoted *risk measures* in modern portfolio theory [116]. As introduced in Section 2.3, cumulants are derived by means of the cumulant generating function

$$\Psi_{\mathsf{P}}(\theta) = \log \mathsf{E}_{\mathsf{P}}[\exp\{\theta J\}] \,. \tag{4.9}$$

Its power series expansion is given by

$$\Psi_{\mathsf{P}}(\theta) = \sum_{r=1}^{\infty} \frac{\theta^r}{r!} \kappa_{\mathsf{P}}^{(r)}(J) \,, \tag{4.10}$$

[3] For convenience, in (4.7) and (4.8) we fixed the weight of the first moment and cumulant to 1. When this is not the case, this formulation is easily recovered by normalizing all weights with the desired factor.

where

$$\kappa_{\mathsf{P}}^{(r)}(J) = \left.\frac{\partial^r \Psi_{\mathsf{P}}(\theta)}{\partial \theta^r}\right|_{\theta=0}, \tag{4.11}$$

providing a compact way to calculate the desired cumulants.

Regarding the second issue, in order to evaluate the cost variability produced marginally by the s-th stochastic process, we define its corresponding *marginal dynamics* as

$$d\xi^*(t) = \boldsymbol{f}(\xi^*(t), \boldsymbol{u}^*(t))dt + \mathscr{G}^s dB^s(t)\,. \tag{4.12}$$

We then aim for the analysis of (4.8) constrained to each of the marginal dynamics. The optimization criterion considered through this work is given by

$$\min_{\boldsymbol{u}(\cdot)} \;\Xi \;=\; \min_{\boldsymbol{u}(\cdot)} \;\frac{1}{S}\Big(\sum_{s=1}^{S} \kappa_{\mathsf{P}_s}^{(1)}(J) + \sum_{r=2}^{K} \gamma_{r,s} \kappa_{\mathsf{P}_s}^{(r)}(J)\Big) \tag{4.13}$$

where the cumulants defined in the s-th probability space are constrained to the s-th marginal dynamics (4.12). Although we do not solve the original problem with dynamics (4.5) this formulation provides more flexible decision-makers, enabled only by means of a marginal treatment of uncertainty sources.

The intuition behind problem (4.13) is simple: we aim for the controls that minimize not just the expected cost but also a *weighted average* of marginal cost variabilities. In contrast to (4.8), a solution to this problem may adopt drastically different or even opposite decisions depending on which uncertainty source is responsible for cost variability. Instrumental in this problem is also the choice of the weighting factors: cost variability terms with a positive weighting factor $\gamma_{r,s}$ will increase the overall cost while negative weighting factors produce the opposite effect. When all $\gamma_{r,s} = 0$ the expected value is recovered, which neglects any risk measure. In line with portfolio theory nomenclature, we will denote these three cases where $\gamma_{r,s} > 0$, $\gamma_{r,s} = 0$ and $\gamma_{r,s} < 0$ as *risk-averse*, *risk-seeking* and *risk-neutral* respectively, as a description of the optimizer's attitude towards a certain risk measure. If $\gamma_{r,s} < 0$ the corresponding risk measure acts as a discounting quantity of the overall cost: variability is interpreted in a risk-seeking attitude as it takes part of the expected cost. In contrast, if $\gamma_{r,s} > 0$ the corresponding risk measure implies additional costs and more control efforts, i.e a risk-averse attitude.

In addition, it is remarkable that risk-sensitive control [106] is a particular case of cost cumulant minimization [109], where weighting factors are given by the McLaurin coefficients of the power series (4.10). In fact, the risk sensitive functional defined in the s-th probability space is given by $\theta_s^{-1}\Psi_{\mathsf{P}_s}(\theta_s)$ [4] and we could similarly formulate problem (4.13) when $K = \infty$

[4] This expression is also found in the literature with a negated exponent [107], i.e. $-\theta^{-1}\log \mathsf{E}_{\mathsf{P}_s}[\exp\{-\theta J\}]$.

as

$$\min_{u^*(\cdot)} \overline{\Psi} = \min_{u^*(\cdot)} \frac{1}{S} \sum_{s=1}^{S} \theta_s^{-1} \Psi_{\mathsf{P}_s}(\theta_s)$$
$$= \min_{u^*(\cdot)} \frac{1}{S} \sum_{s=1}^{S} \theta_s^{-1} \log \mathsf{E}_{\mathsf{P}_s}[\exp\{\theta_s J\}]$$
$$= \min_{u^*(\cdot)} \frac{1}{S} \Big(\sum_{s=1}^{S} \kappa_{\mathsf{P}_s}^{(1)}(J) + \sum_{r=2}^{\infty} \frac{\theta_s^{r-1}}{r!} \kappa_{\mathsf{P}_s}^{(r)}(J) \Big) \,, \tag{4.14}$$

by fixing the the high order statistics weighting factors to the above-mentioned McLaurin coefficients divided by θ_s. The cost functional to minimize reduces therefore to an average of standard risk-sensitive functionals. Due to its relevance we will explore both the risk-sensitive (4.14) and the K-cost-cumulant control (4.13) problems.

4.1.1 Approach

The solution of (4.14) or (4.13) with non-linear dynamics and arbitrary costs is in general not attainable. As an alternative, we aim for a local optimum by means of a numerical solution. By linearizing the dynamics and quadratically approximating the cost around a discretized nominal trajectory $\big(\xi^*_{0\cdots T}, \boldsymbol{u}^*_{0\cdots T-1}\big)$, a discrete-time linear quadratic (LQ) approximation of state and control *deviations*, i.e. $\xi = (\delta\xi^* - \xi^*)$ and $\boldsymbol{u} = (\delta\boldsymbol{u}^* - \boldsymbol{u}^*)$ is obtained. Its solution is a gradient towards the local optimum, found by iteratively updating the nominal trajectory and repeating the whole process until convergence.

The local deviations problem is defined as follows. Time is discretized in T steps with sample time $\Delta = T_c/(T)$. The linearized marginal dynamics at time step k are given by

$$\xi_{k+1} = A_k \xi_k + B_k \boldsymbol{u}_k + \boldsymbol{\epsilon}_k^s \quad , \tag{4.15}$$

where $A_k \in \mathbb{R}^{n\times n}$, $B_k \in \mathbb{R}^{n\times m}$, $\boldsymbol{\epsilon}_k^s \in \mathbb{R}^{q_s}$ is an independent identically distributed Gaussian random variable such that $\mathcal{N}(\boldsymbol{\epsilon}_k^s|\boldsymbol{0}, \Sigma_k^s)$ and $\boldsymbol{x}_0 = \boldsymbol{0}$. The involved quantities are calculated as

$$A_k = I_n + \Delta \frac{\partial f_k}{\partial \xi_k^*} \quad B_k = \Delta \frac{\partial f_k}{\partial \boldsymbol{u}_k^*} \quad \Sigma_k^s = \Delta \mathscr{G}_k^{s2} \,.$$

The control input is constrained to the form

$$\boldsymbol{u}_k = \boldsymbol{l}_k + L_k \boldsymbol{x}_k \,, \tag{4.16}$$

where $\boldsymbol{l}_k$ is an affine input and $L_k \in \mathbb{R}^{m\times n}$ is the feedback matrix. The cost functional (4.3) results in the quadratic approximation

$$J(\xi_0, \boldsymbol{u}_{0\cdots T-1}) = \frac{1}{2}\xi_T^{\mathrm{T}} Q_T \xi_T + \xi_T^{\mathrm{T}} \boldsymbol{q}_T +$$
$$\sum_{k=0}^{T-1} \frac{1}{2}\xi_k^{\mathrm{T}} Q_k \xi_k + \xi_k^{\mathrm{T}} \boldsymbol{q}_k + q_k + \frac{1}{2}\boldsymbol{u}_k^{\mathrm{T}} R_k \boldsymbol{u}_k + \boldsymbol{u}_k^{\mathrm{T}} \boldsymbol{r}_k, \tag{4.17}$$

where $Q_k \in \mathbb{R}^{n\times n}$, $R_k \in \mathbb{R}^{m\times m}$, $\boldsymbol{q}_k \in \mathbb{R}^n$ and $\boldsymbol{r}_k \in \mathbb{R}^m$ and $q_k \in \mathbb{R}$ are computed as

$$
\begin{aligned}
q_k &= \Delta h_k \qquad \boldsymbol{q}_k = \Delta \frac{\partial h_k}{\partial \boldsymbol{\xi}_k^*} \qquad Q_k = \Delta \frac{\partial^2 h_k}{\partial (\boldsymbol{\xi}_k^*)^2} \\
q_T &= h_{T_c} \qquad \boldsymbol{q}_T = \frac{\partial h_{T_c}}{\partial \boldsymbol{\xi}_k^*} \qquad Q_T = \frac{\partial^2 h_{T_c}}{\partial (\boldsymbol{\xi}_k^*)^2} \\
\boldsymbol{r}_k &= \Delta \frac{\partial h_k}{\partial \boldsymbol{u}_k^*} \quad R_k = \Delta \frac{\partial^2 h_k}{\partial (\boldsymbol{u}_k^*)^2} \ ,
\end{aligned}
$$

where $Q_k \geq 0$ and $R_k > 0$.

Note that, from (4.15), $\boldsymbol{\xi}_k$ is normally distributed and hence (4.17) is a generalized non-central chi-squared distribution due to terms $\frac{1}{2}\boldsymbol{\xi}_k^{\mathrm{T}} Q_k \boldsymbol{\xi}_k$ and $\boldsymbol{\xi}_k^{\mathrm{T}} \boldsymbol{q}_k$. It remains therefore an expressive approximation to the original performance measure where cumulants higher than the expected cost are relevant.

The local deviations problem consists of minimizing (4.13) or (4.14) with J as in (4.17) and constrained to (4.15) and (4.16). The following sections study this LQ problem for two different risk-aware solutions in the complete observation case. In Section 4.2, we first explore the risk-sensitive solution by solving problem (4.14). Results on cost cumulant control will follow in Section 4.3 by solving the more general problem (4.13). The main iteration which drives the nominal trajectory towards the local optimum is explained in Section 4.4.

4.2 Risk-sensitive Solution

In this section we study the solution to the average of marginal risk-sensitive LQ problems $\overline{\Psi}$. A discrete-time LQ optimal control problem such as the one from Section 4.1.1 is solved by means of dynamic programming applying Bellman's optimality principle. From a reinforcement learning perspective, this procedure is also an offline instance of a value iteration problem with closed form solution [159].

For simplicity, consider first the risk-sensitive problem in a single probability space. The cost function to minimize is then $\theta^{-1}\Psi_{\mathsf{P}}(\theta)$ constrained to dynamics

$$
\boldsymbol{\xi}_{k+1} = A_k \boldsymbol{\xi}_k + B_k \boldsymbol{u}_k + \boldsymbol{\epsilon}_k \quad , \tag{4.18}
$$

where $\boldsymbol{\epsilon}_k \in \mathbb{R}^q$ is an independent Gaussian random variable such that $\mathcal{N}(\boldsymbol{\epsilon}_k | \mathbf{0}, \Sigma_k)$. Assume now the system is at sample time k. Given a control law $\boldsymbol{u}_{k\cdots T-1}$ in the form (4.16) and having observed $\boldsymbol{\xi}_k$, the remaining cost is calculated by means of the backwards recursion that the Bellman equation provides

$$
\begin{aligned}
\theta^{-1}\Psi_{\mathsf{P}}(\theta, \boldsymbol{\xi}_k, \boldsymbol{u}_{k\cdots T-1}) = \theta^{-1} \log \mathsf{E}_{\mathsf{P}}[\exp\{\theta J(\boldsymbol{\xi}_k, \boldsymbol{u}_k)\}] \\
+ \theta^{-1}\Psi_{\mathsf{P}}(\theta, \boldsymbol{\xi}_{k+1}, \boldsymbol{u}_{k+1\cdots T-1}) \ .
\end{aligned}
$$

The quantity $\theta^{-1}\Psi_{\mathsf{P}}(\theta, \boldsymbol{\xi}_k, \boldsymbol{u}_{k\cdots T-1})$ is the *cost-to-go*. At time-step k, as $\boldsymbol{\xi}_k$ is completely observed and therefore deterministic, it follows that

$$
\begin{aligned}
\theta^{-1}\Psi_{\mathsf{P}}(\theta, \boldsymbol{\xi}_k, \boldsymbol{u}_{k\cdots T-1}) = J(\boldsymbol{\xi}_k, \boldsymbol{u}_k) \\
+ \theta^{-1}\Psi_{\mathsf{P}}(\theta, \boldsymbol{\xi}_{k+1}, \boldsymbol{u}_{k+1\cdots T-1}) \ .
\end{aligned} \tag{4.19}
$$

Similarly, the optimal policy at time step k is computed applying the Bellman optimal equation

$$V(\xi_k) = \min_{\boldsymbol{u}_k} \left\{ J(\xi_k, \boldsymbol{u}_k) + \theta^{-1} \log \mathsf{E}_{\mathsf{P}}[V(\xi_{k+1})] \right\} , \tag{4.20}$$

where $V(\xi_k)$ is the value function at time step k.

The following lemma applies (4.19) in order to compute the risk-sensitive cost-to-go in closed form yielding a quadratic form on ξ_k. For the sake of clarity we omit dependencies on future values of $\boldsymbol{u}$.

Lemma 4.1. If $({\Sigma_k}^{-1} - \theta W_{k+1}) > 0$ for $k = 0 \cdots T-1$, the analytic solution to (4.19) is given recursively by the quadratic form

$$\theta^{-1}\Psi_{\mathsf{P}}(\theta, \xi_k, \boldsymbol{u}_{k\cdots T-1}) = \frac{1}{2}\xi_k^{\mathrm{T}} W_k \xi_k + \xi_k^{\mathrm{T}} \boldsymbol{w}_k + w_k , \tag{4.21}$$

where

$$W_k = Q_k + A_k^{\mathrm{T}} \widetilde{W}_{k+1} A_k + L_k^{\mathrm{T}} H_k L_k + G_k^{\mathrm{T}} L_k + L_k^{\mathrm{T}} G_k \tag{4.22}$$

$$\boldsymbol{w}_k = \boldsymbol{q}_k + A_k^{\mathrm{T}} \tilde{\boldsymbol{w}}_{k+1} + L_k^{\mathrm{T}} H_k \boldsymbol{l}_k + L_k^{\mathrm{T}} \boldsymbol{g}_k + G_k^{\mathrm{T}} \boldsymbol{l}_k \tag{4.23}$$

$$w_k = q_k + \tilde{w}_{k+1} + \frac{1}{2}\boldsymbol{l}_k^{\mathrm{T}} H \boldsymbol{l}_k + \boldsymbol{l}_k^{\mathrm{T}} \boldsymbol{g}_k \tag{4.24}$$

$$H_k = R_k + B_k^{\mathrm{T}} \widetilde{W}_{k+1} B_k$$

$$G_k = B_k \widetilde{W}_{k+1} A_k$$

$$\boldsymbol{g}_k = \boldsymbol{r}_k + B_k^{\mathrm{T}} \tilde{\boldsymbol{w}}_{k+1}$$

with final conditions

$$W_N = Q_N \qquad \boldsymbol{w}_N = \boldsymbol{q}_N \qquad w_N = q_N$$

and

$$\widetilde{W}_{k+1} = W_{k+1} + \theta W_{k+1} ({\Sigma_k}^{-1} - \theta W_{k+1})^{-1} W_{k+1} \tag{4.25}$$

$$\tilde{\boldsymbol{w}}_{k+1} = \boldsymbol{w}_{k+1} + \theta W_{k+1} ({\Sigma_k}^{-1} - \theta W_{k+1})^{-1} \boldsymbol{w}_{k+1} \tag{4.26}$$

$$\tilde{w}_{k+1} = w_{k+1} + \theta {\boldsymbol{w}_{k+1}}^{\mathrm{T}} ({\Sigma_k}^{-1} - \theta W_{k+1})^{-1} \boldsymbol{w}_{k+1} - \frac{1}{2}\theta^{-1} \log F_k \tag{4.27}$$

$$F_k = \left| I_q - \theta W_{k+1} \Sigma_k \right|$$

with $F_T = 1$.

Proof. The solution to the Bellman equation (4.19) entails only some complication in the term

$$\theta^{-1}\Psi_{\mathsf{P}}(\theta, \xi_{k+1}, \boldsymbol{u}_{k+1\cdots T-1})$$

Assuming quadratic form (4.21) holds yields

$$\theta^{-1}\Psi_{\mathsf{P}}(\theta, \xi_{k+1}, \boldsymbol{u}_{k+1\cdots T-1}) = \theta^{-1} \log \mathsf{E}_{\mathsf{P}}[\exp\{\theta(\xi_{k+1}^{\mathrm{T}} W_{k+1} \xi_{k+1} + \xi^{\mathrm{T}} \boldsymbol{w}_{k+1} + w_{k+1})\}] .$$

Let now $M_k = A_k \boldsymbol{\xi}_k + B_k \boldsymbol{u}_k$. Considering the system dynamics (4.18), the expression for the expectation is given by the Gaussian integral

$$\begin{aligned}
&\int \frac{1}{\sqrt{(2\pi)^{q_s}|\Sigma_k|}} \exp\left\{-\frac{1}{2}\boldsymbol{\epsilon}_k^{\mathrm{T}}\Sigma_k^{-1}\boldsymbol{\epsilon}_k\right\} \\
&\qquad \cdot \exp\left\{\theta\Big(\frac{1}{2}(M_k+\epsilon_k)^{\mathrm{T}}W_{k+1}(M_k+\epsilon_k) + (M_k + \ \epsilon_k)^{\mathrm{T}}\boldsymbol{w}_{k+1} + w_{k+1}\Big)\right\} d\boldsymbol{\epsilon}_k \\
&= \frac{1}{\sqrt{(2\pi)^{q_s}|\Sigma_k|}} \exp\left\{\theta\Big(\frac{1}{2}M_k^{\mathrm{T}}W_{k+1}M_k + M_k^{\mathrm{T}}\boldsymbol{w}_{k+1} + w_{k+1}\Big)\right\} \\
&\qquad \cdot \int \exp\left\{-\frac{1}{2}\epsilon_k^{\mathrm{T}}(\Sigma_k^{-1} - \theta W_{k+1})\epsilon_k + \epsilon_k^{\mathrm{T}}\Big(2\theta W_{k+1}M_k + \theta\boldsymbol{w}_{k+1}\Big)\right\} d\boldsymbol{\epsilon}_k \,. \qquad (4.28)
\end{aligned}$$

Recall that if $N > 0$ then

$$\int \exp\left\{-\frac{1}{2}\boldsymbol{\epsilon}^{\mathrm{T}}N\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{n}\right\} d\boldsymbol{\epsilon} = \sqrt{(2\pi)^q|N|^{-1}} \exp\left\{\boldsymbol{n}^{\mathrm{T}}N^{-1}\boldsymbol{n}\right\} \,.$$

Therefore, if $(\Sigma_k^{-1} - \theta W_{k+1}) > 0$, (4.28) has analytical solution, yielding

$$\sqrt{\frac{\left|\left(\Sigma_k^{-1} - \theta_s W_{k+1}\right)^{-1}\right|}{|\Sigma_k|}} \cdot \exp\left\{\theta_s \frac{1}{2}M_k^{\mathrm{T}}\widetilde{W}_{k+1}M_k + \theta_s M_k^{\mathrm{T}}\tilde{\boldsymbol{w}}_{k+1} + \theta_s \tilde{w}_{k+1}\right\}$$

where $\widetilde{W}_{k+1}$, $\tilde{\boldsymbol{w}}_{k+1}$, and $\tilde{w}_{k+1}$ are defined as in (4.25), (4.26) and (4.27). Note that

$$|(\Sigma_k^{-1} - \theta W_{k+1})^{-1}(I - \theta W_{k+1}\Sigma_k)| = |\Sigma_k| \,,$$

which explains the expression for F_k. Going back to the Bellman equation (4.19) and applying these results yields

$$\begin{aligned}
\theta^{-1}\Psi_{\mathrm{P}}(\theta, \boldsymbol{\xi}_k, \boldsymbol{u}_{k+1\cdots T-1}) &= \boldsymbol{\xi}_k^{\mathrm{T}}W_k\boldsymbol{\xi}_k + \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{w}_k + w_k \\
&= \frac{1}{2}\boldsymbol{\xi}_k^{\mathrm{T}}Q_k\boldsymbol{\xi}_k + \boldsymbol{\xi}_k^{\mathrm{T}}\boldsymbol{q}_k + \frac{1}{2}\boldsymbol{u}_k^{\mathrm{T}}R_k\boldsymbol{u}_k \ + \boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{r}_k \\
&\qquad + \frac{1}{2}M_k^{\mathrm{T}}\widetilde{W}_{k+1}M_k + M_k^{\mathrm{T}}\tilde{\boldsymbol{w}}_{k+1} + \tilde{w}_{k+1} \,.
\end{aligned}$$

Constraining $\boldsymbol{u}_k$ to (4.16) yields expressions (4.22), (4.23) and (4.24). □

Note that condition $(\Sigma_k{}^{-1} - \theta W_{k+1}) > 0$ is satisfied if $\theta < (\lambda_{max}(\Sigma_k)\lambda_{max}(W_{k+1}))^{-1}$, where λ_{max} denotes the largest eigenvalue. This upper bounds the degree of risk-aversion, $\theta > 0$, accepted by a risk-sensitive cost evaluation. It is also interesting to observe the alternative formulations of (4.25) and (4.26) derived in the proof of lemma 4.2, yielding

$$\begin{aligned}
\widetilde{W}_{k+1} &= (I - \theta W_{k+1}\Sigma_k)^{-1}W_{k+1} \\
\tilde{\boldsymbol{w}}_{k+1} &= (I - \theta W_{k+1}\Sigma_k)^{-1}\boldsymbol{w}_{k+1} \,.
\end{aligned}$$

The influence of θ on the cost becomes now apparent. A positive θ implies higher quadratic cost coefficients as it diminishes the eigenvalues of the inverted term. In contrast, a negative θ increases the inverted term and therefore decreases the overall cost. In fact, extreme risk-seekingness, i.e. $\theta << 0$, yields $\widetilde{W}_{k+1} \to 0$ and $\tilde{\boldsymbol{w}}_{k+1} \to \mathbf{0}$. This degenerate case is described as *euphoria* in [155], as cost variability completely nullifies the overall cost.

The cost-to-go for problem (4.14), i.e. the remaining average cost of all marginal problems at time step k, follows now straightforwardly as

$$\overline{\Psi}(\xi_k, \boldsymbol{u}_{k\cdots T-1}) = \frac{1}{2}\xi_k^{\mathrm{T}}\overline{W}_k\xi_k + \xi_k^{\mathrm{T}}\overline{\boldsymbol{w}}_k + \overline{w}_k \ , \tag{4.29}$$

where

$$\overline{W}_k = \frac{1}{S}\sum_{s=1}^{S} W_k^s \quad \overline{\boldsymbol{w}}_k = \frac{1}{S}\sum_{s=1}^{S} = \boldsymbol{w}_k^s \quad \overline{w}_k = \frac{1}{S}\sum_{s=1}^{S} w_k^s \ ,$$

and where W_k^s, $\boldsymbol{w}_k^s$ and w_k^s are the quadratic coefficients resulting from applying lemma 4.1 in the s-th probability space constrained to the s-th marginal dynamics, i.e. with $\Sigma_k = \Sigma_k^s$ and $\theta = \theta_s$. Note that the overall cost given control policy $\boldsymbol{u}_{0\cdots T-1}$ corresponds to the cost-to-go at $k = 0$, i.e. $\overline{\Psi}(\xi_0, \boldsymbol{u}_{0\cdots T-1})$.

The risk-sensitive solution is computed applying Bellman's optimal equation (4.20). This is equivalent to minimizing (4.29) w.r.t L_k and $\boldsymbol{l}_k$ at each step of the backwards recursion yielding

$$\begin{aligned} L_k &= -(R_k + B_k^{\mathrm{T}}\overline{\widetilde{W}}_{k+1}B_k)^{-1}(B_k^{\mathrm{T}}\overline{\widetilde{W}}_{k+1}A_k) \\ \boldsymbol{l}_k &= -(R_k + B_k^{\mathrm{T}}\overline{\widetilde{W}}_{k+1}B_k)^{-1}(B_k^{\mathrm{T}}\overline{\widetilde{\boldsymbol{w}}}_{k+1} + \boldsymbol{r}_k) \ , \end{aligned} \tag{4.30}$$

where

$$\overline{\widetilde{W}}_k = \frac{1}{S}\sum_{s=1}^{S} \widetilde{W}_k^s \quad \overline{\widetilde{\boldsymbol{w}}}_k = \frac{1}{S}\sum_{s=1}^{S} = \widetilde{\boldsymbol{w}}_k^s \quad \overline{\widetilde{w}}_k = \frac{1}{S}\sum_{s=1}^{S} \widetilde{w}_k^s \ ,$$

and where $\widetilde{W}_k^s$, $\widetilde{\boldsymbol{w}}_k^s$ and $\widetilde{w}_k^s$ follow from the computation of (4.29).

4.3 Cost-cumulant Solution

In this section we approach the cost-cumulant control problem (4.13) studied in [116] for the continuous-time case and in [117] for the discrete-time case. We extend it here to our problem setting with a more general family of linear quadratic systems and explore the average of marginal cost-cumulant problems. Although the risk-sensitive solution from the previous section is an approximation to an infinite summation of cumulants around $\theta = 0$, their respective weighting factors are fixed to the McLaurin coefficients of its power series. The cost-cumulant control problem allows more flexible decision-makers as the number of cumulants and their respective weighting factors are design parameters.

Following a similar treatment to the previous Section, we first consider a single probability space and dynamics (4.18). From lemma 4.1, the cumulant generating function of the cost-to-go at time step k is given by

$$\Psi_{\mathrm{P}}(\theta, \xi_k, \boldsymbol{u}_{k\cdots T-1}) = \theta\Big(\frac{1}{2}\xi_k^{\mathrm{T}}W_k\xi_k + \xi_k^{\mathrm{T}}\boldsymbol{w}_k + w_k\Big) \ . \tag{4.31}$$

In addition, the r-th cumulant is calculated by means of expression (4.11). In the following, in line with previous work, the notation $f(\theta)^{[r]}$ denotes the r-th derivative of $f(\theta)$ w.r.t θ at point $\theta = 0$, i.e. $f(\theta)^{[r]} = \frac{\partial^r f(\theta)}{\partial \theta^r}\Big|_{\theta=0}$. An $r = 0$ index corresponds to the function itself at $\theta = 0$. Additionally, $\mathscr{C}_r^j$ denotes the binomial coefficient $\binom{j}{r}$.

As an illustrative example we derive the first two cumulants. The first cumulant is given by

$$\begin{aligned}\frac{\partial \Psi_{\mathsf{P}}(\theta, \xi_k, \boldsymbol{u}_{k\cdots T-1})}{\partial \theta}\Big|_{\theta=0} &= \frac{1}{2}\xi_k^{\mathrm{T}} W_k^{[0]} \xi_k + \xi_k^{\mathrm{T}} \boldsymbol{w}_k^{[0]} + w_k^{[0]} \\ &\quad + \theta\Big(\frac{1}{2}\xi_k^{\mathrm{T}} W_k^{[1]} \xi_k + \xi_k^{\mathrm{T}} \boldsymbol{w}_k^{[1]} + w_k^{[1]}\Big)\Big|_{\theta=0},\end{aligned}$$

that, after evaluating $\theta = 0$ yields $\mathsf{E}_{\mathsf{P}_s}[J]$, the cost-to-go of a standard LQG problem. The definitions of $W_k^{[0]}$, $\boldsymbol{w}_k^{[0]}$ and $w_k^{[0]}$ are given by lemma 4.1, but with $\widetilde{W}_{k+1} = W_{k+1}^{[0]}$, $\widetilde{\boldsymbol{w}}_{k+1} = \boldsymbol{w}_{k+1}^{[0]}$ and $\widetilde{w}_{k+1} = w_{k+1}^{[0]}$.

Similarly, the second cumulant (the cost variance) is given by

$$\begin{aligned}\frac{\partial^2 \Psi_{\mathsf{P}}(\theta, \xi_k, \boldsymbol{u}_{k\cdots T-1})}{\partial \theta^2}\Big|_{\theta=0} &= 2\Big(\frac{1}{2}\xi_k^{\mathrm{T}} W_k^{[1]} \xi_k + \xi_k^{\mathrm{T}} \boldsymbol{w}_k^{[1]} + w_k^{[1]}\Big) \\ &\quad + \theta\Big(\frac{1}{2}\xi_k^{\mathrm{T}} W_k^{[2]} \xi_k + \xi_k^{\mathrm{T}} \boldsymbol{w}_k^{[2]} + w_k^{[2]}\Big)\Big|_{\theta=0}.\end{aligned}$$

which after evaluating $\theta = 0$ yields also a quadratic form. The expressions for $W_k^{[1]}$, $\boldsymbol{w}_k^{[1]}$ and $w_k^{[1]}$ as well as the ones corresponding to the d-th cumulant are calculated recursively in a similar manner to [117], as a function of cumulants of lower order than d, yielding the following lemma.

Lemma 4.2. The d-th cost cumulant of the random cost (4.17) at sample time k is given by

$$\kappa_{\mathsf{P}}^{(d)}(J(\xi_k, \boldsymbol{u}_{k\cdots T-1})) = d\Big(\frac{1}{2}\xi_k^{\mathrm{T}} W_k^{[d-1]} \xi_k + \xi_k^{\mathrm{T}} \boldsymbol{w}_k^{[d-1]} + w_k^{[d-1]}\Big)$$

where for $d = 1$

$$\begin{aligned}W_k^{[0]} &= Q_k + A_k^{\mathrm{T}} \widetilde{W}_{k+1}^{[0]} A_k + L_k^{\mathrm{T}} H_k^{[0]} L_k + G_k^{[0]\mathrm{T}} L_k + L_k^{\mathrm{T}} G^{[0]}{}_k \\ \boldsymbol{w}_k^{[0]} &= \boldsymbol{q}_k + A_k^{\mathrm{T}} \widetilde{\boldsymbol{w}}_{k+1}^{[0]} + L_k^{\mathrm{T}} H_k^{[0]} \boldsymbol{l}_k + L_k^{\mathrm{T}} \boldsymbol{g}_k^{[0]} + G_k^{[0]\mathrm{T}} \boldsymbol{l}_k \\ w_k^{[0]} &= q_k + \widetilde{w}_{k+1}^{[0]} + \frac{1}{2}\boldsymbol{l}_k^{\mathrm{T}} H_k^{[0]} \boldsymbol{l}_k + \boldsymbol{l}_k^{\mathrm{T}} \boldsymbol{g}_k^{[0]} \\ H_k^{[0]} &= R_k + B_k^{\mathrm{T}} \widetilde{W}_{k+1}^{[0]} B_k \\ G_k^{[0]} &= B_k^{\mathrm{T}} \widetilde{W}_{k+1}^{[0]} A_k \\ \boldsymbol{g}_k^{[0]} &= \boldsymbol{r}_k + B_k^{\mathrm{T}} \widetilde{\boldsymbol{w}}_{k+1}^{[0]}\end{aligned}$$

with final conditions

$$W_N^{[0]} = Q_T \qquad \boldsymbol{w}_N^{[0]} = \boldsymbol{q}_T \qquad w_N^{[0]} = q_T$$

and for $d > 1$

$$\begin{aligned}
W_k^{[r]} =& A_k^{\mathrm{T}} \widetilde{W}_{k+1}^{[r]} A_k + L_k^{\mathrm{T}} H_k^{[r]} L_k + G_k^{[r]\mathrm{T}} L_k + L_k^{\mathrm{T}} G_k^{[r]} \\
\boldsymbol{w}_k^{[r]} =& A_k^{\mathrm{T}} \widetilde{\boldsymbol{w}}_{k+1}^{[r]} + L_k^{\mathrm{T}} H_k^{[r]} \boldsymbol{l}_k + L_k^{\mathrm{T}} \boldsymbol{g}_k^{[r]} + G_k^{[r]\mathrm{T}} \boldsymbol{l}_k \\
w_k^{[r]} =& \widetilde{w}_{k+1}^{[r]} + \frac{1}{2} \boldsymbol{l}_k^{\mathrm{T}} H_k^{[r]} \boldsymbol{l}_k + \boldsymbol{l}_k^{\mathrm{T}} \boldsymbol{g}_k^{[r]} \\
H_k^{[r]} =& B_k^{\mathrm{T}} \widetilde{W}_{k+1}^{[r]} B_k \\
G_k^{[r]} =& B_k^{\mathrm{T}} \widetilde{W}_{k+1}^{[r]} A_k \\
\boldsymbol{g}_k^{[r]} =& B_k^{\mathrm{T}} \widetilde{\boldsymbol{w}}_{k+1}^{[r]}
\end{aligned}$$

with final conditions

$$W_N^{[r]} = 0^{n \times n} \qquad \boldsymbol{w}_N^{[r]} = \boldsymbol{0} \qquad w_N^{[r]} = 0$$

and

$$\widetilde{W}_{k+1}^{[r]} = W_{k+1}^{[r]} + r \sum_{j=0}^{r-1} \mathscr{C}_{r-1}^{j} W_{k+1}^{[j]} \Sigma_k \widetilde{W}_{k+1}^{[r-1-j]} \tag{4.32}$$

$$\widetilde{\boldsymbol{w}}_{k+1}^{[r]} = \boldsymbol{w}_{k+1}^{[r]} + r \sum_{j=0}^{r-1} \mathscr{C}_{r-1}^{j} W_{k+1}^{[j]} \Sigma_k \widetilde{\boldsymbol{w}}_{k+1}^{[r-1-j]} \tag{4.33}$$

$$\widetilde{w}_{k+1}^{[r]} = w_{k+1}^{[r]} + r \sum_{j=0}^{r-1} \mathscr{C}_{r-1}^{j} \boldsymbol{w}_{k+1}^{[j]} \Sigma_k \widetilde{\boldsymbol{w}}_{k+1}^{[r-1-j]} + \sum_{t=k}^{T-1} \operatorname{tr}(H_t^{[r-1]}) \tag{4.34}$$

with

$$H_t^{[r]} = (r+1) W_{t+1}^{[r]} \Sigma_t + r \sum_{j=1}^{r-1} \mathscr{C}_{r-1}^{j} W_{t+1}^{[j]} \Sigma_t H_t^{[r-1-j]} \,.$$

Proof. The results for $W_k^{[r]}$, $\boldsymbol{w}_k^{[r]}$ and $w_k^{[r]}$ are straightforward from their definitions in (4.22), (4.23) and (4.24). For the proof of expression $\widetilde{W}_{k+1}^{[r]}$ see [117]. The solution to $\widetilde{\boldsymbol{w}}_{k+1}^{[r]}$ and $\widetilde{w}_{k+1}^{[r]}$ is derived with the same procedure. The definition of $\widetilde{\boldsymbol{w}}_{k+1}$ from (4.26) is reformulated as follows

$$\begin{aligned}
\widetilde{\boldsymbol{w}}_{k+1} &= \boldsymbol{w}_{k+1} + \theta W_{k+1} (\Sigma_k^{-1} - \theta W_{k+1})^{-1} \boldsymbol{w}_{k+1} \\
&= \boldsymbol{w}_{k+1} - (\Sigma_k^{-1} - \theta W_{k+1} - \Sigma_k^{-1})(\Sigma_k^{-1} - \theta W_{k+1})^{-1} \boldsymbol{w}_{k+1} \\
&= (I_q - \theta W_{k+1} \Sigma_k)^{-1} \boldsymbol{w}_{k+1} \,,
\end{aligned}$$

from which we can write

$$\widetilde{\boldsymbol{w}}_{k+1} = \boldsymbol{w}_{k+1} + \theta W_{k+1} \Sigma_k \widetilde{\boldsymbol{w}}_{k+1} \,. \tag{4.35}$$

Applying Leibniz's Theorem twice we get

$$\begin{aligned}
\widetilde{\boldsymbol{w}}_{k+1}^{[r]} &= \boldsymbol{w}_{k+1}^{[r]} + \theta (W_{k+1} \Sigma_k \widetilde{\boldsymbol{w}}_{k+1})^{[r]} + r (W_{k+1} \Sigma_k \widetilde{\boldsymbol{w}}_{k+1})^{[r-1]} \\
&= \boldsymbol{w}_{k+1}^{[r]} + \theta (W_{k+1} \Sigma_k \widetilde{\boldsymbol{w}}_{k+1})^{[r]} + r \sum_{j=0}^{r-1} \mathscr{C}_{r-1}^{j} W_{k+1}^{[j]} \Sigma_k \widetilde{\boldsymbol{w}}_{k+1}^{[r-1-j]},
\end{aligned}$$

which after evaluating $\theta = 0$ yields (4.33).

The procedure for $\widetilde{w}^{[r]}_{k+1}$ is similar. We formulate (4.27) as

$$\begin{aligned}\widetilde{w}_{k+1} &= w_{k+1} + \theta \boldsymbol{w}^{\mathrm{T}}_{k+1} W^{-1}_{k+1} W_{k+1} (\Sigma^{-1}_{k} - \theta W_{k+1})^{-1} \boldsymbol{w}_{k+1} \\ &= w_{k+1} - \boldsymbol{w}^{\mathrm{T}}_{k+1} W^{-1}_{k+1} \boldsymbol{w}_{k+1} + \boldsymbol{w}^{\mathrm{T}}_{k+1} W^{-1}_{k+1} (I_q - \theta W_{k+1} \Sigma_k)^{-1} \boldsymbol{w}_{k+1} \ ,\end{aligned}$$

where we ignored the term $-\frac{1}{2} \log F_k$ that gives rise to expression $H^{[r]}_t$ as already proven in [117]. From (4.35), writing $\boldsymbol{w}_{k+1}$ as a function of $\widetilde{\boldsymbol{w}}_{k+1}$ yields

$$\begin{aligned}\widetilde{w}_{k+1} &= w_{k+1} - \boldsymbol{w}^{\mathrm{T}}_{k+1} W^{-1}_{k+1} (I_q - \theta W_{k+1} \Sigma_k) \widetilde{\boldsymbol{w}}_{k+1} \\ &\qquad + \boldsymbol{w}^{\mathrm{T}}_{k+1} W^{-1}_{k+1} (I_q - \theta W_{k+1} \Sigma_k)^{-1} (I_q - \theta W_{k+1} \Sigma_k) \widetilde{\boldsymbol{w}}_{k+1} \\ &= w_{k+1} + \theta \boldsymbol{w}^{\mathrm{T}}_{k+1} \Sigma_k \widetilde{\boldsymbol{w}}_{k+1} \ .\end{aligned}$$

Applying Leibniz's Theorem twice and evaluating $\theta = 0$ yields (4.34).

□

The evaluation of the average of marginal cost-cumulants from (4.13) at time step k follows immediately as

$$\Xi(\xi_k) = \frac{1}{2} \xi^{\mathrm{T}}_k \widehat{W}_k \xi_k + \xi^{\mathrm{T}}_k \widehat{\boldsymbol{w}}_k + \widehat{w}_k \ , \tag{4.36}$$

where

$$\widehat{W}_k = \frac{1}{S} \sum_{s=1}^{S} \sum_{r=1}^{K} r \gamma_{r,s} W^{s[r-1]}_{k} \qquad \widehat{\boldsymbol{w}}_k = \frac{1}{S} \sum_{s=1}^{S} \sum_{r=1}^{K} r \gamma_{r,s} \boldsymbol{w}^{s[r-1]}_{k+1}$$

$$\widehat{w}_k = \frac{1}{S} \sum_{s=1}^{S} \sum_{r=1}^{K} r \gamma_{r,s} w^{s\,[r-1]}_{k}$$

where $W^{s[r]}_k$ and $\boldsymbol{w}^{s\,[r]}_k$ and $w^{s\,[r]}_k$ are the quadratic coefficients of the r-th cumulant resulting from applying lemma 4.2 in the s-th probability space constrained to the s-th marginal dynamics and $\gamma_{1,s} = 1$.

The solution to the cost cumulant control problem (4.13) is equivalent to the risk-sensitive solution: minimizing (4.36) w.r.t L_k and $\boldsymbol{l}_k$ at each step of the backwards recursion yields

$$\begin{aligned}L_k &= -(R_k + B^{\mathrm{T}}_k \widehat{\widetilde{W}}_{k+1} B_k)^{-1} (B^{\mathrm{T}}_k \widehat{\widetilde{W}}_{k+1} A_k) \\ \boldsymbol{l}_k &= -(R_k + B^{\mathrm{T}}_k \widehat{\widetilde{W}}_{k+1} B_k)^{-1} (B^{\mathrm{T}}_k \widehat{\widetilde{\boldsymbol{w}}}_{k+1} + \boldsymbol{r}_k) \ ,\end{aligned} \tag{4.37}$$

where

$$\widehat{\widetilde{W}}_k = \frac{1}{S} \sum_{s=1}^{S} \sum_{r=1}^{K} \gamma_{r,s} \widetilde{W}^{s[r-1]}_{k} \qquad \widehat{\widetilde{\boldsymbol{w}}}_k = \frac{1}{S} \sum_{s=1}^{S} \sum_{r=1}^{K} \gamma_{r,s} \widetilde{\boldsymbol{w}}^{s[r-1]}_{k}$$

where $\widetilde{W}^{s[r-1]}_{k+1}$ and $\widetilde{\boldsymbol{w}}^{s[r-1]}_{k+1}$ result from the computation of the r-th cumulant in the s-th probability space in (4.36).

The effect of weightings $\gamma_{r,s}$ on the resulting cost (4.36) is evident: risk-aversion is achieved by selecting $\gamma_{r,s} > 0$ and thereby increasing the resulting quadratic coefficients. Selecting $\gamma_{r,s} < 0$ has the opposite effect yielding a risk-seeking evaluation, i.e. 'less' cost than the expected value. Note that the existence of solutions (4.37) is only warranted if $(R_k + B_k^{\mathrm{T}}\widehat{W_{k+1}}B_k) > 0$. Although this condition imposes no constraints in terms of risk aversion, negative $\gamma_{r,s}$s must be selected such that the positive definiteness of $(R_k + B_k^{\mathrm{T}}\widehat{W_{k+1}}B_k)$ holds.

4.4 Main Iteration

A numerical approximation that computes locally optimal solutions in nonlinear and non-quadratic problems requires a procedure that iteratively approximates and updates a nominal trajectory. The main iteration of our approach follows the iLQG algorithm presented in [**?**, 151]. In this section, we summarize it pointing out the subtle changes that arise due to the different problem setting. The resulting algorithm is either an iterative Linear Exponential Quadratic Regulator (iLEQR) for the risk-sensitive case or an iterative K-Cost Cumulant Regulator (iKCCR) for the cost cumulant optimization.

In order to find a local optimum and given an initial state ξ_0^*, the algorithm iterates around the nominal control trajectory $\overline{\boldsymbol{u}}^*_{0\cdots T-1}$ by calculating the optimal control deviations that improve the expected performance. At the i-th iteration, the optimal solution is denoted $\boldsymbol{u}^{*(i)}_{0\cdots T-1} = \overline{\boldsymbol{u}}^{*(i)}_{0\cdots T-1} + L^{(i)}_{0\cdots T-1}\xi_{0\cdots T-1}$ and is updated to obtain the next $\boldsymbol{u}^{*(i+1)}_{0\cdots T-1}$ following the next steps:

1) The corresponding state trajectory $\xi^*_{0\cdots T}$ is computed simulating the discretized dynamics, for instance by Euler integration .i.e. $\xi^*_{k+1} = \xi^*_k + \Delta f(\xi^*_k, \overline{\boldsymbol{u}}^{*(i)}_k)$ or any other approach such as the Runge-Kutta method.

2) The dynamics are linearized and the costs quadratically approximated around $(\xi^*_{0\cdots T}, \overline{\boldsymbol{u}}^*_{0\cdots T-1})$, obtaining an LQ approximation of state and control deviations, i.e. $\xi_k = \delta\xi^*_k - \xi^*_k$ and $\boldsymbol{u}_k = \delta\boldsymbol{u}^*_k - \boldsymbol{u}^*_k$ as explained in Section 4.1.1.

3) Depending on the desired optimization criteria, the optimal deviations law $\boldsymbol{u}_k = \boldsymbol{l}_k + L_k\xi_k$ is computed by means of either the risk-sensitive (4.30) or cost-cumulant solution (4.37).

4) As the solution $\boldsymbol{u}_k$ is only locally optimal, if the feedforward term of optimal law computed in 3) only holds in the close vicinity of the current nominal trajectory, deviations $\boldsymbol{l}_k$ might yield solutions with higher cost than the current one. To cope with this issue, a line search algorithm aims for an adapted step in the feedforward component of $\boldsymbol{u}_k$ that yields a policy improvement. Starting with a line search parameter $\alpha = 1$, a potential policy update is given by $\boldsymbol{u}^{*(i+1)}_k(\alpha) = \overline{\boldsymbol{u}}^{*(i)}_k + \alpha\boldsymbol{l}_k + L^{(i+1)}_k\xi_k$ and its expected performance is evaluated. If it improves, the new policy is accepted. If not, the line search parameter is halved, i.e. $\alpha = \frac{1}{2}\alpha$ and the corresponding expected performance for $\boldsymbol{u}^{*(i+1)}_{0\cdot T-1}(\alpha)$ is evaluated again until improvement. If all $\boldsymbol{l}_{0\cdots T-1}$ are close to 0, the algorithm ends. Otherwise, the next iteration starts going back to step 1)

with $\boldsymbol{u}^{*(i+1)}_{0\cdots T-1} = \boldsymbol{u}^{*(i+1)}_{0\cdots T-1}(\alpha)$. This algorithm ensures convergence to a locally optimal control policy [160].

Note that the evaluation of the expected performance in step 4) is not straightforward. In fact, an analytical expression of (4.13) or (4.14) for an arbitrary J is rarely available. As an alternative, an LQ approximation of the cost around the new trajectory is obtained as in step 2) and its expected performance is computed by means of lemma 4.1 or 4.2 respectively. For this computation the nominal trajectory is $\boldsymbol{u}^{*(i+1)}_k(\alpha) = \boldsymbol{u}^{*(i+1)}_k + L_k\xi_k$ and the control law to be considered has no feedforward term, i.e. $\boldsymbol{l}_k = 0$, as the deviations have been already applied. This approximation of the cost works well when the nominal trajectory is close to the local optimum and feedforward steps $\boldsymbol{l}_k$ are close to zero. However, if this is not the case and high deviations are still involved, the feedback matrix L_k used in the approximation may lead to inaccurate results as second order information may vary significantly. In order to avoid this issue, a two-step optimization is applied. First, a solution close to the local optimum is found evaluating only the feedforward trajectory and hence $L_k = 0$. Once the feedforward trajectory converges, the quadratic feedback problem is solved. This second solution converges in few iterations, as the local optimum is close to the starting nominal trajectory.

4.5 Uncertain Static Parameters

In our problem setting from Section 4.1 we study the case where the goal to be tracked or the obstacle to be avoided are governed by uncertain dynamics. However, many robotic settings, especially when sensing is involved, comprise *static* uncertain obstacles or goal. A problem with such prerequisites has a slightly different treatment, as the goal and obstacle diffusions are not present in the solution any more. Preceding the optimization, time independence allows cost terms depending on the corresponding uncertain elements to be replaced by expressions capturing the desired cumulants. To illustrate this, in this section we analyze two examples where the cost function considers a static goal and a static obstacle. Note that the solutions presented here are also valid for piecewise constant variables, i.e. a discretized uncertain trajectory of desired set points and/or obstacles. These problems arise in many PbD settings when task models are obtained as generalized trajectories, for example in [63] or [70].

Consider a cost functional in the form

$$J(\boldsymbol{u}(\cdot)) = h_g(\boldsymbol{x}^r(T_c), \boldsymbol{x}^g(T_c)) + \int_{t=0}^{T_c} h_g(\boldsymbol{x}^r(t), \boldsymbol{x}^g(t)) + h_o(\boldsymbol{x}^r(t), \boldsymbol{x}^o(t)) + h_{so}(\boldsymbol{x}^r(t), \boldsymbol{x}^{so}(t)) + h_u(\boldsymbol{u}(t))dt\ , \tag{4.38}$$

where $h_g(\boldsymbol{x}^r(t), \boldsymbol{x}^g(t))$ penalizes the distance to desired configurations, $h_u(\boldsymbol{u}^r(t))$ penalizes control efforts and $h_o(\boldsymbol{x}^r(t), \boldsymbol{x}^o(t))$ favors configurations distant to dynamic obstacles and $h_{so}(\boldsymbol{x}^r(t), \boldsymbol{x}^{so}(t))$ to static obstacles. Note that h_o and h_{so} represent the same performance measure but we explicitly defined different terms here for later clarity.

4.5.1 Static goal

Let $\boldsymbol{x}^g$ be normally distributed and constant such that $\boldsymbol{x}^g(t) \sim \mathcal{N}(\boldsymbol{x}^g|\boldsymbol{\mu}^g, \Sigma^g)$. A typical convex functional for goal-oriented behavior synthesis is the quadratic functional

$$h_g(\boldsymbol{x}^r(t), \boldsymbol{x}^g) = \frac{1}{2}(\boldsymbol{x}^r(t) - \boldsymbol{x}^g)^{\mathrm{T}} Q (\boldsymbol{x}^r(t) - \boldsymbol{x}^g) ,$$

where $Q > 0$. In this case, a marginal analysis of the cumulant generating function of (4.38) in probability space $(\Omega_g, \mathscr{F}_g, \mathsf{P}_g)$ is only relevant concerning $h_g(\boldsymbol{x}^r(t), \boldsymbol{x}^g)$ (any other terms are only present in the expected value as they are constant). Hence, considering an alternative $\widehat{h}_g$ in (4.38) in terms of the desired cumulants yields a deterministic problem w.r.t to the goal's uncertainty. Its cumulant generating function is given by

$$\begin{aligned}\log \mathsf{E}_{\mathsf{P}_g}[\exp\{\theta h_g(\boldsymbol{x}^r(t), \boldsymbol{x}^g)\}] = |I_{q_g} - \theta Q \Sigma^g| \\ + \frac{1}{2}\theta(\boldsymbol{x}^r(t) - \boldsymbol{\mu}^g)^{\mathrm{T}}(I - \theta Q \Sigma^g)^{-1} Q (\boldsymbol{x}^r(t) - \boldsymbol{\mu}^g) .\end{aligned} \tag{4.39}$$

By applying (4.11) and ignoring the constant term which does not depend on $\boldsymbol{x}^r(t)$, the calculation of any cumulant is straightforward. For example, the mean and variance of h_g are given by

$$\begin{aligned}\mathsf{E}_{\mathsf{P}_g}[h_g] &= \frac{1}{2}(\boldsymbol{x}^r(t) - \boldsymbol{\mu}^g)^{\mathrm{T}} Q (\boldsymbol{x}^r(t) - \boldsymbol{\mu}^g) + const \\ \mathsf{Var}_{\mathsf{P}_g}[h_g] &= \frac{1}{2}(\boldsymbol{x}^r(t) - \boldsymbol{\mu}^g)^{\mathrm{T}} Q \Sigma^g Q (\boldsymbol{x}^r(t) - \boldsymbol{\mu}^g) + const\end{aligned}$$

Any desired linear combination of cumulants is a valid substitute $\widehat{h}_g(\boldsymbol{x}^r(t), \boldsymbol{x}^g)$. Another alternative is to directly use the risk-sensitive functional of h_g

$$\widehat{h}_g(\boldsymbol{x}^r(t), \boldsymbol{x}^g) = (\boldsymbol{x}^r - \boldsymbol{\mu}^g)^{\mathrm{T}}(I - \theta Q \Sigma^g)^{-1} Q (\boldsymbol{x}^r - \boldsymbol{\mu}^g) + const . \tag{4.40}$$

It is remarkable that previous work rely on the Mahalanobis distance in order to approach this problem [9, 161], i.e.

$$\frac{1}{2}(\boldsymbol{x}^r(t) - \boldsymbol{\mu}^g)^{\mathrm{T}}(\Sigma^g)^{-1} Q (\boldsymbol{x}^r(t) - \boldsymbol{\mu}^g) .$$

This expression lacks a definition when the goal tends to the deterministic case, i.e. $\Sigma^g \to 0_{q_g}$. Adding a regularization quantity given by the identity matrix to avoid this issue yields

$$\frac{1}{2}(\boldsymbol{x}^r(t) - \boldsymbol{\mu}^g)^{\mathrm{T}}(I + \Sigma^g)^{-1} Q (\boldsymbol{x}^r(t) - \boldsymbol{\mu}^g) . \tag{4.41}$$

Observing (4.40), it becomes apparent that (4.41) is a specific instance of a risk-seeking evaluation of h_g, where $\theta = -1$ and the goal variance is $Q^{-1}\Sigma^g$.

4.5.2 Static obstacle

Let $\boldsymbol{x}^{so}$ be normally distributed and constant such that $\boldsymbol{x}^{so}(t) \sim \mathcal{N}(\boldsymbol{x}^{so}(t)|\boldsymbol{\mu}^{so}, \Sigma^{so})$. An effective convex and continuously differentiable cost functional when dealing with obstacles is given by

$$h_{so}(\boldsymbol{x}^r(t), \boldsymbol{x}^{so}) = \alpha \exp\{-0.5(\boldsymbol{x}^r(t) - \boldsymbol{x}^{so})^{\mathrm{T}} Q (\boldsymbol{x}^r(t) - \boldsymbol{x}^{so})\} ,$$

where $\alpha \in \mathbb{R}$ is a weighting factor. In this case, its cumulant generating function has no closed form. However, its expected value is given by

$$\mathsf{E}_{\mathsf{P}_{so}}[h_{so}] = const \cdot \alpha \cdot \exp\{(\boldsymbol{x}^r(t) - \boldsymbol{\mu}^{so})^{\mathrm{T}} (I - Q\Sigma^{so})^{-1} Q (\boldsymbol{x}^r(t) - \boldsymbol{\mu}^{so})\} . \tag{4.42}$$

It is also straightforward to show that any higher order moments of h_{so} do not depend on $\boldsymbol{x}^r(t)$; the expected value is the only cumulant of interest for the control problem. Hence $\widehat{h}_{so} = \mathsf{E}_{\mathsf{P}_{so}}[h_{so}]$ is the only uncertainty-dependent treatment of this problem based on cumulants or moments. This result is equivalent to the generalized binary saturating cost considered in [83] as an attractor instead of a repelling obstacle.

4.6 Evaluation

In order to illustrate the peculiarities of the proposed approach we study first the family of solutions of a scalar discrete-time LQ regulator problem. Simulations on a point-mass robot follow, where we consider first stochastic goal dynamics, later complemented with both static and stochastic dynamic obstacles. Similar implementations of a car-like robot and a two-link manipulator validate the proposed approach considering also non-linear plants.

4.6.1 Linear Quadratic Feedback Regulator

For simplicity, we consider first a 1-dimensional discrete-time mass-damper system with quadratic costs in a infinite horizon feedback regulator scenario. The one-dimensional plant dynamics of a point-mass with s additive uncertainty sources is given by

$$\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta \\ 0 & 1 - m^{-1} d\Delta \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & m^{-1}\Delta \end{bmatrix} u_k + \sum_{s=1}^{S} \epsilon_k^s , \tag{4.43}$$

where $x \in \mathrm{R}$ is the point mass position, m and d its mass and viscous friction respectively and the control input u is the applied force. The discrete-time cost function is

$$J = \sum_{k=1}^{k=\infty} \xi_k^{\mathrm{T}} Q \xi_k + \boldsymbol{u}_k^{\mathrm{T}} R \boldsymbol{u}_k , \tag{4.44}$$

where $\xi = \begin{bmatrix} x & \dot{x} \end{bmatrix}^{\mathrm{T}}$ Q and R are the weighting matrix defining the trade-off between regulation precision and control effort respectively. The control law is constrained to the feedback term, i.e. $u_k = -[L_{xk}\ L_{\dot{x}k}]\xi$ and therefore $\boldsymbol{l}_k = 0$. The infinite horizon solution is approximated iteratively until convergence to a stationary cost-to-go, yielding a single stationary feedback gain. In the following results we fixed $m = 1$ kg, $d = 1$ Ns/m, $R = I_2$, $Q = I_2$ and $\Delta = 10^{-3}$.

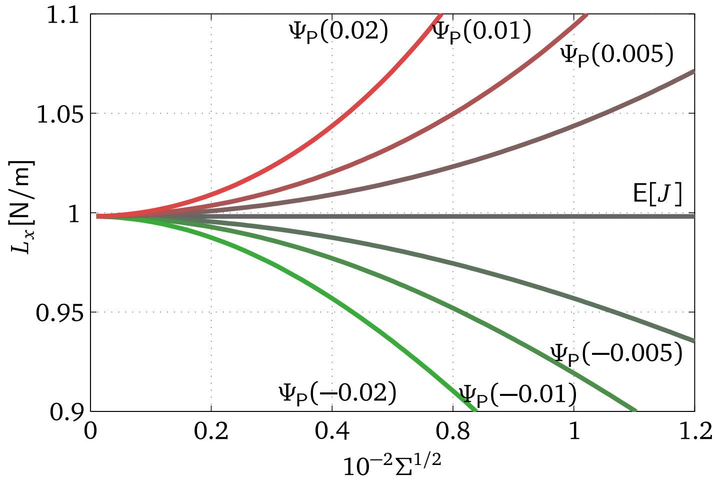


Figure 4.1: Optimal positional gains w.r.t variance square root $\Sigma^{1/2}$ for a linear quadratic problem with dynamics (4.43), cost (4.44) with a single uncertainty source for several risk-sensitive controllers with different risk-sensitivities in comparison with the expected cost solution.

We first consider a single uncertainty source and therefore $s = 1$ in (4.43). The positional feedback gains L_x w.r.t the variance square root are depicted in Fig. 4.1 for a range of risk-sensitive controllers and a single uncertainty source. The expected cost solution depicted by the gray solid line ignores any cost variability and remains constant neglecting the level of uncertainty. However, risk-aversion, i.e. $\theta > 0$, increases the resulting gains together with the variance; variability is expected to diminish performance and feedback gains are increased accordingly. In contrast, risk-seekingness, i.e. $\theta < 0$ interprets cost variability as a cost discounting agent consequently reducing both cost and gain. Equivalent results are illustrated in Fig. 4.2 for several cost-cumulant controllers. Aiming for a fair comparison, cumulant factors $\gamma_{i,s}$ were set according to the power of the respective cumulant's statistical order. The influence of risk-aversion and risk-seekingness are similar to the risk-sensitive case: positive factors $\gamma_{i,s} > 0$ increase cost with the respective cumulant yielding higher gains while negative factors $\gamma_{i,s} > 0$ have opposite effect. The role of the cumulant's order in the solution is also relevant. As uncertainty increases, the corresponding optimal feedback differs from the expected cost according to the power of the respective cumulant's statistical order.

In order to explore the average problems, we extend the previous evaluation to two identical uncertainty sources Σ_1 and Σ_2 respectively. The positional feedback gains L_x w.r.t to both factors are depicted in Fig. 4.3 for 4 different combinations of risk-sensitive and cost-cumulant controllers. In all cases, uncertainty source 1 and 2 are evaluated with equivalent weighting factors but in a risk-averse and risk-sensitive manner respectively. Note that the resulting gain for the classical E[J] solution is, as before, depicted in gray and corresponds to $L_x = 0.998N/m$. Fig. 4.3(a) shows the risk-sensitive solution. The 2 marginal problems are complementary: as Σ_1 increases, so does the resulting feedback gain while as Σ_2 decreases the gain diminishes. As a result, when both variance factors are at the same level, the effects of both marginal computations balance out yielding the expected cost solution. When a factor is higher than the other the corresponding marginal effect prevails. Fig. 4.3(b), Fig. 4.3(c) and Fig. 4.3(d) show similar results considering only the second, third and fourth

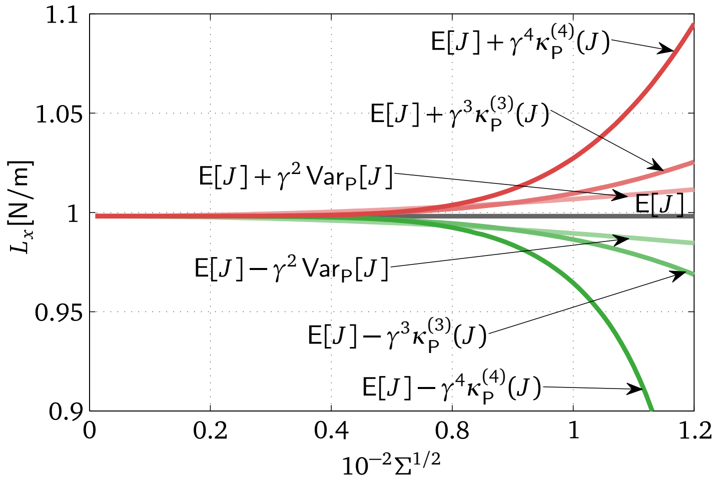
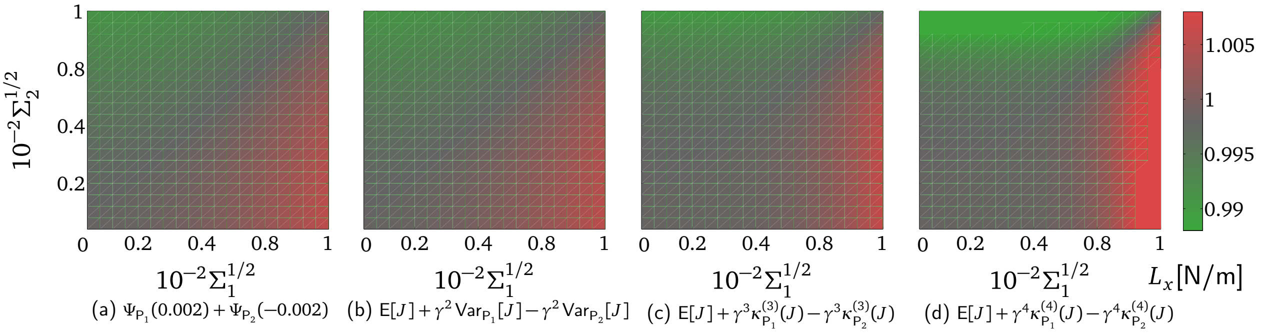


Figure 4.2: Optimal positional gains w.r.t variance square root $\Sigma^{1/2}$ for a linear quadratic problem with dynamics (4.43), cost (4.44) with a single uncertainty source for several cost-cumulant controllers with different weighting factors in comparison with the expected cost solution.



(a) $\Psi_{\mathrm{P}_1}(0.002) + \Psi_{\mathrm{P}_2}(-0.002)$ (b) $\mathrm{E}[J] + \gamma^2 \mathrm{Var}_{\mathrm{P}_1}[J] - \gamma^2 \mathrm{Var}_{\mathrm{P}_2}[J]$ (c) $\mathrm{E}[J] + \gamma^3 \kappa_{\mathrm{P}_1}^{(3)}(J) - \gamma^3 \kappa_{\mathrm{P}_2}^{(3)}(J)$ (d) $\mathrm{E}[J] + \gamma^4 \kappa_{\mathrm{P}_1}^{(4)}(J) - \gamma^4 \kappa_{\mathrm{P}_2}^{(4)}(J)$

Figure 4.3: Optimal positional gains w.r.t variances square root $\Sigma_1^{1/2}$ and $\Sigma_2^{1/2}$ for a linear quadratic problem with dynamics (4.43), cost (4.44) and two uncertainty sources for different risk-sensitive and cost-cumulant controllers, where $\gamma = \sqrt{0}.1$.

cumulant respectively. As before, when higher order cumulants are considered, gains are adapted to the power of the respective cumulant's statistical order. This explains the remarkably high or low gains as Σ_1 and Σ_2 approach their highest value in Fig. 4.3(d) for the fourth cumulant. Note also that the mean-variance solution from Fig. 4.3(b) is almost identical to the risk-sensitive solution from Fig. 4.3(a). In fact, as appointed in [155], when $\theta \approx 0$, the risk-sensitive cost tends to a mean variance optimization as higher order terms cancel out. To observe this equivalence, we set the factor of the mean variance optimizer such that it coincides with the variance factor assotiated with the risk-sensitive solution as given in the series expansion (4.14), i.e. $\gamma^2 = \theta/2$. However, this only holds for low risk-sensitivities θ_s: when higher values are considered, higher order terms play a relevant role and the equivalence does not hold anymore.

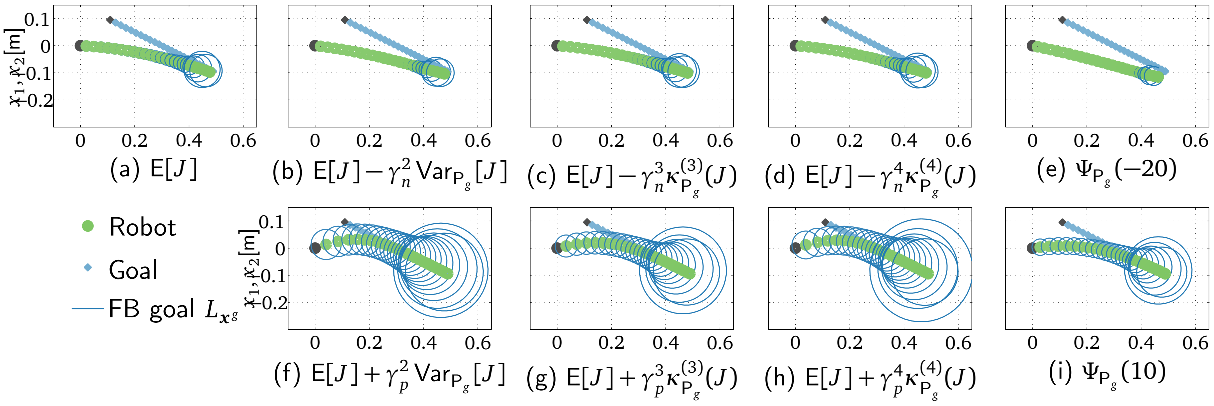


Figure 4.4: Optimal trajectories and feedback gains for a 2D point mass damper robot $\boldsymbol{x}^r$ tracking goal $\boldsymbol{x}^g$ with noisy passive mass-damper system dynamics and where $\gamma_n = 2$ and $\gamma_p = 7$ for a horizon of $T_c = 0.5s$. Initial states of robot and goal are $\boldsymbol{x}_0^r = [0\ 0]$, $\dot{\boldsymbol{x}}_0^r = [0\ 0]$, $\boldsymbol{x}_0^g = [0.1\ 0.1]$ and $\dot{\boldsymbol{x}}_0^g = [1\ -0.5]$. Results are shown every 0.02s.

4.6.2 2D Point-Mass robot

To illustrate the influence of high-order cost statistics in simple robotic scenarios in the following we apply the algorithm described in Section 4.4 to simple robotic scenarios. The consequences of risk-seeking and risk-averse cost evaluations are depicted considering first stochastic goal dynamics, then stochastic obstacle dynamics and finally both cases at the same time. An additional scenario complements these solutions by adding static uncertain obstacles evaluated as explained in Section 4.5.

Consider a two-dimensional point robot with position $\boldsymbol{x}^r \in \mathrm{R}^2$, no orientation and second-order dynamics given by the mass-damper system

$$M\ddot{\boldsymbol{x}}^r + D\dot{\boldsymbol{x}}^r = \boldsymbol{u}^* \ .$$

The control task consists of following a goal $\boldsymbol{x}^g \in \mathrm{R}^2$ whose stochastic dynamics are constrained to a passive mass-damper system with noise. This is an equivalent scenario to the problem of tracking a learned DMP, explained in Section 2.2.2, where an additional uncertain forcing term drives the goal. The error state of the system is given by

$$\xi^* = [(\boldsymbol{x}^g - \boldsymbol{x}^r)^{\mathrm{T}}\ (\dot{\boldsymbol{x}}^g - \dot{\boldsymbol{x}}^r)^{\mathrm{T}}]^{\mathrm{T}} \ .$$

The cost function takes the form (4.38), where

$$h_g = \frac{1}{2}\xi^{*\mathrm{T}} Q_g \xi^* \quad h_u = \frac{1}{2}\boldsymbol{u}^{*\mathrm{T}} R \boldsymbol{u}^* \quad h_o = 0 \quad h_{so} = 0 \ .$$

The optimal policy takes the form $\boldsymbol{u}^* = \overline{\boldsymbol{u}}^* + [L_{\boldsymbol{x}^g}\ L_{\dot{\boldsymbol{x}}^g}]\xi$. In the following results, parameters were fixed to $M = I_2$ kg, $D = I_2$ Ns/m, $R = 10^{-2} I_2$, $Q_g = diag\{\ 1\ \ 1\ \ 0.1\ \ 0.1\ \}$, $\Delta = 10^{-2}$ and $T_c = 0.5s$. Goal dynamics are assumed to have infinitesimal variance $\mathcal{G}^g = I_4$ and identical mass and damping to the robot's.

Optimal trajectories and positional gains $L_{\boldsymbol{x}^g}$ for several cost cumulant and risk-sensitive controllers are depicted in Fig. 4.4. Ellipses represent the corresponding feedback gain matrices in terms of their eigenvectors. The solution corresponding to the expected cost depicted in Fig. 4.4(a) serves as the risk-neutral reference. Fig. 4.4(e) and Fig. 4.4(i) show risk-sensitive solutions in their seeking and averse variants respectively. The risk-seeking policy not only adopts significantly lower feedback gains but also adapts its feedforward trajectory to a less accurate positional tracking. This policy is desirable when goal uncertainty suggests more flexibility, for instance in PbD settings [73]. The risk-averse solution has the opposite effect, tracking goal dynamics more aggressively as well as increasing feedback gains. This behavior is more in accordance with navigation scenarios where uncertainty may hinder performance [150]. Cost cumulant solutions are shown in Fig. 4.4(f), Fig. 4.4(g), Fig. 4.4(h) for the second, third and fourth cumulant in their risk-averse variants. All three cases are similar: the feedforward trajectory is significantly adapted aiming for more accurate tracking together with higher feedback gains. Their risk-seeking counterparts depicted in Fig. 4.4(b), Fig. 4.4(c), Fig. 4.4(d) show little difference w.r.t the risk-neutral policy Fig. 4.4(a). Although one could decrease the respective negative weightings to boost their effect, the presented solutions are close to the limit yielding a positive-definite inverted term in lemma (4.37).

To evaluate the influence of high-order statistics when uncertain dynamic obstacles are present, the considered dynamics are extended to include a dynamic obstacle with position $\boldsymbol{x}^o$ with passive mass-damper dynamics with noise. This is a similar setting to the one considered in [157]. The error state of the system is given by

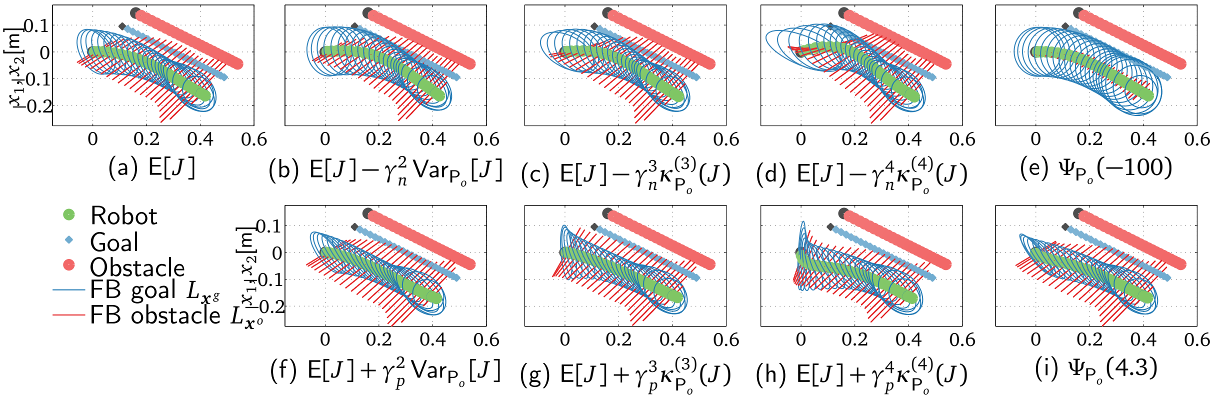
$$\xi^* = [(\boldsymbol{x}^g - \boldsymbol{x}^r)^{\mathrm{T}} \; (\dot{\boldsymbol{x}}^g - \dot{\boldsymbol{x}}^r)^{\mathrm{T}} \; (\boldsymbol{x}^o - \boldsymbol{x}^r)^{\mathrm{T}} \; (\dot{\boldsymbol{x}}^o - \dot{\boldsymbol{x}}^r)^{\mathrm{T}}]^{\mathrm{T}} \,. \tag{4.45}$$

The cost remains identical to the previous setting except for the dynamic obstacle-related term, which is set to $h_o = w \exp\{-0.5(\boldsymbol{x}^r - \boldsymbol{x}^o)^{\mathrm{T}} Q_o (\boldsymbol{x}^r - \boldsymbol{x}^o)\}$, with $Q_o = 200 I_2$ and $w = 0.1$. The optimal policy is now

$$\boldsymbol{u}^* = \overline{\boldsymbol{u}}^* + \begin{bmatrix} L_{\boldsymbol{x}^g} & L_{\dot{\boldsymbol{x}}^g} \\ L_{\boldsymbol{x}^o} & L_{\dot{\boldsymbol{x}}^o} \end{bmatrix} \xi.$$

Obstacle dynamics are assumed to have infinitesimal variance $\mathscr{G}^o = I_4$ and identical mass and damping to the robot's.

The optimal trajectory, positional goal gain $L_{\boldsymbol{x}^g}$ and obstacle gain $L_{\boldsymbol{x}^o}$ for several cost cumulant and risk-sensitive controllers are illustrated in Fig. 4.5 considering only the marginal variability produced by the obstacle's diffusion process. The standard risk-neutral policy is shown in Fig. 4.5(a). Risk-sensitive solutions are depicted in Fig. 4.5(e) and Fig. 4.5(i) for the risk-seeking and the risk-averse variants. The risk-seeking solution decreases obstacle gains $L_{\boldsymbol{x}^o}$ dramatically and as a consequence positional gains increase. In contrast, the risk-averse solution adapts feedforward terms, choosing a trajectory further away to the obstacle. Cost-cumulant solutions are illustrated in Fig. 4.5(f), Fig. 4.5(g), Fig. 4.5(h) for risk-averse variants of the second, third and fourth cumulant respectively. Especially for the fourth cumulant, solutions show adapted trajectories aiming for configurations distant to the obstacle. The risk-seeking cost-cumulant solutions are shown in Fig. 4.5(b), Fig. 4.5(c), Fig. 4.5(d) for the second, third and fourth cumulant respectively. In contrast to the risk-sensitive solution



(a) $\mathsf{E}[J]$ (b) $\mathsf{E}[J] - \gamma_n^2 \mathsf{Var}_{\mathsf{P}_o}[J]$ (c) $\mathsf{E}[J] - \gamma_n^3 \kappa^{(3)}_{\mathsf{P}_o}(J)$ (d) $\mathsf{E}[J] - \gamma_n^4 \kappa^{(4)}_{\mathsf{P}_o}(J)$ (e) $\Psi_{\mathsf{P}_o}(-100)$

(f) $\mathsf{E}[J] + \gamma_p^2 \mathsf{Var}_{\mathsf{P}_o}[J]$ (g) $\mathsf{E}[J] + \gamma_p^3 \kappa^{(3)}_{\mathsf{P}_o}(J)$ (h) $\mathsf{E}[J] + \gamma_p^4 \kappa^{(4)}_{\mathsf{P}_o}(J)$ (i) $\Psi_{\mathsf{P}_o}(4.3)$

Figure 4.5: Optimal trajectories and feedback gains for a 2D point mass damper robot $\boldsymbol{x}^r$ tracking goal $\boldsymbol{x}^g$ with passive mass-damper system dynamics and avoiding obstacle $\boldsymbol{x}^o$ with noisy passive mass-damper system dynamics and where $\gamma_n = 1.9$ and $\gamma_p = 3$ for a horizon of $T_c = 0.5s$. Initial states of robot, goal and obstacle are $\boldsymbol{x}_0^r = [0\ 0]$, $\dot{\boldsymbol{x}}_0^r = [0\ 0]$, $\boldsymbol{x}_0^g = [0.1\ 0.1]$, $\dot{\boldsymbol{x}}_0^g = [1\ -0.5]$, $\boldsymbol{x}_0^o = [0.15\ 0.15]$ and $\dot{\boldsymbol{x}}_0^o = [1\ -0.5]$. Results are shown every 0.02s.

from Fig. 4.5(e), feedforward terms are significantly adapted obtaining a trajectory closer to the obstacle, especially for the fourth cumulant. Note that, with obstacle avoidance in mind, risk-aversion is a desirable property as it diminishes the risk of collisions.

We now explore solutions considering both the goal's and the obstacle's marginal variability. In the following, as goal variability expresses uncertainty about the desired trajectory to follow, we consider it in a risk-seeking way. In contrast, variability produced by the obstacle's uncertainty may result in potential collision and is evaluated in a risk-averse manner. The results for this setting for a risk-sensitive controller and a mean-variance controller are depicted in Fig. 4.6. The mean-variance solution is depicted in Fig. 4.6(b), which clearly shows a different trajectory to the risk-neutral one from Fig. 4.6(a). It is also interesting to compare this solution with Fig. 4.5(f), as both policies consider the same risk-averse marginal obstacle variability $\gamma_p^2 \mathsf{Var}_{\mathsf{P}_o}[J]$. The inclusion of the risk-seeking marginal goal variability $-\gamma_n^2 \mathsf{Var}_{\mathsf{P}_o}[J]$ demands less accurate goal tracking enabling a more pronounced adaptation to the obstacle's stochasticity. Similar although less remarkable are the results for the risk-sensitive optimization shown in Fig. 4.6(c) in comparison with Fig. 4.5(i) as the resulting trajectory avoids the obstacle trajectory more notably.

To envisage a more realistic setting we let the goal and obstacle mass-damper systems be driven by external forces $\boldsymbol{u}_g = \left[\cos(2x_1^g) \quad \sin(2x_2^g)\right]^{\mathrm{T}}$ and $\boldsymbol{u}_o = \left[\cos(10x_1^o) \quad \sin(10x_2^o)\right]^{\mathrm{T}}$. We further consider static normally distributed obstacles in the scene by setting the static obstacle term h_{so} from (4.38) to the expected value of the weighted exponential decay (4.42) as explained in Section 4.5, with $Q = I_2$ and $w = 0.1$. The resulting trajectories are shown in Fig. 4.7(b) for the mean-variance case in comparison with the risk-neutral policy from Fig. 4.7(a). On one side, risk-aversion towards obstacle variability yields a pronounced deviation at the beginning of the simulation due to the proximity of the obstacle. As a result, obstacle feedback gains are almost nullified. On the other side risk-seekingness w.r.t the goal variability favors less accurate tracking enabling better obstacle avoidance. The influ-

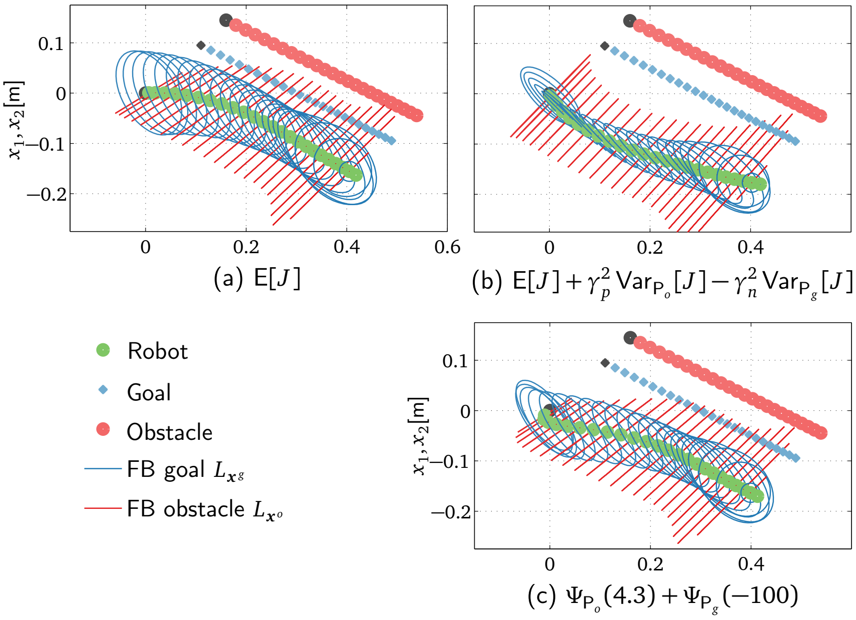


Figure 4.6: Optimal trajectories and feedback gains of the same problem from Fig.4.5 considering both the goal's and the obstacle's marginal variability, where $\gamma_n = 3.5$ and $\gamma_p = 3$. Results are shown every 0.02s.

ence of the static obstacles' uncertainty is also evident; the one with low variance centered at $(-0.3, 0.8)$ produces almost no deviations from the expected goal trajectory in comparison with the obstacle with high variance centered at $(-0.6, 0.8)$. In order to illustrate the influence of the uncertainty level in the resulting plocy, Fig. 4.8 shows solutions for the same setting but with different infinitesimal variance. From Fig. 4.8(a) to Fig. 4.8(d), decreasing uncertainty for both obstacle and goal dynamics are considered, showing results with deviations between Fig. 4.7(b) and Fig. 4.7(a). In fact, as noise levels tend to zero, results approach the risk-neutral solution as shown by the almost deterministic setting from Fig. 4.8(d). This suggests that the application of the proposed approach is a natural and systematic method of considering uncertainty.

4.6.3 Non-linear plants

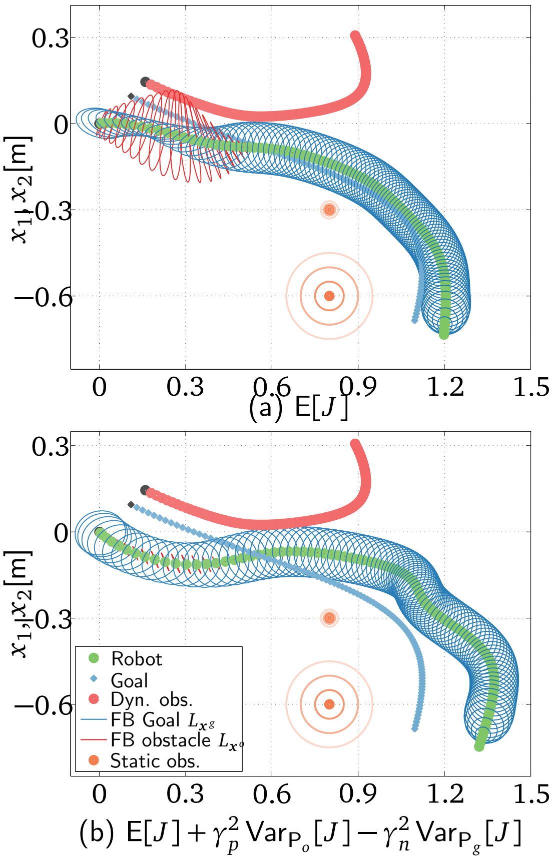
In order to evaluate the validity of the algorithm when non-linear plants are considered, similar settings to the one presented in Fig. 4.7 are applied to a two-link manipulator and a car-like robot.

Two-link manipulator

Consider a torque-controlled arm with two joints moving in the horizontal plane with inverse dynamics

$$\mathscr{M}(\theta)\ddot{\theta} + \mathscr{C}(\theta, \dot{\theta}) + \mathscr{B}\dot{\theta} = \boldsymbol{\tau} \ ,$$

where $\theta \in \mathrm{R}^2$ are the joint angles, $\mathscr{M}(\theta)$ is the inertia matrix, $\mathscr{C}(\theta, \dot{\theta})$ is the vector of centripetal and Coriolis forces, $\mathscr{B}$ is the joint friction matrix and $\boldsymbol{\tau} \in \mathrm{R}^2$ are the joint torques.



(a) E[J]

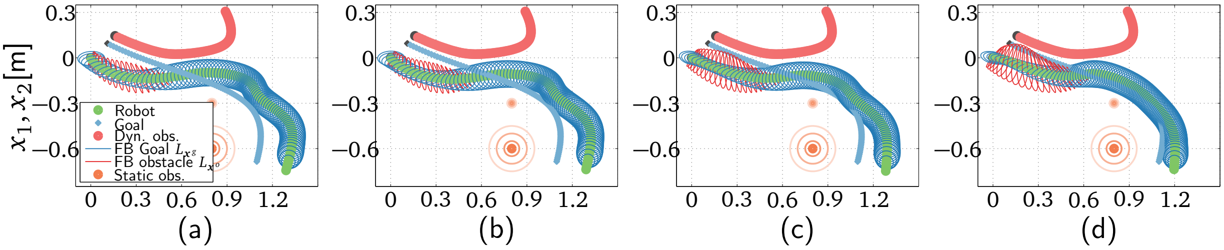
(b) E[J]$+\gamma_p^2$Var$_{\mathrm{P}_o}$[J]$-\gamma_n^2$Var$_{\mathrm{P}_g}$[J]

Figure 4.7: Optimal trajectories and feedback gains for a 2D point mass damper robot $\boldsymbol{x}^r$ tracking goal $\boldsymbol{x}^g$ with mass-damper system dynamics and avoiding obstacle $\boldsymbol{x}^o$ with noisy mass-damper system dynamics including static obstacles and where $\gamma_n = 2$ and $\gamma_p = 4$ for a horizon of $T_c = 2s$. The two static obstacles considered are normally distributed, centered in $(-0.3, 0.8)$ and $(-0.6, 0.8)$ with covariance matrices $0.1I_2$ and $0.5I_2$ respectively. Initial states of robot, goal and obstacle are $\boldsymbol{x}_0^r = [0\ 0]$, $\dot{\boldsymbol{x}}_0^r = [0\ 0]$, $\boldsymbol{x}_0^g = [0.1\ 0.1]$, $\dot{\boldsymbol{x}}_0^g = [1\ -0.5]$, $\boldsymbol{x}_0^o = [0.15\ 0.15]$ and $\dot{\boldsymbol{x}}_0^o = [1\ -0.5]$. Results are shown every 0.02s.

Following [**?**] we set the mass of each link to $m1 = 1.4$kg and $m2 = 1.1$kg, the length of each link to $l1 = 0.3$m and $l1 = 0.33$, the moments of inertia to $I_1 = 0.025\ \mathrm{kg}\cdot\mathrm{m}^2$ and $I_2 = 0.045\ \mathrm{kg}\cdot\mathrm{m}^2$ and we assume the center of mass of each link is placed at the link's center. The joint friction matrix is set to

$$\mathscr{B} = \begin{bmatrix} 0.05 & 0.025 \\ 0.025 & 0.05 \end{bmatrix}.$$

In line with the previous subsection, the control task consists of tracking a goal $\boldsymbol{x}^g \in \mathrm{R}^2$ with uncertain passive mass-damper dynamics while avoiding a dynamic obstacle $\boldsymbol{x}^o \in \mathrm{R}^2$ with similar dynamics. The state of the system is given by (4.45), where $\boldsymbol{x}^r$ are now the workspace coordinates of the manipulator and the control input $\boldsymbol{u}^*$ are the joint torques in workspace coordinates. The cost function is identical to the previous subsection with the same parameters.



(a) (b) (c) (d)

Figure 4.8: Optimal trajectories and feedback gains for the same setting from Fig. 4.7 but considering different infinitesimal variances for the obstacle and goal diffusions. (a) shows results for $\mathscr{G}^g = \mathscr{G}^o = 0.8I_4$, (b) for $\mathscr{G}^g = \mathscr{G}^o = 0.5I_4$, (c) for $\mathscr{G}^g = 0.2I_4$ and $\mathscr{G}^o = 0.1I_4$ and (d) for $\mathscr{G}^g = \mathscr{G}^o = 0.01I_4$.

Optimal trajectories and feedback gains are shown in Fig. 4.9 for the expected cost policy in comparison with the mean variance solution. In accordance with the behavior observed in the point mass example, the risk-averse evaluation of obstacle variability drives the optimal trajectory away from the goal at the beginning of the simulation. This effect is also boosted by the risk-seeking evaluation of goal variability, which also enables a more pronounced avoidance of the static obstacle.

Car-like Robot

Consider a simplified car-like robot model with state space $\boldsymbol{x}^r = \begin{bmatrix} x_1 & x_2 & \theta & v & c \end{bmatrix}^{\mathrm{T}}$ and dynamics

$$\dot{x}_1 = v\cos\theta \quad \dot{x}_2 = v\sin\theta \quad \dot{\theta} = vc \quad \dot{v} = u_1 \quad \dot{c} = u_2 \tag{4.46}$$

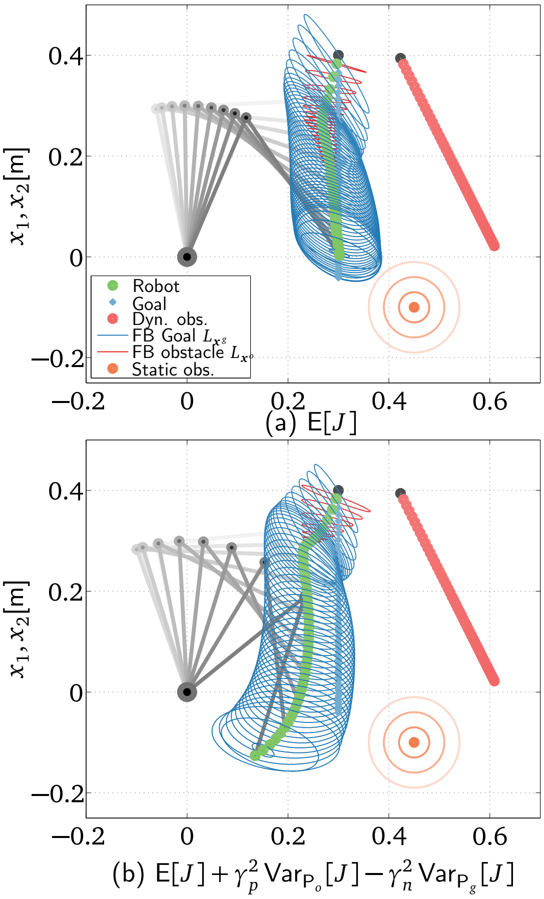
where x_1 and x_2 denote the 2D position, θ the orientation, v the velocity, c the curvature and the control input is given by $\boldsymbol{u} = \begin{bmatrix} u_1 \, u_2 \end{bmatrix}^{\mathrm{T}}$.

The control task consists of tracking goal $\boldsymbol{x}^g$ with passive uncertain dynamics (4.46) while avoiding an obstacle $\boldsymbol{x}^o$ with uncertain passive mass-damper dynamics with five- and four-dimensional Brownian motions respectively. Their infinitesimal variances are given by $\mathscr{G}^g = diag\{\ 1 \ \ 1 \ \ 1 \ \ 0 \ \ 0\ \}$ and $\mathscr{G}^o = I_4$. The state of the system is given by

$$\xi^* = \left[(\boldsymbol{x}^g - \boldsymbol{x}^r)^{\mathrm{T}} \ (\boldsymbol{x}^o - [x_1 \ x_2]^{\mathrm{T}})^{\mathrm{T}} \ (\dot{\boldsymbol{x}}^o - [\dot{x}_1 \ \dot{x}_2]^{\mathrm{T}} \right]^{\mathrm{T}} .$$

The cost function considered is identical to the previous setting, except for the goal's quadratic weighting matrix and the control input's weighting matrix, which are set to $Q_g = diag\{\ 1 \ \ 1 \ \ 0 \ \ 0 \ \ 0\ \}$ and $R = 10^{-5}$ respectively.

The resulting optimal trajectories are depicted in Fig. 4.10 for the risk neutral and the mean variance solution with aversion towards obstacle-related variability and risk-seekingness towards goal-related cost variance. The goal tracking flexibility provided by the risk-preferring evaluation of goal variability enables deviations from the desired trajectory, providing improved obstacle avoidance.



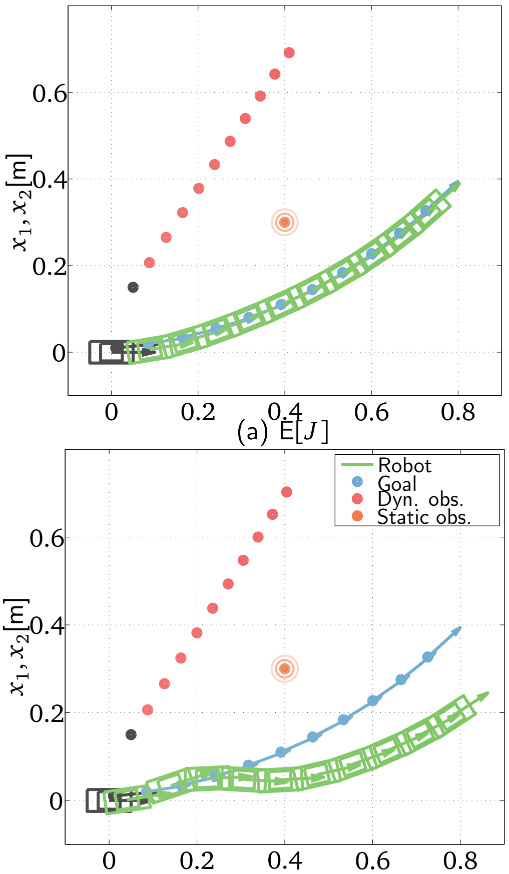
(a) $\mathsf{E}[J]$

(b) $\mathsf{E}[J] + \gamma_p^2 \mathsf{Var}_{\mathsf{P}_o}[J] - \gamma_n^2 \mathsf{Var}_{\mathsf{P}_g}[J]$

Figure 4.9: Optimal trajectories and feedback gains for a two-link manipulator $\boldsymbol{x}^r$ tracking goal $\boldsymbol{x}^g$ with passive mass-damper system dynamics and avoiding obstacle $\boldsymbol{x}^o$ with noisy passive mass-damper system dynamics including static obstacles and where $\gamma_n = 6$ and $\gamma_p = 3$. The static obstacle considered is normally distributed, centered at $(-0.1, 0.45)$ with covariance matrix $0.3I_2$. Initial states of robot, goal and obstacle are $\boldsymbol{x}_0^r = [0.3\ 0.4]$, $\dot{\boldsymbol{x}}_0^r = [0\ 0]$, $\boldsymbol{x}_0^g = [0.3\ 0.4]$, $\dot{\boldsymbol{x}}_0^g = [0\ -0.7]$, $\boldsymbol{x}_0^o = [0.42\ 0.4]$ and $\dot{\boldsymbol{x}}_0^o = [0.3\ -0.6]$. Results are shown every 0.02s.

4.7 Summary

In this chapter we studied the design of uncertainty-dependent optimal control schemes. The emergence of probabilistic methods in robotic settings motivates the design of such decision-makers, which consider uncertainty as a decisive parameter that influence decisions. We presented a systematic approach to this problem by analyzing stochastic optimal control problems for dynamics with multiple additive uncertainty sources. In constrast to classical stochastic optimal control solutions, the optimization criterion presented in Section 4.1 considers not only the expected cost but also an arbitrary weighted sum of cost cumulants. Cumulants represent high-order cost statistics and properly capture the influence of uncertainty on the cost. The weighting factors determine the assessment of cumulants on the cost, acting either as a discounting measure or an additive penalty. In addition, the cost variability produced by each uncertainty source is treated marginally. This fact enables the design of



(a) $\mathsf{E}[J]$

(b) $\mathsf{E}[J]+\gamma_p^2\,\mathsf{Var}_{\mathsf{P}_o}[J]-\gamma_n^2\,\mathsf{Var}_{\mathsf{P}_g}[J]$

Figure 4.10: Optimal trajectories for a car-like robot $\boldsymbol{x}^r$ tracking a goal $\boldsymbol{x}^g$ with similar but stochastic and passive dynamics while avoiding an obstacle $\boldsymbol{x}^o$with noisy passive mass-damper dynamics for a horizon of $T_c = 2\text{s}$, where $\gamma_n = 0.35$ and $\gamma_p = 0.35$. The static obstacle considered is normally distributed, centered at $(0.4, 0.3)$ with covariance matrix $0.1 I_2$. Initial states of robot, goal, and obstacle are $\boldsymbol{x}^r = [\,0 \;\; 0 \;\; 0 \;\; 0.01 \;\; 0.01\,]$, $\boldsymbol{x}^g = [\,0.01 \;\; 0.01 \;\; 0.1 \;\; 0.4 \;\; 0.8]$,$\boldsymbol{x}_0^o = [0.05\;0.15]$ and $\dot{\boldsymbol{x}}_0^o = [0.2\;0.3]$. Results are shown every 0.2s.

decision-makers that react differently depending on which uncertainty source is responsible for cost variability. The risk-sensitive solution, which considers a specific infinite sum of cost cumulants, and the arbitrary cost-cumulant solution are developed in Section 4.2 and Section 4.3 respectively for linear quadratic systems. The application of the proposed approach in non-linear and non-quadratic settings is then studied in Section 4.4. Locally optimal feedback solutions are found by iteratively performing linear quadratic approximations around a nominal trajectory, solving the local problem and updating the trajectory until convergence. To facilitate the application of the proposed approach in more general problems, Section 4.5 introduced solutions for settings where uncertainty is not present in the equations of motion themselves but in static parameters of the cost such as a desired goal and an obstacle's position. The experimental evaluations from Section 4.6 present simulations in both linear and non-linear and non-quadratic problems, which validate our method and depict its peculiarities. The proposed approach is proven a valid and systematic option for the increasingly frequent problem of robot control under uncertainty.

5

Model-based Anticipatory Control for Human-Robot Cooperative Manipulation

Among all goal-oriented control schemes for physical assistance, anticipation based on human behavior prediction is a specially compelling option. It provides superior performance in terms of human behavior minimization by exploiting human behavior models. However, its implementation entails additional challenging issues concerning robot control. Due to the continuous physical coupling between agents, when robot's expectations mismatch human intentions undesired interaction forces appear incurring safety risks and discomfort. This fact illustrates the impasse that robot decisions must cope with: on one side, predictions enables effective anticipation but on the other side, they potentially produce unpleasant or even dangerous disagreements when prediction errors occur. A deeper understanding of causes and consequences of disagreements and an a priori estimation of potential errors are instrumental steps towards designing intuitive and helpful physical assistants.

This chapter explores anticipatory model-based control schemes with special emphasis on learned probabilistic/statistical behavior models. To identify advantages and limitations of anticipatory robotic helpers, in Section 5.1 we first evaluate the feasibility of proactive assistants acquiring behavior models during interaction with simple tracking control schemes. Results reveal the importance of successful human behavior prediction, which effectively reduces human effort. In contrast, wrong predictions significantly hinder the interaction producing undesired interaction forces. These insights motivate the study of three relevant open issues:

- The identification of internal wrench components which express disagreement between agents and the effective load shares that each agent accounts for. Their characterization enables the acquisition of richer interaction models and intuitive control schemes that react to disagreements.

- The explicit consideration of prediction uncertainty in robot control. Prediction uncertainty serves as an a priori estimation of potential prediction errors and should influence robot actions.
- The selection of the most appropriate model when several behavior models are available. Realistic applications comprise multiple tasks and therefore multiple models. A strategy to select the most promising one is necessary.

The rest of this chapter addresses these three open challenges. Section 5.2 presents a physically-consistent characterization of internal and effective wrench components and their relation with disagreement between agents and load distributions. Supported by this analysis, Section 5.3 presents a novel uncertainty-dependent anticipatory control scheme that adapts to disagreements and prediction variance for navigation tasks. Section 5.4 extends this approach to tasks including environmental contacts. Finally, when multiple behavior models are available, the benefits of an efficient dynamic strategy selection scheme are presented in Section 5.5.

5.1 Emerging Assistance: a Feasibility Study

The presence of a human physically interacting with a robot imposes remarkable challenges for robot control design. At the same time, it is also an opportunity to exploit the human partner's cognitive capabilities and acquire valuable experience. Here, we study the concept of emerging assistance where a robotic partner, initially purely reactive, increases its proactive task contribution as it gathers more experience, as depicted in Fig. 5.1. When first executing a task jointly with the human, the robot only reacts to the haptic input received from its partner through the object. At this point, being unaware of the task goal, the robot is limited to a passive follower and the human consequently leads the task. As the execution advances, the robot incrementally builds a task model based on observed data coming from its sensors. This acquired task knowledge allows the robot to anticipate human action increasing its proactive contribution. The goal of the generated behavior is to reduce human workload during task execution. This concept is similar to the PbD scenario but where both demonstration and reproduction phases take place at the same time.

In the following, the robot behavior is realized implementing an admittance control law

$$M_r\ddot{\boldsymbol{x}} + D_r\dot{\boldsymbol{x}} = \boldsymbol{u}_h + \boldsymbol{u}_r \tag{5.1}$$

with a rendered virtual mass M_r and rendered virtual viscous friction D_r. The human-desired trajectory is estimated based on previous experience by means of the incremental learning algorithm described in Section 3.1, which models the task in terms of observed configuration and twist, i.e. $\xi = \left[\boldsymbol{x}^{\mathrm{T}}\ \dot{\boldsymbol{x}}^{\mathrm{T}}\right]^{\mathrm{T}}$. The assistive robot behavior is then rendered by a proportional-derivative controller given normally distributed predictions $\hat{\xi}_d = \left[\hat{\boldsymbol{x}}_d^{\mathrm{T}}\ \hat{\dot{\boldsymbol{x}}}_d^{\mathrm{T}}\right]^{\mathrm{T}}$ as

$$\boldsymbol{u}_r = K_{\dot{x}}(\mathsf{E}[\hat{\dot{\boldsymbol{x}}}_d] - \dot{\boldsymbol{x}}) + K_x(\mathsf{E}[\hat{\boldsymbol{x}}_d] - \boldsymbol{x}) \tag{5.2}$$

where $K_{\dot{x}}$ and K_x denote the proportional and derivative control gains respectively.

With this simple control scheme, in this section we evaluate the feasibility and properties of emerging assistance in two scenarios. In Section 5.1.1 we present a user study in a two-dimensional virtual scenario. A proof-of-concept implementation with a human-sized robot is exposed in Section 5.1.2. Results indicate reduced human effort but also reveal interesting open challenges described in Section 5.1.3.

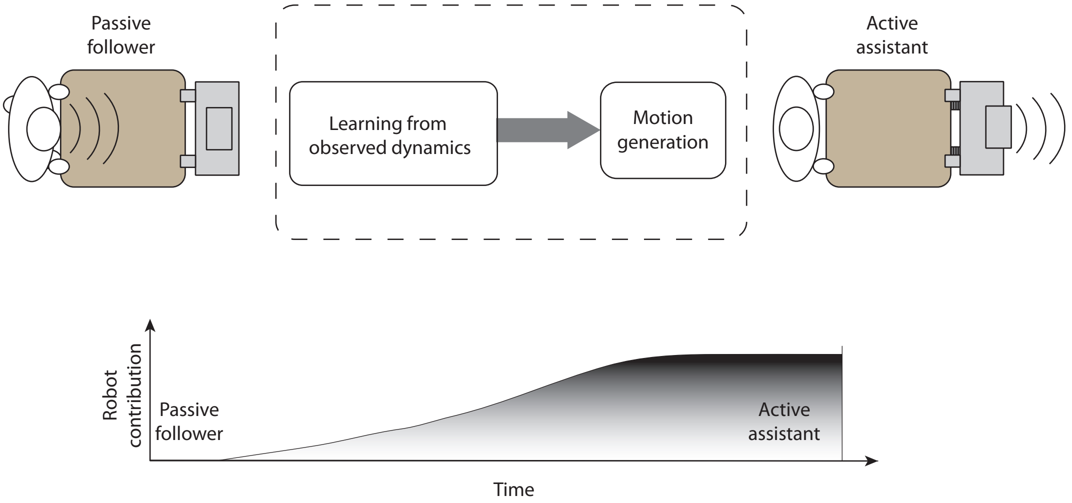


Figure 5.1: Proposed approach for emerging assistance: from a passive follower to a proactive task contributor ny means of learning from demonstrations and interaction control.

5.1.1 Experiments in a 2 DoF Virtual Scenario

To confirm the validity of the proposed approach we chose a simple virtual scenario for a task consisting of tracking a predefined path using the two-dimensional virtual scenario. A user study confirms that, after several repetitions of the task, the robot succesfully builds a model of the human behavior and its assistance emerges reducing the applied human force.

Experimental Setup

The two dimensional haptic interface used in this experiment is described in appendix A.1. The learning and prediction framework is implemented in an online fashion in C++ and executed on a personal computer with an update rate of 20 Hz. Each segment and each node is encoded in an HMM with $N = 10$ states and $K = 1$ Gaussians per state as explained in Section 3.1. In order to form a node in the primitive tree, a group of at least $M = 3$ members, see Section 3.1.2, is needed and only the upper nodes of each branch are considered for the prediction. For the primitive recognition, the incoming observations consider a window of 30 samples and a window HMM with 20 states is used. The predicted data is generated for a 0.3 s time horizon in the future. When only part of the task has been learned, predictions are, at some points, far away than the expected future trajectory. To avoid disturbing the human in such cases, we ignore the predicted points that are further away than 0.02 m from the current position.

The parameters from equations (5.1) and (5.2) are set to the following values

$$\begin{aligned} M_r &= \mathrm{diag}(40\,\mathrm{kg} \quad 40\,\mathrm{kg}) \\ D_r &= \mathrm{diag}(90\,\mathrm{Ns/m} \quad 90\,\mathrm{Ns/m}) \\ K_x &= \mathrm{diag}(200\,\mathrm{N/m} \quad 200\,\mathrm{N/m}) \\ K_{\dot{x}} &= \mathrm{diag}(90\,\mathrm{Ns/m} \quad 90\,\mathrm{Ns/m}) \end{aligned}$$

The high mass and damping values of the robot admittance emulate a heavy virtual object making tracking the pre-defined trajectory difficult for the human. As the assistance of the robot provides a force in the human desired direction, once the assistance is effectively working, a lower human force is expected or, equally valid, the task execution time decreases due to the robot's help.

We performed an experiment with 6 subjects[1] (all male, average age 28) for the task presented in Section 3.1 consisting on following a path describing a flower, as shown in Fig. 3.3. The task was repeated cyclically 20 times, and the robot was learning and assisting during the whole experiment. To decouple the influence of the robot assistive input from any possible learning effect of the human user interacting with the haptic device, after 15 repetitions, the robot assistive input was nullified, i.e. $\boldsymbol{u}_r = 0$, and activated again 3 repetitions later. The subjects had the opportunity to familiarize with the haptic device for a few minutes before the experiment and they were told to follow the pre-defined trajectory for the given repetitions.

Results

As shown in Fig. 5.2(b) the human force decreases together with the prediction error, depicted in Fig. 5.2(a), but starts rising on the ninth repetition, caused by a progressively faster speed execution, as shown in Fig. 5.2(c). As the assistance is turned off between repetitions 16 to 18, the human force clearly rises and the executions even become slower. Turning the assistance on again produced the expected previous results, which discards any improved efficiency due to a learning effect from the human side.

In order to decouple results from task execution speed, we performed the same experiments fixing a constant task completion time of 32 s by adding an exemplary execution to be followed. Subjects were told to follow a black dot showing the exemplary run. As shown in Fig. 5.3 the assistance's effect becomes more significant and the consequences of nullifying it more evident: human force clearly decreases together with prediction errors and when the robot stops assisting the initial values of human force are recovered. It is also remarkable, that, due to the anticipatory nature of the proposed approach and as the learning algorithm does not discard previously learned information, even when the robot was left on its own (the human releases the handle), it still accomplishes the task. This serves as an example of a continuous kinesthetic teaching where the robot learns as it reproduces the task while being corrected by the human teacher.

[1] Although results from 6 different subjects do not allow for statistical significance, trends can be observed.

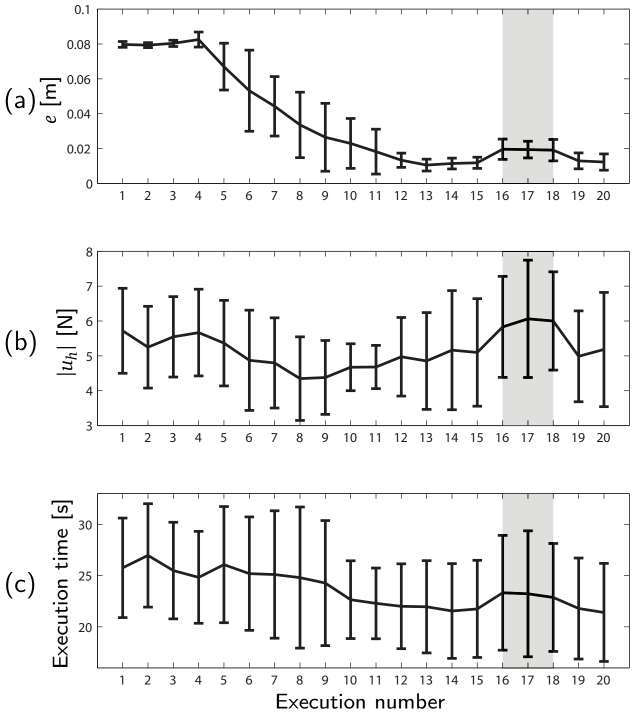


Figure 5.2: Assistance results. All three plots show the evolution per repetition of the task. The mean values of the prediction error, the human force and the time execution per repetition of the task are presented in a), b) and c) respectively. The gray region marks the 3 executions where the assistive input is nullified, i.e. $\boldsymbol{u}_r = 0$ in (5.1).

5.1.2 Experiments in a Full-Scale Scenario

To evaluate the considered approach in more realistic settings we conducted a full-scale experiment in our laboratory. As an exemplary domain, we chose a classic-car restoration setting.

Experimental Setup

The robot used in this experiment is described in appendix A.2. The manipulated object, a 1.2 m long *Mini*'s steel bumper (b) weighing 1.9 kg is depicted in Fig. A.2. The distance between the pre-defined grasp points of human and robot is 1.1 m. The task consists of the joint transportation of the bumper together with a human partner from an initial position, at the top left of the top-view of our lab, see Fig. 5.6, to the mounting position at the front of the car. A wall and a trolley represent the environmental obstacles present in the scene. In this proof-of-concept implementation, a human user performed nine trials following three different paths.

The underlying control concept of the manipulator-base coordination adopted in this experiment is depicted in Fig. 5.4. Similar to [162], the admittance control law is calculated in inertial coordinates so that repositioning of the mobile base does not affect the end-effector position. The actual end-effector pose ${}^{R}\boldsymbol{x}_m$ is used to derive a velocity command ${}^{R}\dot{\boldsymbol{x}}_b$ to the

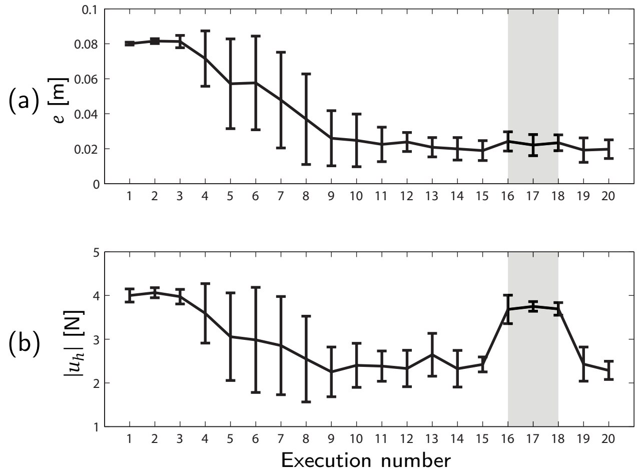


Figure 5.3: Assistance results with constant execution time. The mean values of the prediction error and the human force per repetition of the task are shown in a) and b) respectively. The gray region marks the 3 executions where the assistive input is nullified, i.e. $\boldsymbol{u}_r = 0$ in (5.1).

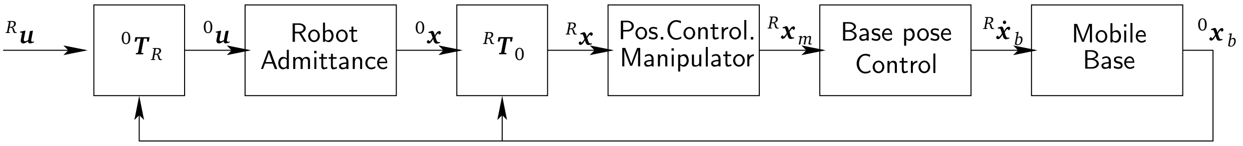


Figure 5.4: Inertial admittance-type control scheme

mobile base, following the base control law:

$$^R\dot{\boldsymbol{x}}_b = \begin{pmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{pmatrix} = \mathrm{diag}\begin{pmatrix} K_{hdg} & K_{dst} & K_{tng} \end{pmatrix} \begin{pmatrix} e_{hdg} \\ e_{dst} \\ e_{tng} \end{pmatrix}$$

Three independent proportional control laws move the mobile base minimizing heading error e_{hdg}, distance error e_{dst} and tangential error e_{tng} of the base pose w.r.t. the end-effector pose $^R\boldsymbol{x}_m$, as illustrated in Fig. 5.5.

A reference pose of the end-effector $^R\boldsymbol{x}_d$ is chosen to meet certain requirements regarding task-related manipulability. The resulting motion command $^R\dot{\boldsymbol{x}}_b$ is then executed by an omnidirectional velocity control law as proposed in [163].

For simplicity, the implementation of the impedance control law is reduced to the x/y-plane. The parameters from equations (5.1) and (5.2) to

$$\begin{aligned} M_r &= \mathrm{diag}\begin{pmatrix} 15\,\mathrm{kg} & 15\,\mathrm{kg} & 0.3\,\mathrm{kgm}^2 \end{pmatrix} \\ D_r &= \mathrm{diag}(80\,\mathrm{Ns/m} \quad 80\,\mathrm{Ns/m} \quad 7\,\mathrm{Nms/rad}) \\ K_x &= \mathrm{diag}(0 \quad 0 \quad 0) \\ K_{\dot{x}} &= \mathrm{diag}(30\,\mathrm{Ns/m} \quad 30\,\mathrm{Ns/m} \quad 0.1\,\mathrm{Nms/rad}) \end{aligned}$$

Note, that the zero-value for K_x leads to a drift-free behavior of the assistance controller to compensate for drift induced by the robot's odometry. The 12-dimensional observation

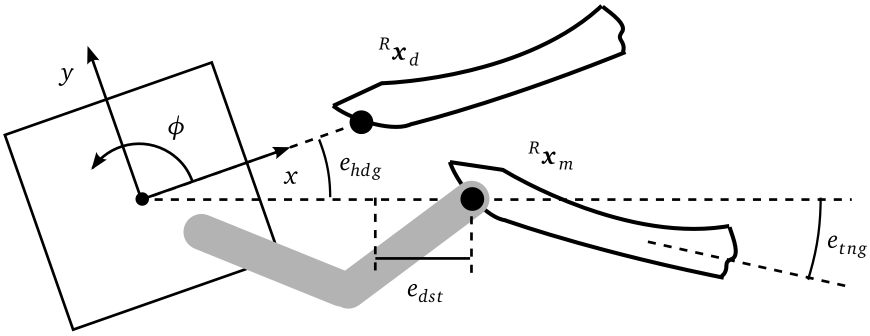


Figure 5.5: Base pose control w.r.t. to a reference pose $^R\boldsymbol{x}_d$ of the manipulator's end-effector.

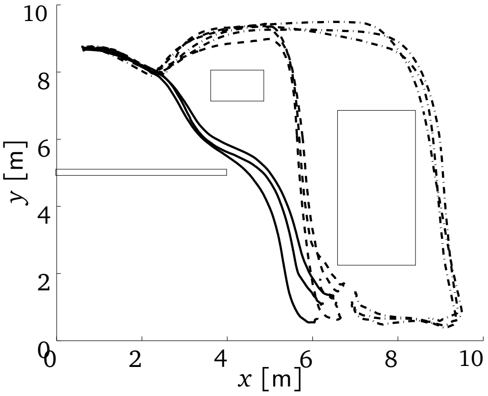


Figure 5.6: 2-dimensional position component of 12-dimensional training data from three trials per each of three semantically different paths

space ξ of the time-based HMM training method is composed of the following dimensions in inertial coordinates:

- the 3-dimensional position of the end-effector $^0\boldsymbol{p}$
- the 3-dimensional angular velocity $^0\boldsymbol{\omega}$
- the 6-dimensional wrench $^0\boldsymbol{u}$ in inertial coordinates

As unified Gaussian computations on 6-D poses remain a computationally extensive problem [164], we use angular velocities as unambiguous training input.

The online learning algorithm implementation is the same as in the previous experiment but in this case is parameterized as follows. Each segment and each node is encoded in an HMM with $N = 15$ states and 1 Gaussian per state. In order to form a node in the primitive tree, a group of at least $M = 2$ members is needed. We use a window of 90 samples over the incoming observations for the primitive recognition and a window HMM with 30 states with a prediction horizon of 0.3 s.

Results

Fig. 5.6 depicts the 2-dimensional position component of the nine trials performed with our system in this scenario. Note that the odometry drift leads to diverging paths. In Fig. 5.7 the

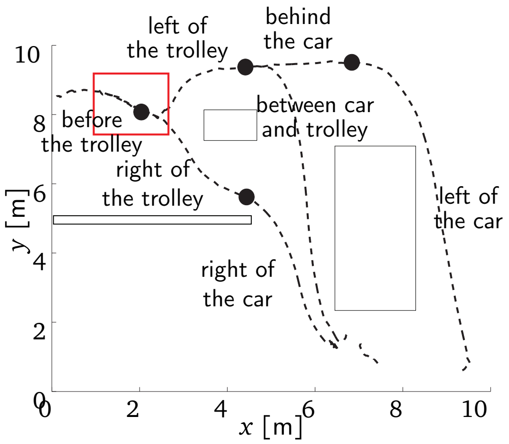


Figure 5.7: 2-dimensional position component of 12-dimensional generalized output for the graph nodes. The red box represents the furcation area examined in Fig. 5.8.

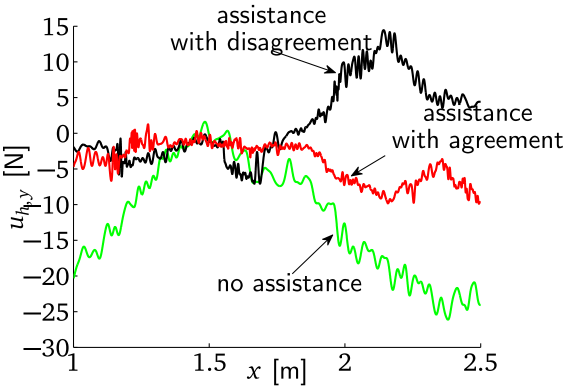


Figure 5.8: Wrench component $u_{h,y}$ plotted for three different conditions of assistance. The window for position x is centered around the node transition 'before the trolley' - 'left/right of the trolley'.

2-dimensional position component of the 12-dimensional generalized output of the learning algorithm after observing all trials from Fig. 5.6 is shown. It is remarkable that training data did not yield the necessary characteristics for successful segmentation in the region close to the car, as the human partner had difficulties to avoid collisions between the robot and the car in narrow space.

The effect of successful prediction is visible in Fig. 5.8. Due to its relevance, we present data of the first bifurcation as marked in Fig. 5.7. The green line shows the baseline implementation: a passively following robot. Significant force is required to pull the robot into the y-direction, perpendicular to the primary direction of motion. The red line shows how completely successful prediction due to user feedback leads to significantly reduced forces. The case of false prediction was provoked to generate the force trajectory shown in black. However, after the positive slope along a distance of approx. 0.5 m, the prediction is corrected and the force returns to a comparable absolute value as in the correctly predicted case.

As an indicator of effective assistance, we consider the required human wrench input as a suitable measure for the performance evaluation of the assistance. Fig. 5.9 shows the average human applied wrench components per bifurcation in the passive case (green) in comparison

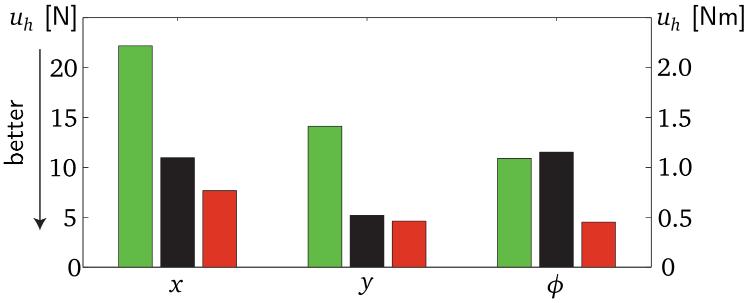


Figure 5.9: Average human force and torque input per bifurcation situation for the *no assistance*, *false* and *successful prediction* from left to right.

with anticipatory assistance for both false (black) and successful (red) prediction. In case of a successful motion prediction (red), results show that human force can be significantly reduced w.r.t. a passive follower (green). In contrast, it is remarkable that in the case of wrong predictions (black), the human wrench contribution considering torque components exceeds the passive follower. This fact reveals that human behavior anticipation is a double-edged sword: in the case of successful prediction it diminishes the necessary human workload but mispredictions produce higher interaction forces than pure passive followers.

5.1.3 Challenges of Model-based Anticipatory Physical Assistants

The experiments presented in the previous two subsections reveal that anticipation based on learned models is an effective strategy for reducing human wrench contribution. Results from Section 5.1.1 show a significant decrement of the required human force as the robot accumulates task experience. In contrast, in case of misprediction, anticipatory motion generation yield clear disadvantages even in comparison with passive control schemes, as results from Section 5.1.2 illustrate. In such cases, the human partner must exert additional counteracting wrench components that compensate for the robot's wrong assistive input. These results motivate the study of the following open issues for anticipatory physical assistants:

- The identification of **counteracting wrench components** that arise during interaction. These components indicate the existence of discrepancies with the human partner in terms of some the control problem assumptions. Observing such components enables the acquisition of behavior models that predict not just a potential desired trajectory ξ_d but also the predicted interaction wrenches that following ξ_d imply. This issue is studied in Section 5.2, where effective and internal wrenches are identified together with the effective loads that each agent accounts for.

- The consideration of the expected **prediction precision** in the anticipatory control scheme. In the case of probabilistic or statistical behavior models, prediction variance is a suitable indicator of accuracy. This provides an *a priori* estimation of potential disagreements with the human. Control schemes studying this problem are presented in Section 5.3 and Section 5.4 for navigation tasks and task requiring contacts with the environment respectively.

- The **selection** of the most appropriate model when several models are available. This issue is addressed in Section 5.5, where we present a dynamic strategy selection scheme depending on a short-term retrospective evaluation of prediction performance. In addition, a simple model relying on human sensed wrenches is also introduced, becoming a suitable alternative when no available behavior model matches human intentions.

5.2 Wrench Decomposition and Effective Load

The identification of internal wrench components is a key step in the design of efficient goal-oriented physical assistants. It enables control design regulating counteracting components as well as the acquisition of richer behavior models. Internal wrench components depend on the effective load that each agent accounts for. In this section we study both quantities and present relevant measures for evaluating physical interaction between proactive agents.

5.2.1 Problem Formulation

Consider the N-agents system from Section 2.1. For simplicity of illustration, no environmental contacts are considered and the object-centered dynamics of the system are given by

$$M_o\ddot{\boldsymbol{x}}(t)+\boldsymbol{h_o}(\boldsymbol{x}(t),\dot{\boldsymbol{x}}(t))=\boldsymbol{u}(t)=G\tilde{\boldsymbol{u}}(t)=GA\boldsymbol{u}_d(t)\,. \tag{5.3}$$

This section explores two complementary problems:

- Given a desired resulting wrench $\boldsymbol{u}_d$, what are the possible physically consistent decomposition matrices A that produce no counteracting wrenches? (Section 5.2.2)
- Given the observed agents' wrenches $\tilde{\boldsymbol{u}}$, what is the corresponding load share that each agent accounts for and which are the effective and counteracting components? (Section 5.2.3)

Physically consistent definitions of disagreement and load share are further proposed in Section 5.2.4. With this analysis, behavior models for cooperative manipulation tasks can be characterized not just by the observed trajectory $\boldsymbol{x}(t)$ but also the observed load distribution and internal wrench components. These models are better suited for later control design than models considering $\boldsymbol{x}(t)$ and applied wrenches $\tilde{\boldsymbol{u}}$, as in [41]. Although applied wrenches implicitly represent both internal and effective components, they are not isolated. This prevents from any control design targeting internal or effective components separately.

5.2.2 Non-Uniform Wrench Decomposition Matrices

Introduced in Section 2.1, the design of the matrix A in (5.3) determines the agent wrenches $\tilde{\boldsymbol{u}}$ that yield a desired $\boldsymbol{u}_d$. A desirable characteristic is the absence of counteracting components in the resulting $\tilde{\boldsymbol{u}}$, which produce unnecessary work but have no influence on dynamics (5.3). A recurrent approach in the literature is the Moore-Penrose pseudoinverse of G, denoted G^+, which yields the solution minimizing $||\boldsymbol{u}_d - A\tilde{\boldsymbol{u}}||$ with minimum norm $\tilde{\boldsymbol{u}}$ and thereby assuming a uniform allocation policy. This approach lacks physical consistency as the norm of a wrench, consisting of both forces [N] and torques [Nm], is physically undefined. A valid option are generalized inverses [165], which define some predefined weighting in order to define the otherwise physically inconsistent norm [166], similarly yielding uniform allocations. However, humans clearly exhibit non-uniform distributions and even favor asymmetric load distributions interacting in dyads [46]. In this subsection we aim for a physically consistent decomposition that supports non-uniform allocations policies.

To ensure physical consistency, we first abstract the analysis from object geometry and grasping constraints by setting it in the object frame as in [167]. In this basis set force and torque components are decoupled and treated independently. Let $\overline{\boldsymbol{u}}_i$ be the applied wrench of agent i on the object frame, such that $\overline{\boldsymbol{u}}_i = G_i \widetilde{\boldsymbol{u}}_i$. Its force and torque components and the resulting wrench are denoted

$$\overline{\boldsymbol{u}}_i = \begin{bmatrix} \overline{\boldsymbol{f}}_i \\ \overline{\boldsymbol{\tau}}_i \end{bmatrix} \qquad \boldsymbol{u}_d = \begin{bmatrix} \boldsymbol{f}_d \\ \boldsymbol{\tau}_d \end{bmatrix}$$

In the object frame the decomposition problem is the design of A such that

$$\overline{\boldsymbol{u}} = A\boldsymbol{u}_d \qquad \boldsymbol{u}_d = \overline{G}A\boldsymbol{u}_d, \tag{5.4}$$

where we defined the stacked forms

$$\overline{G} = \begin{bmatrix} I_1 & I_2 & \cdots & I_N \end{bmatrix} \qquad \overline{\boldsymbol{u}} = \begin{bmatrix} \overline{\boldsymbol{u}}_1 & \overline{\boldsymbol{u}}_2 & \cdots & \overline{\boldsymbol{u}}_N \end{bmatrix},$$

with similar definitions for force and torque components.

The decomposition matrices A representing allocation policies inducing no internal stress are defined in the following proposition.

Proposition 5.1. *The family of solutions satisfying* (5.4) *yielding no counteracting wrenches is given by*

$$A = \begin{bmatrix} \overline{A}_1 \, \overline{A}_2 \, \dots \, \overline{A}_N \end{bmatrix}^{\mathrm{T}},$$

where

$$\overline{A}_i = \operatorname{diag}\begin{bmatrix} \alpha_{f,i} & \alpha_{f,i} & \alpha_{f,i} & \alpha_{\tau,i} & \alpha_{\tau,i} & \alpha_{\tau,i} \end{bmatrix}, \tag{5.5}$$

and

$$\begin{aligned} &\alpha_{f,i}, \alpha_{\tau,i} \geq 0 & &\forall i = 1 \cdots N \\ &\sum_{i=1}^{L} \alpha_{f,i} = 1 & &\sum_{i=1}^{L} \alpha_{\tau,i} = 1\,. \end{aligned} \tag{5.6}$$

Proof. Consider first only the force components. Gauss's principle of least constraint or least forcing states that the true motion of a mechanical system of N masses minimizes

$$J = \sum_{i=1}^{N} m_i || \ddot{\boldsymbol{p}} - \frac{\overline{\boldsymbol{f}}_i}{m_i} ||, \tag{5.7}$$

where m_i is the i-th effective mass and $\boldsymbol{p}$ is the object's position. The multi-agent formulation of (5.3) represents a multibody system where each of the bodies are constrained to the same position $\boldsymbol{p}$ and share the overall mass of the object m_0, i.e.

$$\sum_{i=1}^{N} m_i = m_0 \quad m_i \geq 0 \ . \tag{5.8}$$

From (5.4) and (5.5), the assigned force component of the i-th agent is given by

$$\overline{\boldsymbol{f}}_i = \alpha_{f,i} \boldsymbol{f}_d \tag{5.9}$$

Applying (5.5) produces a resulting wrench $\sum_{i=1}^{N} \alpha_{f,i} \boldsymbol{f_d} = \boldsymbol{f_d}$ and an acceleration $\ddot{\boldsymbol{p}} = \dfrac{\boldsymbol{f}_d}{m_0}$.

The effective mass that agent i is accounts for must fulfill Gauss's principle and therefore minimizes J. The stationary point fulfills

$$\frac{\partial J}{\partial m_i} = \frac{\partial \left(m_i \left\| \dfrac{\boldsymbol{f}_d}{m_0} - \dfrac{\alpha_{f,i} \boldsymbol{f}_d}{m_i} \right\| \right)}{\partial m_i} = 0 \ .$$

Solving for m_i yields

$$m_i = \alpha_{f,i} m_0 \ ,$$

and therefore $\alpha_{f,i} = \dfrac{m_i}{m_0}$ represents the *normalized effective mass* for agent i. Conditions (5.6) ensure the fulfillment of constraints (5.8).

From (5.8), the resulting acceleration considering the sum of effective masses must be consistent with the overall acceleration, i.e.

$$\frac{\boldsymbol{f}_d}{\sum_{i=1}^{N} m_i} = \frac{\boldsymbol{f}_d}{m_0} \ .$$

As a result the components accounting for the effective masses fulfill

$$\frac{\sum_{i=1}^{N} m_i}{m_0} \boldsymbol{f}_d = \sum_{i=1}^{N} \alpha_{f,i} \boldsymbol{f}_d = \boldsymbol{f}_d \ ,$$

yielding definition (5.9) and ensuring that $\overline{\boldsymbol{f}}_i$ comprise only effective wrenches.

The proof for torque components yielding the normalized effective moment of inertia $\alpha_{\tau,i}$ are derived in an identical manner. □

The duple $\{\alpha_{f,i}, \alpha_{\tau,i}\}$ represents the load share in terms of the fractions of the overall mass and inertia of the system (5.3) that agent i accounts for. Consider now a dyadic setting where a human interacts with a robot with load shares $\{\alpha_{f,h}, \alpha_{\tau,h}\}$ and $\{\alpha_{f,r}, \alpha_{\tau,r}\}$ respectively. It is interesting to comment extreme cases of load distributions

- $\{\alpha_{f,h} = 1, \alpha_{\tau,h} = 1\}$ and $\{\alpha_{f,r} = 0, \alpha_{\tau,r} = 0\}$, which corresponds to a passive robotic partner or, similarly, a kinesthetic teaching scenario for a learning assistant.

- $\{\alpha_{f,h} = 0, \alpha_{\tau,h} = 0\}$ and $\{\alpha_{f,r} = 1, \alpha_{\tau,r} = 1\}$, which corresponds to a proactive robotic partner that accounts for all necessary effort or, similarly, the reproduction phase in a PbD setting.

All load distributions lying in between correspond to cooperative settings where the load is effectively shared. Note that the generalized Moore-Penrose inverse solution yields $\{\alpha_{f,h} = 0.5, \alpha_{\tau,h} = 0.5\}$ and $\{\alpha_{f,r} = 0.5, \alpha_{\tau,r} = 0.5\}$, uniformly distributing load among agents.

5.2.3 Effective and Internal Wrenches

Proposition 5.1 defines the decomposition matrices A yielding no counteracting wrenches in the object frame. It enables control design with arbitrary load distributions. Another relevant issue in pHRI settings is the inverse problem, where, given an observed $\overline{\boldsymbol{u}}$, the respective *effective* and *internal* components are identified. The separation of these two additive components is defined as

$$\begin{aligned} \overline{\boldsymbol{u}} &= \overline{\boldsymbol{u}}_{\text{eff}} + \overline{\boldsymbol{u}}_{\text{int}} \\ s.t \quad \overline{G}\,\overline{\boldsymbol{u}}_{\text{eff}} &= \overline{G}\,\overline{\boldsymbol{u}} = \boldsymbol{u}_d \\ \overline{G}\,\overline{\boldsymbol{u}}_{\text{int}} &= \boldsymbol{0} \quad , \end{aligned} \tag{5.10}$$

where:

- the effective wrenches $\overline{\boldsymbol{u}}_{\text{eff}}$ are components that constitute the resulting wrench $\boldsymbol{u}_d$ and can be written in terms of an instance of A from proposition 5.1 as

$$\overline{\boldsymbol{u}}_{\text{eff}} = A\boldsymbol{u}_d \ . \tag{5.11}$$

- the internal wrenches $\tilde{\boldsymbol{u}}_{\text{int}}$ produce unnecessary internal stress in the object and lie in

$$\overline{\boldsymbol{u}}_{\text{int}} \in \text{Null}(\overline{G}) \ , \tag{5.12}$$

  where $\text{Null}(\overline{G})$ denotes the nullspace of $\overline{G}$.

For a given $\overline{\boldsymbol{u}}$, the possible effective wrench components are defined in the following proposition.

Proposition 5.2. *The* effective wrench *components are given by*

$$\overline{\boldsymbol{u}}_{\mathit{eff},i} = \begin{bmatrix} \alpha_{f,i}\boldsymbol{f}_d \\ \alpha_{\tau,i}\boldsymbol{\tau}_d \end{bmatrix} \tag{5.13}$$

where $\alpha_{f,i} = \frac{m_i}{m_0}, \alpha_{\tau,i} = \frac{j_i}{j_0} \in \mathbb{R}$ *satisfy*

$$\begin{aligned} &\min_{m_i} \sum_{i=1}^{N} m_i ||\ddot{\boldsymbol{p}} - \frac{\overline{\boldsymbol{f}}_i}{m_i}|| && \min_{j_i} \sum_{i=1}^{N} j_i ||\dot{\boldsymbol{\omega}} - \frac{\overline{\boldsymbol{\tau}}_i}{j_i}|| \\ &s.t.\ m_i \geq 0 && s.t.\ j_i \geq 0 \\ &\quad \sum_{i=1}^{N} m_i = m_0 && \quad \sum_{i=1}^{N} j_i = j_0 \end{aligned}$$

and $m_0 \in \mathbb{R}$ *and* $j_0 \in \mathbb{R}$ *are the mass and moment of inertia of the object respectively.*

Proposition 5.2 follows from ensuring Gauss's principle of least constraint. In this thesis we only consider dyadic settings. This particular case constraining $N = 2$ admits simple solutions exposed in the following proposition.

Proposition 5.3. *The* effective wrench *components of a 2-agent system are given by*

$$\overline{\boldsymbol{u}}_{\mathit{eff},i} = \begin{bmatrix} \alpha_{f,i}\boldsymbol{f}_d \\ \alpha_{\tau,i}\boldsymbol{\tau}_d \end{bmatrix} \tag{5.14}$$

where $\alpha_{f,i}, \alpha_{\tau,i} \in \mathbb{R}$ *are given by*

$$\begin{aligned} \alpha_{f,i} &= \min\left(\max\left(\frac{\overline{\boldsymbol{f}}_i^{\mathrm{T}}\boldsymbol{f}_d}{||\boldsymbol{f}_d||^2}, 0\right), 1\right) \\ \alpha_{\tau,i} &= \min\left(\max\left(\frac{\overline{\boldsymbol{\tau}}_i^{\mathrm{T}}\boldsymbol{\tau}_d}{||\boldsymbol{\tau}_d||^2}, 0\right), 1\right) . \end{aligned} \tag{5.15}$$

Proof. Consider first only the force components. The stationary point of the unconstrained expression of (5.7) is

$$\frac{\partial J}{\partial m_i} = \frac{\partial\left(m_i \left|\left| \frac{\boldsymbol{f}_d}{m_0} - \frac{\overline{\boldsymbol{f}}_i}{m_i} \right|\right| \right)}{\partial m_i} = 0 .$$

Let $\alpha_{f,i} = \dfrac{m_i}{m_0}$. Solving for m_i yields

$$\alpha_{f,i} = \frac{\overline{\boldsymbol{f}}_i^{\mathrm{T}}\boldsymbol{f}_d}{||\boldsymbol{f}_d||^2} .$$

The minimum and maximum from expression (5.15) ensure constraints (5.8) and hold only for $N = 2$.

The normalized effective moment of inertia components $\alpha_{\tau,i}$ are derived in an identical manner. □

The definition of the internal wrench components follows immediately from (5.10) as

$$\overline{\boldsymbol{u}}_{\mathrm{int}} = \overline{\boldsymbol{u}} - \overline{\boldsymbol{u}}_{\mathrm{eff}} . \tag{5.16}$$

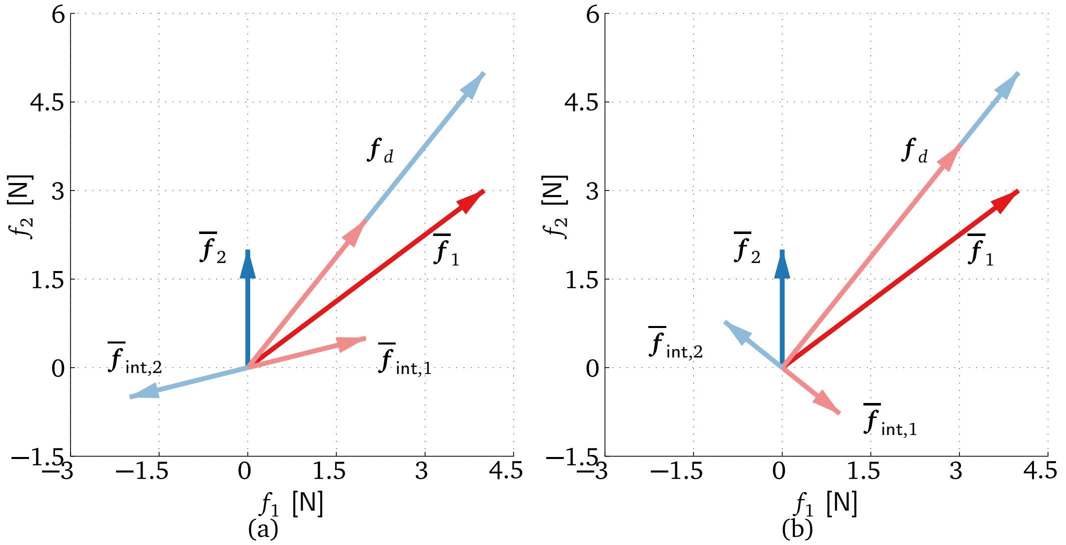


Figure 5.10: Exemplary force decomposition for 2 agents using the Moore-Penrose pseudoinverse (a) and decomposition from proposition 5.3 and (5.16).

The proposed decomposition is depicted in Fig. 5.10 in comparison with the generalized pseudoinverse for a simple example two-dimensional example with two agents. In contrast to the generalized pseudoinverse $\overline{G}^{+}$ from Fig. 5.10(a) which assumes effective load shares $\alpha_{f,1} = \alpha_{f,2} = 0.5$, the decomposition from proposition 5.3 depicted in Fig. 5.10(b) follows from the computation of the effective mass that each agent accounts for, yielding $\alpha_{f,1} = 0.756$ and $\alpha_{f,2} = 0.244$.

The decomposition of observed wrenches in effective and disagreement components enables the definition of interesting performance measures for physical multi-agent interaction tasks.

Definition 5.1. The *effective work* done by agent i of the multi-agent system between time instants t_1 and t_2 is given by

$$\int_{t_1}^{t_2} \bar{\boldsymbol{u}}_{\text{eff},i}^{\text{T}} \dot{\boldsymbol{x}} dt \ .$$

Definition 5.2. The *counteracting work* done by agent i of the multi-agent system between time instants t_1 and t_2 is given by

$$\int_{t_1}^{t_2} \bar{\boldsymbol{u}}_{\text{int},i}^{\text{T}} \dot{\boldsymbol{x}} dt \ .$$

Similar measures for the work of the overall system are straightforwardly computed.

An analysis of possible causes yielding internal wrench components is also relevant in order to understand their role in the interaction. In the following we study the decomposition of wrenches arising from the interaction of decentralized systems, a condition more in accordance with human-robot settings. Consider now the *decentralized* case where each agent follows its own desired resulting wrench and decomposition matrix. From (5.4), the i-th agent assumes

$$\overline{\boldsymbol{u}} = A_i \boldsymbol{u}_{d\,i} \ , \tag{5.17}$$

where $\boldsymbol{u}_{d\,i}$ is the i-th agent desired resulting wrench and A_i an assumed load distribution from Proposition 5.1. If all desired wrenches and load distribution matrices are equal, i.e. $A_i = A_j \ \wedge \ \boldsymbol{u}_{d\,i} = \boldsymbol{u}_{d\,j} \ \forall 1 \leq i, j \leq N$, then the standard problem considered in Proposition 5.1 is recovered and no internal wrench components arise. Consider the case when desired resulting wrenches match but load distributions are different, i.e. $A_i \neq A_j \wedge \boldsymbol{u}_{d\,i} = \boldsymbol{u}_{d\,j} \forall 1 \leq i, j \leq N$. It is straightforward to show that, although only a scaled desired resulting wrench is effectively exerted, no internal wrenches are produced. In contrast, when desired resulting wrenches differ, if the angle between desired resulting forces or torques is not zero, internal wrenches appear as illustrated in the following proposition.

Proposition 5.4. *For a decentralized system where each agents follows* (5.17)*,* $\overline{\boldsymbol{u}}_{int} \neq \boldsymbol{0}$

$$\textit{iff} \quad \exists i, j \mid \frac{\boldsymbol{f}_{d\,i}^{\mathrm{T}} \boldsymbol{f}_{d\,j}}{||\boldsymbol{f}_{d\,i}|| \, ||\boldsymbol{f}_{d\,j}||} < 1 \qquad \textit{or} \qquad \exists i, j \mid \frac{\boldsymbol{\tau}_{d\,i}^{\mathrm{T}} \boldsymbol{\tau}_{d\,j}}{||\boldsymbol{\tau}_{d\,i}|| \, ||\boldsymbol{\tau}_{d\,j}||} < 1 \ . \tag{5.18}$$

Proof. By contradiction. Consider only force components. The contradictory case assumes $\overline{\boldsymbol{f}}_{\mathrm{int}} \neq 0$ and

$$\forall i, j \mid \frac{\boldsymbol{f}_{d\,i}^{\mathrm{T}} \boldsymbol{f}_{d\,j}}{||\boldsymbol{f}_{d\,i}|| \, ||\boldsymbol{f}_{d\,j}||} = 1 \tag{5.19}$$

From (5.17), the applied wrench of the i-th agent is

$$\overline{\boldsymbol{f}}_i = \widetilde{\alpha}_{f,i} \boldsymbol{f}_{d\,i} \ ,$$

where $\widetilde{\alpha}_{f,i}$ is the mass share assumed by agent i in A_i. The overall resulting force is

$$\boldsymbol{f}_d = \sum_{i=1}^{N} \widetilde{\alpha}_{f,i} \boldsymbol{f}_{d\,i} \ .$$

From (5.16), the internal forces are

$$\overline{\boldsymbol{f}}_{\mathrm{int}} = \widetilde{\alpha}_{f,i} \boldsymbol{f}_{d\,i} - \alpha_{f,i} \boldsymbol{f}_d \ . \tag{5.20}$$

From Proposition 5.3, the unconstrained expression for the effective load share $\alpha_{f,i}$ is given by

$$\alpha_{f,i} = \frac{\widetilde{\alpha}_{f,i} \boldsymbol{f}_{d\,i} \boldsymbol{f}_d}{||\boldsymbol{f}_d||^2} = \frac{\widetilde{\alpha}_{f,i} ||\boldsymbol{f}_{d\,i}||}{||\boldsymbol{f}_d||} \ . \tag{5.21}$$

Note that conditions (5.19) ensure constraints (5.8). Substituting (5.21) in (5.20) yields

$$\overline{\boldsymbol{f}}_{\mathrm{int}} = \widetilde{\alpha}_{f,i}\boldsymbol{f}_{d\,i} - \frac{\widetilde{\alpha}_{f,i}||\boldsymbol{f}_{d\,i}||}{||\boldsymbol{f}_{d}||}\boldsymbol{f}_{d} \ . \tag{5.22}$$

From (5.19), all $\boldsymbol{f}_{d\,i}$ are in the same direction and therefore $\boldsymbol{f}_d$ too. $\boldsymbol{f}_d/||\boldsymbol{f}_d||$ is the unit vector of $\boldsymbol{f}_{d\,i}$ and therefore $\overline{\boldsymbol{f}}_{\mathrm{int}} = \boldsymbol{0}$, concluding the proof.

The proof for the inverse condition is also done by contradiction starting from (5.18) and assuming no internal force components and sketched as follows: condition (5.18) implies that in (5.21) a cosinus appears due to the different directions of desired resulting wrenches. Therefore, the difference between applied wrench and effective wrench from (5.22) is non zero yielding the contradiction.

The proof for torque components is derived in an identical manner. □

Proposition 5.4 together with the previous analysis yield interesting conclusions:

- If all desired resulting wrenches $\boldsymbol{u}_{d\,i}$ are equal, a discrepancy in load shares do not produce any internal wrenches; it yields a resulting wrench scaled by a factor.
- If desired resulting wrenches $\boldsymbol{u}_{d\,i}$ are not equal but have same direction, no internal wrenches are exerted.
- The presence of internal wrenches *implies* $\boldsymbol{u}_{d\,i}$'s with different directions.

As a result, internal wrenches play a key role during haptic interaction; in terms of haptic feedback, internal wrench components are the sole communication channel available in order to express the desire to change the direction of the object's acceleration. In other words, internal wrench components can be interpreted as the *haptic negotiation channel* concerning the desired acceleration's direction.

In human-robot settings, assuming that the human partner behaves as in (5.17), i.e. the human does not plan any internal wrenches a priori in its assumed decomposition matrix A, observed internal components imply errors in the estimation of human desired resulting wrench. This fact is very relevant for the design of physical anticipatory control schemes and will be exploited in Section 5.3.

5.2.4 Load Share and Disagreement

There is no consensus in the literature about how to measure disagreement between physically interacting agents. In [46], disagreement between human dyads is interpreted as the difference between applied wrenches. However, this measure only provides no disagreement when agents apply the same wrench, classifying any deviations from this uniform distribution as disagreement. In [52], disagreement is related to internal forces in one-dimension, i.e counteracting force components that do not contribute to the resulting force. Here, we follow this idea and relate disagreement to $\overline{\boldsymbol{u}}_{\mathrm{int}}$. We extend this concept to 6D by means of proposition 5.3 and (5.16), motivated by classical mechanics principles. Note that the observed load share is implicitly identified in the computation of the internal wrenches. The definitions of disagreement, effective load, load share and load distribution follow.

Definition 5.3. The *translational* and *rotational disagreement* of the i-th agent are given by $||\overline{f}_{\text{int},i}||$ and $||\overline{\tau}_{\text{int},i}||$ and measured in [N] and [N/m] respectively.

Definition 5.4. The *effective load* of agent i is given by $\left\{\alpha_{f,i} m_0, \alpha_{\tau,i} j_0\right\}$ and it represents the effective mass and moment of inertia that agent i accounts for.

Definition 5.5. The *load share* of agent i is given by $\left\{\alpha_{f,i}, \alpha_{\tau,i}\right\}$ and it represents the effective share of mass and moment of inertia that agent i accounts for.

Definition 5.6. The *load distribution* of the multi-agent system is given by the set $\left\{\alpha_{f,1}, \alpha_{f,2} \cdots \alpha_{f,N} \quad \alpha_{\tau,1}, \alpha_{\tau,2} \cdots \alpha_{\tau,N}\right\}$ comprising all load shares.

These concepts enable the application of new performance measures and the exploration of new control schemes for pHRI, exploited in the following Sections.

5.3 Uncertainty-dependent Assistance for Navigation Tasks

As presented in Section 2.1.4 proactive assistants rely on feed-forward behavior of motion, force or both of them in order to anticipate human actions. Due to the complexity of human decision-making, acquiring analytical feedforward behavior models is usually not feasible. An effective alternative is given by PbD paradigm. By means of statistical or probabilistic techniques, generalized human task-solving behavior is learned in a data-driven manner. However, learned models (as well as analytical models) are usually far from perfect: prediction errors frequently occur, inducing counteracting forces during haptic interaction thereby significantly diminishing assistive performance. Therefore, an a priori estimation of potential errors or, equivalently, the expected prediction accuracy enriches the prospective capabilities of the robot's assistance and potentially enhances interaction. Such measure is given by the prediction uncertainty in probabilistic models or similarly, by the expected prediction variance or any higher order moment considering statistical models. Although usually neglected in control design for pHRI, model uncertainty plays a key role in a decision-making process in cooperative tasks when an interacting partner is involved [149].

This section presents a novel anticipatory model-based haptic assistance scheme that considers model uncertainty in the robot interaction control. Formulated as an optimal control problem, the robotic assistant adapts its control depending on two sources of uncertainty encoded in a previously learned human behavior model. First, predicted motion uncertainty is included in the optimization criterion by considering the Mahalanobis distance to the expected human-desired trajectory. Second, an estimation of the expected and observed human force variability caused by disagreements with the robot affects the robot dynamics as an additive stochastic input. In order to explicitly take this variability into account, we propose a risk-sensitive control approach. The solutions for this optimal control problem depend on a risk-sensitivity parameter, which defines the attitude towards the partner in case of disagreement. The assistive performance of the proposed interaction control scheme is evaluated in a psychological experiment with naive human users. Not only objective measures such as human effort, power or disagreement are evaluated but also a subjective evaluation of the perceived helpfulness. Results indicate better performance in terms of human effort minimization and perceived helpfulness when uncertainty is considered.

The remainder of this section is structured as follows. Section 5.3.1 describes the interaction control. The anticipatory assistive control is explained in Section 5.3.2. An implementation of the 2-dimensional translational case is detailed in Section 5.3.3. Results of simulations and of the experimental user study are presented in Section 5.3.4.

5.3.1 Problem Formulation

In this section we consider the class of physical human-robot interaction tasks where the robot is physically coupled to a human partner. The goal is to reach a goal configuration starting from an initial configuration. This prototypical setting is representative for many different tasks such as mobility assistance to humans or joint object transport/manipulation. Depending on the application, the interaction can be through an external object, as in cooperative manipulation tasks, or at the end-effector as in movement assistance for elderly or disabled or in exoskeletons. In case of the interaction through an object, its geometry together with the grasping points lead to a decomposition into redundant and non-redundant forces [49]. Redundant components can be exerted by either the robot or the human and are instrumental for the haptic negotiation process. Here, we focus our attention into interaction forces which only appear in redundant degrees of freedom, as explained in Section 5.2. As a representative case for this setting, we consider a common interaction contact point between the robot and its human partner at the end-effector. For simplicity of exposition we assume that the nonlinear robot dynamics is feedback-linearized and an impedance control law renders the robot reactive behavior [168] to

$$M_r\ddot{\boldsymbol{x}}(t)+D_r\dot{\boldsymbol{x}}(t)=\boldsymbol{u}(t)=\boldsymbol{u_h}(t)+\boldsymbol{u_r}(t) \tag{5.23}$$

where $\boldsymbol{x}\in\mathbb{R}^6$ is the pose of the end-effector, $\boldsymbol{u_h}\in\mathbb{R}^6$ is the applied wrench by the human, $\boldsymbol{u_r}\in\mathbb{R}^6$ is the assistive control input of the robot, and $M_r, D_r\in\mathbb{R}^{6\times 6}$ are positive definite matrices representing the rendered inertia and viscous friction respectively. For later convenience, we discretize the system from (5.23) with a sampling time interval Δ, yielding impedance controlled dynamics

$$\boldsymbol{\xi}_{k+1}=\boldsymbol{f}(\boldsymbol{\xi}_k,\boldsymbol{u}_{\boldsymbol{h},k}+\boldsymbol{u}_{\boldsymbol{r},k})\ , \tag{5.24}$$

where k is the time index such that $t=k\Delta$ and $\boldsymbol{\xi}_k=[\ \boldsymbol{x}_k\ \dot{\boldsymbol{x}}_k\ ]^{\mathrm{T}}$ the state of the system [2].

Dynamics (5.23) compensates for small deviations between the human and robot intended motions. In addition to this *reactive behavior* represented by the compliance control, the robot implements an anticipatory *proactive behavior* given by $\boldsymbol{u_r}$. To this end, the robot relies on a behavior model λ, which provides predictions of the next human desired state and human internal wrench in terms of normal distributions $\hat{\boldsymbol{\xi}}_{\boldsymbol{d},k}\sim\mathcal{N}(\hat{\boldsymbol{\mu}}_{\xi,k},\hat{\boldsymbol{\Sigma}}_{\xi,k})$ and $\hat{\boldsymbol{u}}_{\mathrm{int},k}\sim\mathcal{N}(\boldsymbol{u}_{\mathrm{int},k}|\hat{\boldsymbol{\mu}}_{\boldsymbol{u}_{\mathrm{int}},k},\hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}_{\mathrm{int}},k})$. The behavior model represents expected trajectories based on multiple previously performed trials. The resulting normally distributed state space predictions represent the trial to trial statistical mean and variance of human desired motion. In contrast, internal human wrench components, studied in depth in Section 5.2, represent counteracting wrenches not influencing dynamics (5.23) that imply disagreement between

[2] The time-discretized system dynamics will be specified in Section 5.3.3 for a two-dimensional example.

human and robot or, similarly, discrepancies between desired accelerations. This additional unnecessary effort is not observable in state space trajectories. Consider as an example a binary decision problem while approaching an obstacle, where both agents disagree on which way to go, either left or right. We further assume that both partners have a common goal and deviations from the predicted trajectory induced by the human are small and do not express a divergence or a tendency towards a different goal. Given this problem setting, the focus of this work lies on the design of a possibly time-varying assistive control law synthesizing $\boldsymbol{u}_r$ taking uncertainties into account.

5.3.2 Anticipatory Assistance Based on Learned Models

In this work, the objective governing haptic assistance is human force minimization. From an optimality point of view, this concept is expressed by the minimization of a cost function in the form

$$J = \sum_{k=0}^{T-1} ||\boldsymbol{u}_{h,k}||^2 \tag{5.25}$$

where T is the time horizon for the optimization [3]. The human has an urge towards a goal as predicted by the task model λ but relying on a human force estimation of T steps potentially accumulates high tracking errors in task space. A more suitable alternative considers the minimization problem in task space based on the tracking error between task model predictions $\hat{\boldsymbol{\xi}}_{\boldsymbol{d}} = \{\hat{\boldsymbol{\xi}}_{\boldsymbol{d},0} \cdots \hat{\boldsymbol{\xi}}_{\boldsymbol{d},T}\}$ and the current state $\boldsymbol{\xi}$, i.e.

$$J = \sum_{k=0}^{T} ||(\hat{\boldsymbol{\xi}}_{\boldsymbol{d},k} - \boldsymbol{\xi}_k)||^2 \; . \tag{5.26}$$

The solution to this optimization problem leads to controller which does not consider any energy expenditure of the robot nor any torque constraints of the robot actuators. In addition, disturbances to the tracking may result in rather abrupt behavior of the robot, which might not be desirable when targeting intuitive assistance. In order to keep the computational complexity low to achieve real-time computation, we add a soft penalty on the energy expenditure in terms of a quadratic term. As a result the cost function takes the form as in the classical linear quadratic optimal control problem

$$J = ||(\hat{\boldsymbol{\xi}}_{\boldsymbol{d},T} - \boldsymbol{\xi}_T)||^2_{Q_T} + \sum_{k=0}^{T-1} ||(\hat{\boldsymbol{\xi}}_{\boldsymbol{d},k} - \boldsymbol{\xi}_k)||^2_{Q_k} + ||\boldsymbol{u}_{\boldsymbol{h},k}||^2_{R_k} \tag{5.27}$$

where $||x||^2_Q$ stands for the quadratic form $x^{\mathrm{T}}Qx$, Q_T the final cost weight and Q_k and R_k are weighting factors that define the trade-off between tracking performance and robot contribution. Observe that two factors challenge the equivalence between (5.25) and human intentions: i) the prediction error of $\hat{\boldsymbol{\xi}}_{\boldsymbol{d}}$; ii) the weighting factors Q_k and R_k may differ from human preferences. As perfect predictions and weighting factors design is usually not feasible when dealing with humans, corrections/disagreements from the human partner are

[3] Note that we will later employ a receding horizon control scheme, i.e. without fixed final time T. For simplicity of presentation we consider for now the finite horizon control problem (5.25)

expected. This source of variability, together with the normally distributed nature of predictions define the uncertainty that the robotic assistance needs to cope with. In the following we analyze the consequences of uncertainty in the robot behavior and synthesize their effect as variations in the cost function to optimize (5.27) and in the predicted dyad dynamics.

Remark 5.1. Note that in (5.26) and (5.27) we neglect human internal wrench predictions. A valid alternative considers a cost function in the form

$$J = \|(\hat{\xi}_{\boldsymbol{d},T} - \xi_T)\|^2_{Q_T} + \|\boldsymbol{u}_{\text{int},T}\|^2_{Q'_T} + \sum_{k=0}^{T-1} \|(\hat{\xi}_{\boldsymbol{d},k} - \xi_k)\|^2_{Q_k} + \|\boldsymbol{u}_{\text{int},k}\|^2_{Q'_k} + \|\boldsymbol{u}_{\boldsymbol{h},k}\|^2_{R_k} \ ,$$

where Q'_k is an additional weighting factor. This cost explicitly accounts for the predicted human internal wrench in order to compensate for expected counteracting wrench components. Our problem setting assumes that the robot assistance does not hinder the human and predictions $\hat{\xi}_{\boldsymbol{d}}$ are close to human intentions. In this case, internal wrenches are close to zero in expectation , i.e. $\mathrm{E}[\boldsymbol{u}_{\text{int}}] \approx \boldsymbol{0}$, yielding (5.27).

Remark 5.2. Following the analysis from Section 5.2, the minimization of cost function (5.27) yielding the robot assistive input $\boldsymbol{u_r}$ implicitly assumes an a priori load share distribution $\{\alpha_{f,h} = 0, \alpha_{\tau,h} = 0\}$ and $\{\alpha_{f,r} = 1, \alpha_{\tau,r} = 1\}$, i.e. a priori the robot accounts for all load. Other load sharing policies are straightforwardly computed by multiplying the resulting input by the desired decomposition matrix A from Proposition 5.1.

Considering motion prediction uncertainty

Cost function (5.27) depends on the sequence of multivariate normal distributions $\hat{\xi}_{\boldsymbol{d}}$. A classical realization of the cost function for optimal tracking control measures the Euclidean distance to the reference's mean, i.e.

$$J = \|(\hat{\boldsymbol{\mu}}_{\xi,T} - \xi_T)\|^2_{Q_T} + \sum_{k=0}^{T-1} \|(\hat{\boldsymbol{\mu}}_{\xi,k} - \xi_k)\|^2_{Q_k} + \|\boldsymbol{u}_{\boldsymbol{h},k}\|^2_{R_k} \ . \tag{5.28}$$

However, in order to accordingly measure the distance to a multivariate Gaussian, the weighted *Mahalanobis distance* is a suitable option as it includes the covariance of the prediction into the distance metric. In this case, the cost function becomes

$$J = \|(\hat{\boldsymbol{\mu}}_{\xi,T} - \xi_T)\|^2_{\hat{Q}_T} + \sum_{k=0}^{T-1} \|(\hat{\boldsymbol{\mu}}_{\xi,k} - \xi_k)\|^2_{\hat{Q}_k} + \|\boldsymbol{u}_{\boldsymbol{h},k}\|^2_{R_k} \ , \tag{5.29}$$

where $\hat{Q}_k = \hat{\boldsymbol{\Sigma}}_{\xi,k}^{-\frac{1}{2}} Q_k \hat{\boldsymbol{\Sigma}}_{\xi,k}^{-\frac{1}{2}}$ and $\hat{Q}_T = \hat{\boldsymbol{\Sigma}}_{\xi,T}^{-\frac{1}{2}} Q_T \hat{\boldsymbol{\Sigma}}_{\xi,T}^{-\frac{1}{2}}$ and $\hat{\boldsymbol{\Sigma}}_{\xi,i}$ is per definition a symmetric positive-definite matrix $\forall i \quad 0 \leq i \leq T$. This formulation diminishes the cost of tracking errors in directions with high prediction variance while it increases the penalty of deviations in regions with low variance. This is a desirable behavior in most applications where a a low variance of the learned desired trajectory over repeated trials indicates some importance to keep track of it. As an example consider the situation depicted in Fig. 5.11. Low variance directions might be caused for example by environmental constraints while high variance regions imply

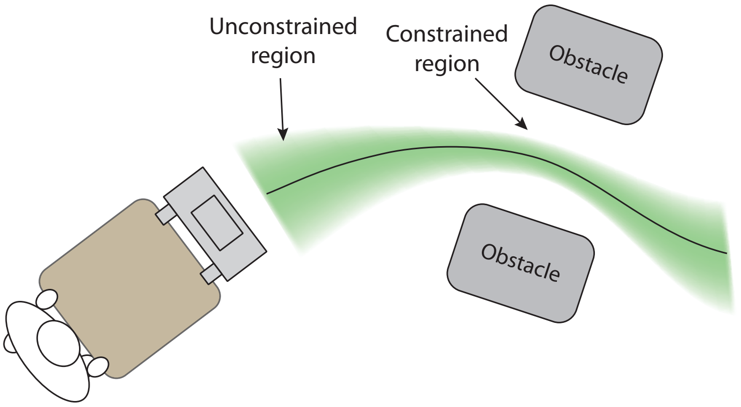


Figure 5.11: Exemplary representation of the predicted state mean and variance $\hat{\xi} = \{\hat{\mu}_{\xi}, \hat{\Sigma}_{\xi}\}$. Low variance directions suggest the possibility of constrained passages, while high variances areas imply unconstrained areas.

unconstrained areas. This suggests flexibility in unconstrained directions and strict tracking through narrow passages, a behavior in accordance with the minimization of the Mahalanobis distance as in (5.29). Note that the opposite effect is achieved using the inverse of the covariance matrix, i.e. $\hat{Q}_k = \hat{\mathbf{\Sigma}}^{\frac{1}{2}}_{\xi,k} Q_k \hat{\mathbf{\Sigma}}^{\frac{1}{2}}_{\xi,k}$ and $\hat{Q}_T = \hat{\mathbf{\Sigma}}^{\frac{1}{2}}_{\xi,T} Q_T \hat{\mathbf{\Sigma}}^{\frac{1}{2}}_{\xi,T}$. In this case the cost increases inversely with the variance: directions with low predicted variance diminish their cost with respect to high variance regions, which increase their penalty. The idea of employing the measure of variance as a way to inform the system on how systematically it should follow a reference trajectory was also applied in [169] for an impedance control scheme and in [63] for kinematic control.

Considering disagreement

In addition to motion prediction, the interaction behavior model λ also provides human internal wrench predictions $\hat{\boldsymbol{u}}_{\text{int}}$. We assume that there exists a human desired nominal motion trajectory. In consequence, if the task model λ contains sufficiently many trials, the mean of the predicted trajectory $\hat{\boldsymbol{\xi}}_d$ approaches the human desired nominal motion trajectory, tracked by the robot. From proposition 5.4, if desired resulting wrenches match in direction no internal wrenches are expected and therefore $\mathrm{E}[\boldsymbol{u}_{\text{int},k}] = \hat{\mu}_{\boldsymbol{u}_{\text{int}},k} \approx \mathbf{0}$. However, human internal wrench covariance encodes valuable information: regions with high variance are caused by significant or recurrent inter-trial corrections, meaning a conflictive area, see Fig. 5.12 for an illustration. To model this unbiased variability, we denote $\boldsymbol{\epsilon_e} \sim \mathcal{N}(\boldsymbol{\epsilon_e}|\mathbf{0}, \hat{\mathbf{\Sigma}}_{\boldsymbol{u}_{\text{int}}})$ the *expected disagreement* with the human partner. Applying the same idea to current observations of $\boldsymbol{u}_{\text{int}}$, an estimation of the *current disagreement*, $\boldsymbol{\epsilon_c} \sim \mathcal{N}(\boldsymbol{\epsilon_c}|\mathbf{0}, \mathbf{\Sigma}_{\boldsymbol{u}_{\text{int}}})$ is computed empirically at time k as the sample covariance over a window of W samples

$$\mathbf{\Sigma}_{\boldsymbol{u}_{\text{int}}} = \frac{1}{W} \sum_{i=k-W+1}^{k} \boldsymbol{u}_{\text{int},i} \boldsymbol{u}^{\mathrm{T}}_{\text{int},i} \ . \tag{5.30}$$

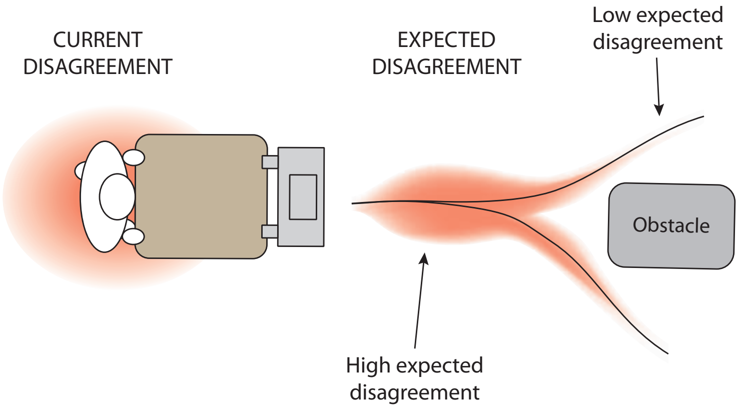


Figure 5.12: Schematic representation of the expected disagreement $\boldsymbol{\epsilon_e}$ and the current disagreement $\boldsymbol{\epsilon_c}$.

To consider both estimations, see Fig. 5.12 for visualization, the maximum of both disagreements

$$\boldsymbol{\epsilon} = \max(\boldsymbol{\epsilon_e}, \boldsymbol{\epsilon_c}) \ , \tag{5.31}$$

is considered a reasonable solution. Note that keeping the zero mean assumption for the current disagreement is a conservative approach. In fact, the intra-trial mean along a window W that the current disagreement considers may differ from zero. Interpreting human wrench corrections as unbiased entails higher uncertainty for the robot instead of (signed) errors. The consequences of this design will be more apparent in the analysis of the optimal control solution.

Disagreements result in variability of the dyad's motion due to conflicts with the human partner. Expressing this idea in mathematical terms, dynamics (5.24) contains an additional additive disagreement vector $\boldsymbol{\epsilon}$, yielding the stochastic dyad dynamics

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{f}(\boldsymbol{\xi}_k, \boldsymbol{u}_{\boldsymbol{r},k} + \boldsymbol{\epsilon}_k) \ . \tag{5.32}$$

Given initial state $\boldsymbol{\xi}_0$, the optimal robotic assistance results from minimizing cost function (5.29), i.e.

$$\min_{\boldsymbol{u}_{r,0}\cdots\boldsymbol{u}_{r,T-1}} J(\boldsymbol{\xi}_0, \boldsymbol{u}_{r,0}\cdots\boldsymbol{u}_{r,T-1}) \ , \tag{5.33}$$

constrained to the dyad dynamics (5.32). As the dynamics are constantly updated with the current disagreement from (5.30), a recalculation of the solution is necessary. To this end, we adopt a Model Predictive Control (MPC) scheme [170], where the dynamics (5.32) are considered stationary for the optimization horizon T and the optimal control problem is solved at every time step. Cost function (5.29) at time step k formulated in a receding horizon fashion becomes

$$J_k = ||(\hat{\boldsymbol{\mu}}_{\xi,k+T} - \boldsymbol{\xi}_{k+T})||^2_{\hat{Q}_T} + \sum_{i=k}^{k+T-1} ||(\hat{\boldsymbol{\mu}}_{\xi,i} - \boldsymbol{\xi}_i)||^2_{\hat{Q}_i} + ||\boldsymbol{u}_{\boldsymbol{r},i}||^2_{R_k} \ . \tag{5.34}$$

Although the optimization is solved for a horizon of T steps, only the solution for the first step is applied as assistive robot control.

Remark 5.3. If the behavior model is constantly updated in an online fashion, e.g. the model presented in Section 3.3, the estimation of current disagreements (5.30) is not necessary.

Risk-sensitive optimization

Problem (5.33) is a stochastic optimization problem due to the stochastic nature of the dynamics (5.32). Traditionally it is solved by minimizing the *expected cost*, $\mathsf{E}[J]$. However, the uncertainty induced by the human behavior represented in $\boldsymbol{\epsilon}$ is not characterized by its expected value but its higher order moments. In order to capture the uncertain human influence in the dynamics here we propose to employ a risk-sensitive optimization [106, 107]. The risk-sensitive solution considers all infinite cumulants of the cost distribution [109]. The risk-sensitive optimization criterion considers a cost function in the form

$$\gamma_k(\theta) = -2\theta^{-1} \ln \mathsf{E}[\exp^{-\frac{1}{2}\theta J_k}] \,, \tag{5.35}$$

where $\theta \in \mathbb{R}$ is the risk-sensitivity parameter and J_k is given in (5.34). The Taylor series expansion of $\gamma(\theta)$ around $\theta = 0$,

$$\gamma(\theta) = \mathsf{E}[J] - \frac{1}{4}\theta \,\mathsf{Var}[J] + \dots \,,$$

shows the influence of higher order moments in the solution, parameterized by θ. If $\theta = 0$ the controller is risk-neutral and corresponds to the classical optimization of the expected cost. For $\theta < 0$ and $\theta > 0$ the controller becomes risk-averse and risk-seeking, respectively. In our robot assistance scenario, the variance of the cost is determined by previously experienced and current disagreements with the human partner (5.31), and θ determines its assessment. In the risk-seeking case, disagreement is considered beneficial and decreases the resulting cost. In contrast, a risk-averse controller considers disagreements as a detrimental influence and the cost increases.

5.3.3 The 2-dimensional Translational Case

The risk-sensitive optimization problem can be solved efficiently through the Riccatti equation for linear dynamics. Real-time computations are fundamental for intuitive haptic assistance; the physical coupling between human and robot requires immediate adaptation. To satisfy this prerequisite, we limit our study to a linear implementation. We consider now dynamics (5.23) in the plane, i.e. $\boldsymbol{x} \in \mathbb{R}^2$. In this case, the discrete-time dynamics from (5.32), written in the form $\xi_{k+1} = A\xi_k + \boldsymbol{B}\boldsymbol{u}_k$ are given by

$$\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \dot{\boldsymbol{x}}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta \\ 0 & 1 - M_r^{-1} D_r \Delta \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \dot{\boldsymbol{x}}_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & M_r^{-1}\Delta \end{bmatrix} (\boldsymbol{u}_{r\,k} + \boldsymbol{\epsilon}_k) \,. \tag{5.36}$$

where $M_r, D_r \in \mathbb{R}^{2\times 2}$.

The disagreement $\boldsymbol{\epsilon}_k$ from (5.31), given by the maximum of the expected and current disagreement is approximated by $\mathcal{N}(\boldsymbol{\epsilon}|0, \widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}_{\text{int}}})$, where $\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}_{\text{int}}}$ is the Löwner-John hyperellipsoid [171]. This Gaussian approximation calculates the minimum volume hyperellipsoid around the set defined by $\mathcal{N}(\boldsymbol{\epsilon}_e|\boldsymbol{0}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}_{\text{int}}})$ and $\mathcal{N}(\boldsymbol{\epsilon}_c|\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{u}_{\text{int}}})$.

For this particular problem setting and for comparison purposes we present two different solutions: a classical linear quadratic regulator (LQR), which considers the reference prediction uncertainty (5.34) but discards the variability of the disagreement in its optimal solution and, in contrast, a linear exponential quadratic regulator (LEQR) which considers both uncertainty sources in its risk-sensitive optimization (5.35).

Linear Quadratic Regulator Solution

The minimization of the expected cost $\mathrm{E}[J_k]$ with J_k as in (5.34) and dynamics (5.36) is an instance of an LQ problem described in Chapter 4. Considering only the regulation term, the solution to this problem at sample time k for a receding horizon of T steps yields a feedback control law given by

$$\boldsymbol{u_r}_i = -K_i(\hat{\boldsymbol{\mu}}_{\xi,i} - \boldsymbol{\xi}_i)\,, \tag{5.37}$$

where K_i is the feedback matrix of the Ricatti recursion

$$K_i = -R^{-1}B^{\mathrm{T}}(BR^{-1}B^{\mathrm{T}} + \Pi_{i+1}^{-1})^{-1}A \tag{5.38}$$

and the cost-to-go is given by

$$\Pi_i = \hat{Q}_i + A^{\mathrm{T}}(BR^{-1}B^{\mathrm{T}} + \Pi_{i+1}^{-1})^{-1}A\,, \tag{5.39}$$

being $\Pi_T = \hat{Q}_T$.

Only the feedback matrix for time step $i = k$, K_k, is applied in the system. It is in the form

$$K_k = \begin{bmatrix} 0\;0 \\ K_{x,k}\;K_{\dot{x},k} \end{bmatrix}, \tag{5.40}$$

where $K_{x,k}$ and $K_{\dot{x},k}$ are the position and velocity gains respectively.

The resulting assistive tracking control synthesizes a variable impedance control law with its gains depending on prediction uncertainty. This becomes more obvious when applying the control law for the limit $\Delta \to 0$ in the system dynamics (5.23)

$$M_r\ddot{\boldsymbol{x}} + D_r\dot{\boldsymbol{x}} = \boldsymbol{u_h} - (K_x(t)(\boldsymbol{x_d} - \boldsymbol{x}) + K_{\dot{x}}(t)(\dot{\boldsymbol{x}}_{\boldsymbol{d}} - \dot{\boldsymbol{x}}))\,, \tag{5.41}$$

where $\boldsymbol{x_d}$, $\dot{\boldsymbol{x}}_{\boldsymbol{d}}$, $K_x(t)$ and $K_v(t)$ are the position and velocity components of $\hat{\boldsymbol{\mu}}_\xi$ and the feedback gains $K_{x,k}$ and $K_{v,k}$ in the limit $\Delta \to 0$ respectively.

This control strategy tracks the predicted trajectory considering its variance by means of the Mahalanobis distance present in $\hat{Q}_i$ and $\hat{Q}_T$. As a result, the robot gets stiffer in the directions where the motion prediction has low variance and becomes more compliant for high variance. However, the variability induced by disagreements with the human partner is neglected.

Linear-Exponential Quadratic Regulator

The optimization problem considers now cost function (5.35) with J_k as in (5.34). In this case, as detailed in Chapter 4, the regulation term of this optimization problem follows from a modified form of the standard Ricatti recursion [172]

$$K_i = -R_i^{-1}B^{\mathrm{T}}(BR_i^{-1}B^{\mathrm{T}} + \theta B\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}_{\mathrm{int}},i}B^{\mathrm{T}} + \Pi_{i+1}^{-1})^{-1}A\,, \tag{5.42}$$

and

$$\Pi_i = \hat{Q}_i + A^{\mathrm{T}}(BR_i^{-1}B^{\mathrm{T}} + \theta B\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}_{\mathrm{int}},i}B^{\mathrm{T}} + \Pi_{i+1}^{-1})^{-1}A\,, \tag{5.43}$$

with $\Pi_T = \hat{Q}_T$.

As in the LQR case, motion prediction uncertainty is equivalently taken into account by means of the Mahalanobis considered in $\hat{Q}_i$ and $\hat{Q}_T$. However, the additive term $\theta B\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}_{\mathrm{int}},i}B'$ present in the risk-sensitive solution leads to different results. The risk-neutral case $\theta = 0$ yields the same solution as for the LQR canceling the additional term. For a risk-seeking optimization $\theta > 0$ the expected variability $\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{u}_{\mathrm{int}},i}$ is assumed to be collaborating as if it was doing part of the work towards fulfilling the task. Accordingly, the resulting gain from (5.42) decreases adopting a more compliant behavior. Understanding the risk-sensitivity parameter as the robot's attitude, a *recessive* attitude is achieved implementing a risk-seeking controller: the robot becomes compliant under disagreement with its partner. In contrast, the risk-averse solution $\theta < 0$ increases the overall cost as if variability were directing the system towards an undesired state. As a result, gains are increased and the robot becomes stiffer. This case corresponds to a *dominant* attitude by generating an aggressive response to disagreements.

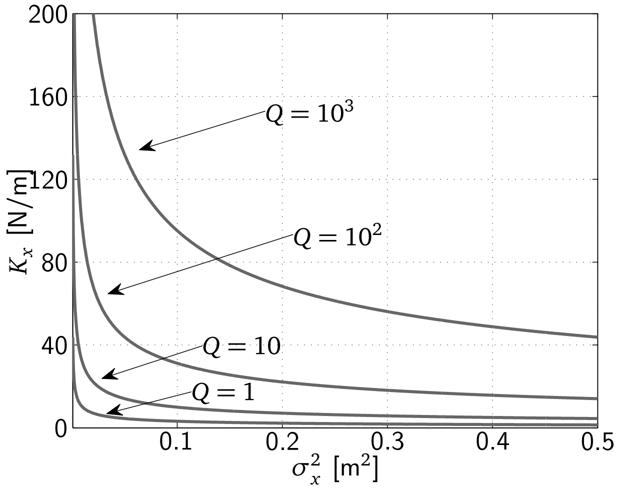


Figure 5.13: Infinite horizon position gains w.r.t position prediction variance for several precision parameters Q with $R = I$.

5.3.4 Evaluation

In order to illustrate the peculiarities and advantages of the proposed controller, we first observe the influence of both prediction uncertainty and negotiation variability on a one-dimensional simulation. For evaluation purposes, we further study human preferences performing a user study where 19 participants interact with 7 different assistive controllers in a virtual scenario using a haptic interface.

One-dimensional simulation

As an exemplary scenario, we study a one-dimensional mass-damper system with dynamics equivalent to (5.36), $M = 1$ kg and $D = 1$ Ns/m. We first analyze the influence of motion

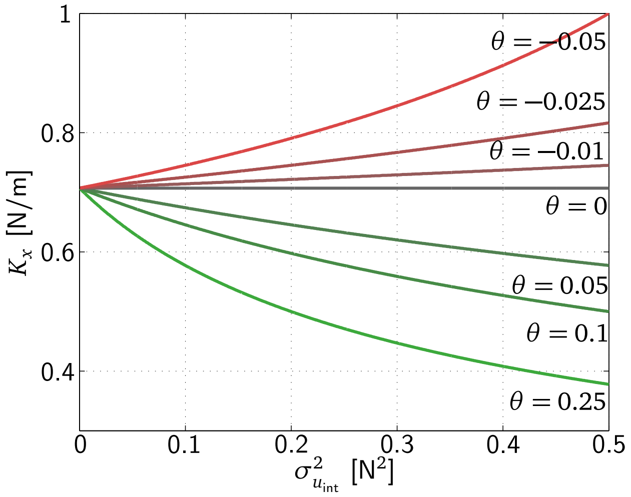


Figure 5.14: Infinite horizon position gains w.r.t process noise variance for several risk-sensitivities in an optimization with $Q = R = I$.

prediction uncertainty using the Mahalanobis distance on the robot assistive behavior following the control scheme proposed in Section 5.3.3. We assume constant weighting matrices $Q = Q_k$ and $R = R_k$. The optimal position tracking gains for an LQR control optimizing the cost (5.29) for the infinite horizon $T \to \infty$ w.r.t. prediction variance σ_x^2 are depicted in Fig. 5.13. The infinite horizon solution provides a stationary feedback gain [173]. The optimal gains decrease with increasing prediction variances, i.e. low prediction variance produces a stiffer robot assistive behavior while high uncertainty leads to a more compliant assistive behavior. Note that, as commented in Section 4.5, when variances tend to zero, gains tend to infinity. This degenerate case is easily avoided adding a regularization quantity, ensuring positive definiteness. For the case of the identity matrix, the weighting matrices of (5.29) become $\hat{Q}_k = (I + \hat{\Sigma}_{\xi,k})^{-\frac{1}{2}} Q (I + \hat{\Sigma}_{\xi,k})^{-\frac{1}{2}}$ and $\hat{Q}_T = (I + \hat{\Sigma}_{\xi,T})^{-\frac{1}{2}} Q_T (I + \hat{\Sigma}_{\xi,T})^{-\frac{1}{2}}$. As a result, when variances tend to 0, the cost generalizes to (5.28).

The influence of disagreement $\sigma^2_{u_{\text{int}}}$ on the optimal gains of a risk-sensitive controller from Section 5.3.3 is shown in Fig. 5.14 for several values of θ. While the risk-neutral solution $\theta = 0$ ignores disagreement, risk-seeking solutions $\theta > 0$ decrease gains as the disagreement increases. In contrast, risk-averse solutions $\theta < 0$ increase tracking gains with increasing disagreement. An intuitive explanation in our scenario follows. A risk-seeking assistant assumes disagreement accounts for part of the tracking task and reduces its gains adopting a recessive attitude. In contrast, a risk-averse assistant understands disagreement as counter-productive and increases its gains in order to perform the tracking task while correcting expected deviations.

In order to illustrate how disagreement affects the dyad behavior during negotiation, we simulate a coupled dyad consisting of agents a and b, with control inputs $\boldsymbol{u_a}$ and $\boldsymbol{u_b}$ respectively. Consider agent b behaves as a PD-controller with constant gain, i.e.

$$\boldsymbol{u}_{\boldsymbol{b},k} = K_b(\boldsymbol{\xi_b} - \boldsymbol{\xi}_k) \ . \tag{5.44}$$

In contrast, agent a implements a risk-sensitive control strategy and tracks a constantly conflicting reference $\boldsymbol{\xi_a} = -\boldsymbol{\xi_b}$. We focus now on the effects that the current disagreement,

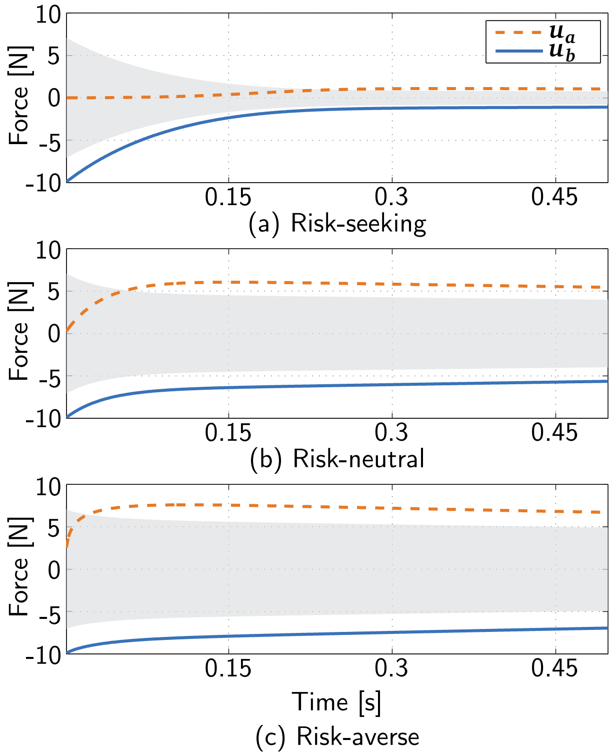


Figure 5.15: Simulation results for the one-dimensional linear mass-damper system example with initial state $\xi_0 = [0\ 0]^{\mathrm{T}}$, references $\xi_b = -\xi_a = [-1\ 0]^{\mathrm{T}}$, mass $m = 1$ kg, damping $d = 1$ Ns/m with two agents, agent b following the control law (5.44) with gain $K_a = 1$ N/m and agent a applying the proposed control scheme in three variants. All three controllers are calculated with parameters $Q = I$ and $R = 5 \cdot 10^{-4} \cdot I$ and sample time of 1 kHz . Fig. 5.15(b) shows the simulation result for the risk-neutral controller, Fig. 5.15(a) for a risk-seeking controller with $\theta = 10^6$ and Fig. 5.15(c) a risk-averse controller with $\theta = -4.5 \cdot 10^2$. Grey regions represents the variance of disagreement considered by agent a.

.

computed according to (5.30) with $W = 1$, produces on agent a. For simplicity of illustration we neglect the expected disagreement. Note that the expected disagreement or the maximum of both disagreements produce equivalent effects on the control gains. The simulation results for a risk-neutral, risk-seeking and risk-averse implementation are depicted in Fig. 5.15. As shown in Fig. 5.15(a), the risk-seeking control remains almost inactive in comparison with the risk-neutral controller depicted in Fig. 5.15(b). In contrast, the risk-averse controller reacts more aggressively, as shown in Fig. 5.15(c). These three different attitudes yield different disagreement levels, depicted as gray regions. As the risk-seeking controller ceases tracking its conflicting reference, the disagreement is significantly reduced, while the risk-averse implementation provokes even higher disagreement levels than the risk-neutral solution. This behavior is clearly explained by the resulting gains and positions for all three controllers, as depicted in Fig. 5.16. While the risk-neutral solution ignores disagreement producing constant gains, the risk-seeking controller's almost cancels its feedback gain due

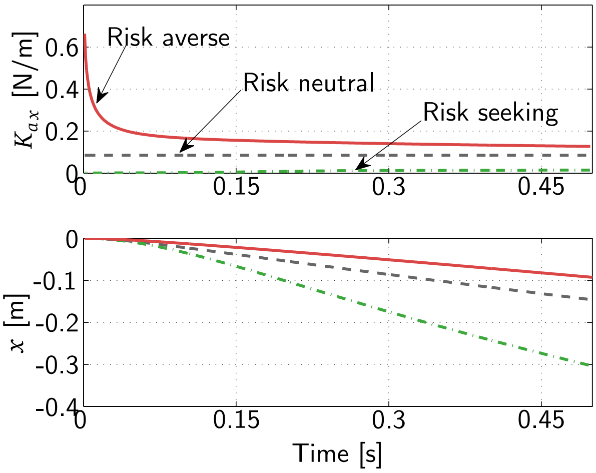


Figure 5.16: Resulting trajectory and tracking gains for agent a in the simulated example from Fig. 5.15.

to the initial conflicts, adopting a recessive attitude. In contrast, the risk-averse controller increases its gain substantially reacting in a dominant fashion.

User Study

To evaluate the proposed controllers and the preferences of humans interacting with it, we designed an experiment in which a human actor has to transport a virtual object from an initial position towards a defined goal position.

Human Behavior Model Acquisition: an interaction behavior model must reflect both the human task execution preferences and the interaction with the robotic partner. Due to complexity, we rely on learning by demonstration techniques. A preliminary model of the task was acquired by initially letting the human lead, i.e., $\boldsymbol{u}_r = 0$. With this first rough representation, the robot actively assists and additionally observes internal wrench patterns during further task trials. Modeling state and human internal wrench input trajectories, $\{\xi, \boldsymbol{u}_{\text{int}}\}$, the robot acquires a task model that represents both the human desired state trajectory and the expected disagreement levels. Observations are encoded into a time-based HMM, later used for prediction as presented in Section 3.1. For simplicity, in our implementation we rely on the complete human wrench $\boldsymbol{u}_h$ instead of internal wrench components $\boldsymbol{u}_{\text{int}}$, yielding higher uncertainty.

Experimental Setup: the human actor applies forces to a haptic interface in order to move the virtual object using the virtual reality interface described in appendix A.1. The shared object from dynamics (5.36) is physically rendered with a mass of $M_r = diag\{m, m\}$ with $m = 90$ kg and damped by a viscous friction of $D_r = diag\{d, d\}$ with $d = 200$ Ns/m, emulating a heavy object. All virtual obstacles and walls are haptically rendered in order to provide a haptic feedback of the environment to the user. The HMM used to encode observations has 40 states and predictions are updated with a rate of 50Hz.

On top of the interface, a virtual maze is presented on a screen, see Fig. 5.17. On the right side the maze includes two obstacles moving horizontally in order to potentially provoke disagreements between the human and the robot. Note that the human behavior model

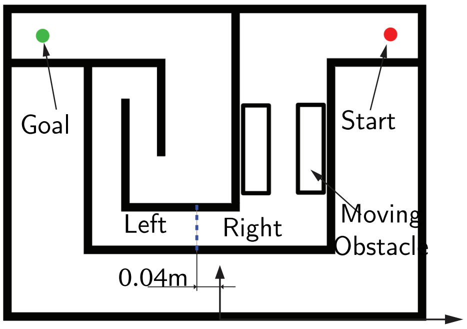


Figure 5.17: Virtual environment (maze). With the handle of the two-dimensional haptic interface participants were able to move the virtual object (red dot) to the goal position (green dot).

does not consider the obstacle's state, which results in an increased predicted variance in the region perturbed by the obstacle's motion. This condition aims for evaluating the assistive performance when the model is not accurate enough and variability becomes more relevant. On the left side, a navigation task requires high movement precision caused by a narrow and curved path.

Task and Procedure: in total 19 persons (5 female) participated in the experiment. They were between 23 and 31 years old ($M = 26.84$ years). The experimental task was to move the virtual object represented by a red dot from the upper right corner (start position) to the upper left corner of the maze (green dot at target position) without hitting any object or wall, see Fig. 5.17. Participants were instructed to move at their comfortable speed and to finish the task even if they might hit the wall or an object.

Overall the experiment was divided into 7 parts corresponding to 7 different control laws of 3 trials each. In the first trial, the human lead together with a passive robot, i.e. $\boldsymbol{u}_r = 0$, and the resulting observations were encoded into the task model. This model was used during the second trial, where the robot actively assisted its human partner. This second task execution already captures negotiation forces between the human and the robot. Observations of the first and second trials were used to train the task model for the third trial, providing already an estimation of the expected trajectory, both in state and internal wrench components, $\{\hat{\xi}_d, \hat{\boldsymbol{u}}_{\text{int}}\}$. The third trial was the only one considered for further evaluations. We tested 7 different control laws for each participant:

(a) No active assistance: $\boldsymbol{u}_r = 0$.

(b) LQR using cost function (5.28).

(c) Risk-neutral with Mahalanobis distance: $\theta = 0$ using cost function (5.35).

(d) Risk-averse with Mahalanobis distance and expected process noise: $\theta = -\alpha$, using cost function (5.35) and $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_e$.

(e) Risk-averse with Mahalanobis distance considering both expected and current process noise : $\theta = -\alpha$, using cost function (5.35) and $\boldsymbol{\epsilon} = \max(\boldsymbol{\epsilon}_e, \boldsymbol{\epsilon}_c)$.

Cont.	M_h	M_{P_h}[W]	$M_{\|\|u_D\|\|}$[N]	$M_{\|\|u_h\|\|}$[N]	$M_{\|\|u_c\|\|}$[s]
(a)	3.90	0.61	-	12.92	1.32
(b)	3.42	0.58	3.15	12.73	2.42
(c)	3.58	0.52	2.91	12.30	2.02
(d)	3.47	0.55	2.28	11.85	2.29
(e)	3.92	**0.39**	2.72	**10.46**	1.38
(f)	3.76	0.47	2.17	11.19	**1.03**
(g)	**4.11**	0.54	**1.29**	11.50	1.12

Tabular 5.1: Average values for perceived helpfulness M_h, human power M_{P_h}, disagreement $M_{||\boldsymbol{u_D}||}$, human force $M_{||\boldsymbol{u_h}||}$ and collision forces$M_{||\boldsymbol{u_c}||}$

(f) Risk-seeking with Mahalanobis distance and expected process noise: $\theta = \beta$, using cost function (5.35) and $\boldsymbol{\epsilon} = \boldsymbol{\epsilon_e}$.

(g) Risk-seeking with Mahalanobis distance considering both expected and current process noise: $\theta = \beta$, using cost function (5.35) and $\boldsymbol{\epsilon} = \max(\boldsymbol{\epsilon_e}, \boldsymbol{\epsilon_c})$.

All implementations rely on the solutions presented in Section 5.3.3. In our experiments $\beta = 8.1 \cdot 10^4$ and $\alpha = 8.1 \cdot 10^2$, $R = I$ and $Q_k = Q_T = diag\{\omega_p, \omega_v\}$, being ω_p and ω_v the position and velocity weightings. For (b), we chose $\omega_p = 10^{10}$ and $\omega_v = 10^7$. For (c)-(g) we set $\omega_p = 10^5$ and $\omega_v = 10$, due to the low variance values, in the order of 10^{-5}. The receding horizon for the optimization was $T = 0.2$s and to estimate the current noise we used a window W corresponding to 0.05s.

The experimental procedure was as follows: participants were asked to face the haptic device and grasp the handle. Next, the experimenter initialized the control algorithm and told the participant to start moving. After reaching the green target, participants were asked to free the handle which was moved back to the initial position automatically. Before every third trial participants were verbally informed that this was going to be the trial they had to rate. Subsequently they had to rate the perceived helpfulness when

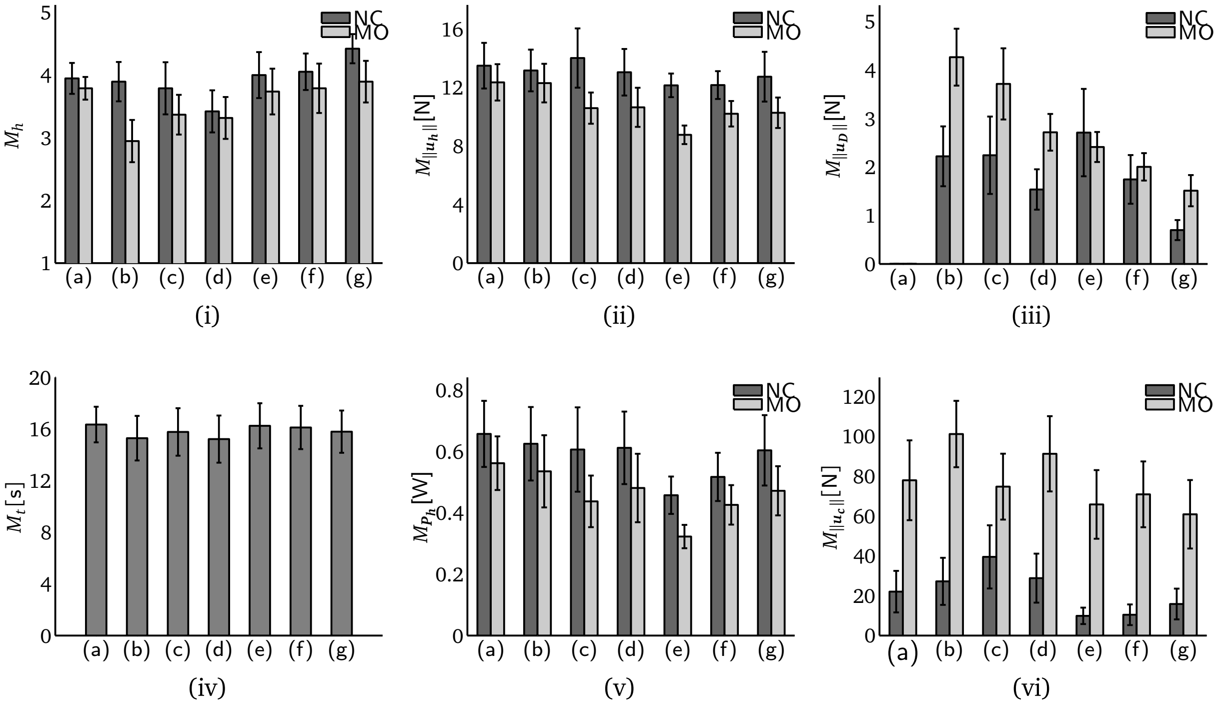
- passing a *moving obstacle* (subtask MO).
- navigating through a *narrow channel* (subtask NC).

Ratings were done on a 6-point scale from 1 (counter productive) to 6 (helpful) and resulted in the explicit measure *perceived helpfulness*. Every participant performed 3 x 7 = 21 trials of which 7 trials were rated.

Regarding implicit measures we evaluated:

- the mean power exerted by the human M_{P_h}, defined as $\frac{1}{t}\int_0^t \boldsymbol{u_h}^{\mathrm{T}}\dot{\boldsymbol{x}}\,d\tau$.
- the mean disagreement $M_{||\boldsymbol{u_D}||}$ between both agents, defined as $\frac{1}{t}\int_0^t ||\boldsymbol{u_D}||\,d\tau$ where

$$\boldsymbol{u_D} = \begin{cases} \dfrac{-\boldsymbol{u_h}}{||\boldsymbol{u_h}||} \cdot \boldsymbol{u_r}, & \text{if } -\boldsymbol{u_h} \cdot \boldsymbol{u_r} > 0 \\ & \quad \wedge \boldsymbol{u_h} \neq \boldsymbol{0} \\ 0, & \text{otherwise.} \end{cases}$$



(i) (ii) (iii)

(iv) (v) (vi)

Figure 5.18: Experimental results. Each of the figures shows the mean and the standard error of the different evaluated measures for the seven control laws (a)-(g). Fig. 5.18(i) shows the mean perceived helpfulness M_h, Fig. 5.18(ii) depicts the human exerted force $M_{\|u_h\|}$, Fig. 5.18(iii) reflects the disagreement $M_{\|u_D\|}$, Fig. 5.18(iv) shows the execution times M_t, Fig. 5.18(v) depicts the human power M_{P_h} and Fig. 5.18(vi) shows the collision forces $M_{\|u_c\|}$.

Perceived Helpfulness M_h			Power M_{P_h}[W]			Disagreement $M_{\|\|u_D\|\|}$[N]		
Comparison	$F(1,18)$	p	Comparison	$F(1,18)$	p	Comparison	$F(1,18)$	p
(a)>(b)	7.15	$< .05$	(a)>(e)	11.35	$< .01$	-	-	-
(e)>(b)	5.90	$< .05$	(b)>(e)	4.54	$< .05$	(b)>(g)	32.47	$< .001$
(g)>(b)	7.08	$< .05$	(c)>(e)	2.58	n.s.	(c)>(g)	16.86	$< .01$
(a)>(d)	5.00	$< .05$	(d)>(e)	2.91	n.s.	(d)>(g)	11.37	$< .01$
(e)>(d)	5.50	$< .05$	(f)>(e)	2.23	n.s.	(e)>(g)	10.26	$< .01$
(g)>(d)	7.21	$< .05$	(g)>(e)	4.46	$< .05$	(f)>(g)	6.08	$< .05$

Tabular 5.2: Overview of results on planned comparisons for perceived helpfulness, human power and disagreement

- the mean human applied force $M_{||u_h||}$, defined as $\frac{1}{t}\int_0^t ||\boldsymbol{u_h}|| d\tau$.
- the mean contact forces during collisions with the virtual environment $M_{||u_c||}$, defined as $\frac{1}{t}\int_0^t ||\boldsymbol{u_c}|| d\tau$, where $\boldsymbol{u_c}$ is the force applied on virtual obstacles.
- the mean execution time M_t.

Means were taken over all participants for the respective controller. To gain knowledge on the human perception and response during different kind of tasks the presented maze was divided into two subtasks. For measures passing the *moving obstacle*, data was calculated until the participant was passing the turning point indicated by the dashed line shown in Fig. 5.17, placed at 0.04 m to the left of the origin. The rest of the task execution corresponded to subtask *narrow channel*. Data analysis was done Matlab and statistical analysis was carried out with SPSS (Statistical Package for the Social Sciences).

Results and Discussion: to access the rating of *perceived helpfulness*, see Fig. 5.18(i), a 2 x 7 repeated measures ANOVA was performed with the between-subject factors subtask (MO vs. NC) and control method (a)-(g). Marginal differences were observed between subtasks, $F(1,18) = 3.95$, $p = .062$, which shows equally perceived helpfulness of the systems response on both sides. Numerically higher ratings for NC($M = 3.96$) compared to MO($M = 3.51$) reflect the fact that the task model is not aware of the moving obstacle and is not as accurate as for NC. Regarding control schemes there was a significant main effect, $F(6,108) = 2.46$, $p < .05$. Comparisons show that control laws (a), (e) and (g) resulted in significantly higher ratings compared to controllers (b) and (d), see Table 5.1 and Table 5.2. This shows that control laws (e) and (g), both risk-sensitive and considering the current disagreement, were perceived as more helpful than method (b), the classical LQR control. It is also remarkable, that numerically all control methods considering the Mahalanobis distance (c)-(g) were rated higher than the classical LQR control (b). Furthermore, only control laws considering the current disagreement in the dynamics, (e) and (g), were numerically rated higher than the pure passive follower (a).

For all implicit measures 2x7 repeated measures ANOVAs were carried out with the between-subject factors subtask(NC vs. MO) and control method (a-g). If the sphericity criterion was not met, Greenhouse-Geisser correction was applied.

Regarding subtasks there was a higher human power applied on NC($M = 0.58$W) compared to MO($M = 0.46$W), $F(1,18) = 15.59$, $p < .01$, see Fig. 5.18(v). After correction, the differences between applied power for the control methods only marginally reached significance, $F(3.0,54.8) = 2.44$, $p = .074$. Concerning control schemes, a tendency towards higher human power applied in control laws (a), (b) and (g) is identified compared to control law (e), see Table 5.2. The risk-averse control law (e) becomes stiffer under the presence of disagreement and, as it also considers the current disagreements with its partner, it is the most agressive partner. The observed differences suggest that this control method reduces the power applied by the human as the robot constantly takes a leading role.

Disagreement, see Fig. 5.18(iii), was marginally higher on MO($M = 2.77$ N) than on NC($M = 1.86$ N), $F(1,18) = 4.22$, $p = .055$, in accordance with the results from perceived helpfulness. The behavior models lacks obstacle awareness and provides low prediction performance for MO thereby yielding disagreements. Looking at the results for control law (g), a highly significant main effect was observed, $F(5,90) = 5.97$, $p < .001$. Planned comparisons show that all control methods cause a higher disagreement than (g), see Table 5.2. As (g) is risk-seeking, it consequently tends to avoid disagreements adopting a more passive role. The fact that it also considers the current disagreement boosts this effect.

Applied forces, Fig. 5.18(ii), were higher on NC($M = 12.97$N) than on MO($M = 10.74$N), $F(1,18) = 15.76$, $p < .01$. Although differences between control methods were not significant after correction, $p > .1$, the risk-averse with current disagreement, control law (e), numerically required the lowest applied force.

On NC, the contact force from collisions, see Fig. 5.18(vi), was smaller ($M = 0.65$ N) compared to MO($M = 2.67$ N), $F(1,18) = 28.73$, $p < .001$.

No significant differences were found between movement times of the different control methods, see Fig. 5.18(iv), $F(6,108) = 0.46$, $p > .8$.

In summary, the consideration of uncertainty in the optimization is shown beneficial for haptic assistance, as exhibited by the superior performance of all proposed control schemes w.r.t a naive LQR control ignoring any variability. The inclusion of the Mahalanobis distance in the optimization criterion leads to higher perceived helpfulness as it considers the learned trajectory's uncertainty. The benefits of a risk-sensitive optimization are only significant when an estimation of current disagreement is additionally considered. The risk-seeking control law achieves minimal disagreement as it adopts a recessive attitude during negotiation with its partner. This is especially noticeable passing through the obstacle: as the behavior model lacks any obstacle awareness, counteracting forces arise frequently due to poor prediction performance. In terms of human power, the risk-averse policy is more effective. However, the risk-seeking variant is preferred by human users, favoring disagreement avoidance to effort minimization. If the behavior model is accurate risk-aversion effectively minimizes human effort but the complexity of human behavior makes fulfilling this prerequisite highly improbable. Nevertheless, other pHRI application domains may take advantage of risk-averse policies. For instance, consider a kinesthetic teaching scenario where the robot has the role of the teacher. In this case the robot aims for minimizing variations in human performance, a condition in accordance with a risk-averse control scheme.

5.4 Uncertainty-dependent Assistance for Tasks with Environmental Contacts

The previous section considers navigation tasks neglecting any contacts with the environment. This possibility is studied here for tasks where the regulation of environmental wrenches is necessary. For simplicity of illustration, we consider a PbD scenario where the human role is limited to teaching. The robot control goal reduces to the reproduction of the demonstrated manipulation task.

Robotic manipulation covers a wide field of applications, such as grasping or door opening tasks [174]. The PbD paradigm [38] provides efficient methods for rapid skill transfer to robots, applicable to manipulation tasks through teleoperation [175] or physical coaching [70]. Manipulators are easily programmed following this principle extracting motion-based generalized plans [74]. Motion-based approaches successfully execute manipulation tasks if reproduced in the same environment [176], but fail in uncertain environments. The inclusion of contact forces into the control scheme facilitates a desired compliant interaction of the manipulator and thereby improves its behavior in contact [1]. Task plans based on demonstrations including contact forces provide better generalization capabilities than pure position models [177]. However, such an approach implies the challenge of controlling two different references at the same time, i.e. position and force. Due to environmental variations, additional adaptation is required during the reproduction phase. One option is the adaptation of the position reference depending on the observed force error using force feedback [178], while other methods follow hybrid control [179] or parallel force and position control [180]. A valid alternative relies on varying the stiffness of the robot depending on the observed position error during kinesthetic teaching [169]. Here we aim for a systematic approach to adapt both, the compliance and the reference of the robotic manipulator based on observed force variability.

In this section we will explicitly address the problem of environment uncertainty during the reproduction of a manipulation task, i.e. we assume a mismatch between the planned position and force trajectories and the current environment condition. As shown in the previous section, uncertainty-dependent optimal control is an effective strategy when the robot control needs to adapt to uncertain human behavior. Here, we show that this concept is also applicable to adapt the trade-off between motion and force goals. We interpret the manipulation tasks as an interaction problem between two agents given by a motion and a force controller. Modeling environmental force variability as an uncertain input for the motion control and following a Model Predictive Control Scheme (MPC), an online variable stiffness as well as a reference adaptation is achieved by means of a risk-sensitive optimization. The proposed control scheme provides increased adaptation in different environmental situations as validated in experiments for an implementation on a KUKA lightweight robot performing two different tasks.

The remainder of this section is structured as follows. Section 5.4.1 describes formally the considered problem. The proposed controller is presented in Section 5.4.2. Experiment are presented in Section 5.4.3.

5.4.1 Problem Formulation

The task considered in this section consist of the control of a robotic manipulator moving from an initial configuration to a final goal following a trajectory of desired position/velocity and desired force. This desired trajectory describes a manipulation task including contacts with the environment. We assume that the manipulator measures contact forces at its end-effector. Its movement can be represented directly in joint or task space. For simplicity we consider that the robot's motion is represented in task space and is governed by an admittance-type control law given by

$$M_r\ddot{\boldsymbol{x}} + D_r\dot{\boldsymbol{x}} = \boldsymbol{u_p} + \boldsymbol{u_f} \ , \tag{5.45}$$

where $\boldsymbol{x}$ is the position of the end-effector and the admittance parameters are given by the rendered mass M_r and friction D_r; $\boldsymbol{u_p}$ represents the motion control input which corrects the error in position and velocity space and $\boldsymbol{u_f}$ the force control input that accounts for errors in force space. We denote the robot's motion state by $\xi = \left[\boldsymbol{x}^{\mathrm{T}}\ \dot{\boldsymbol{x}}^{\mathrm{T}}\right]^{\mathrm{T}}$. The right side of (5.45) reflects the coupling between two controllers, which can be interpreted as *cooperation* only if a common final goal is assumed, i.e. both, the desired force and the desired motion trajectory are followed without the motion and force control counteracting each other. This is achievable only if the task is executed in the same environment as during demonstrations. Otherwise, and this is the more general case that includes environment variability from demonstration to reproduction, it is interpreted as a *competition*.

We assume that the desired motion/force trajectories are encoded in a task model λ, which is extracted from previous human demonstrations and is represented as a trajectory of normal distributions of position/velocity $\{\hat{\mu}_\xi, \hat{\Sigma}_\xi\}$ and force $\{\hat{\mu}_f, \hat{\Sigma}_f\}$. We also consider the possibility of environmental variability during the reproduction of the task, which might not be reflected in the task plan.

A control scheme that reproduces a learned task given by λ would minimize the tracking error with respect to the expected trajectories of both position/velocity $\xi_{\boldsymbol{d}} = \left[\boldsymbol{x_d}^{\mathrm{T}}\ \dot{\boldsymbol{x}}_{\boldsymbol{d}}^{\mathrm{T}}\right]^{\mathrm{T}}$, and force, $\boldsymbol{f}_{\mathrm{env}d}$. From (5.45) we can write

$$M_r\ddot{\boldsymbol{x}} + D_r\dot{\boldsymbol{x}} = (D(\dot{\boldsymbol{x}}_{\boldsymbol{d}} - \dot{\boldsymbol{x}}) + K(\boldsymbol{x_d} - \boldsymbol{x})) + K_f(\boldsymbol{f}_{\mathrm{env}d} - \boldsymbol{f}_{\mathrm{env}}) \ , \tag{5.46}$$

where $\boldsymbol{f}_{\mathrm{env}}$ is the measured force at the end-effector and $\boldsymbol{x_d}$ and $\dot{\boldsymbol{x}}_{\boldsymbol{d}}$ are the desired position and velocity respectively; K, D are the stiffness and damping constituting the motion control and K_f is the proportional gain for the force control.

In order to reproduce the expected forces at the end-effector, the force control scheme tracks the expected force trajectory, i.e., $\mathsf{E}[\boldsymbol{f}_{\mathrm{env}d}] = \hat{\mu}_f$. Note that when K_f differs from the identity matrix, its effect is equivalent to modifying the rest of the admittance parameters as

$$M_r' = M_r K_f^{-1} \quad D_r' = D_r K_f^{-1} \quad D' = DK_f^{-1} \quad K' = KK_f^{-1}.$$

For simplicity, we assume K_f as the identity matrix.

Given this problem setting, the focus of this paper lies on the design of the motion control scheme that generates $\boldsymbol{u_p}$, i.e. on the design of the possibly time-varying matrices K and D and the position/velocity reference $\xi_{\boldsymbol{d}}$.

5.4.2 Risk-Sensitive Motion Control

Manipulation tasks usually comprise different phases depending on the existence of contact with the environment. During pure unconstrained motion phases no contact forces are present and tracking the desired motion trajectory is the only control goal. However, when forces must be exerted in contact with the environment, their role is typically critical for the task success and need to be prioritized over following the motion trajectory. Let us consider the grasping of an object as an exemplary task: during the grasping phase forces are crucial and tracking the desired motion becomes a secondary goal. If the object to be grasped is placed in a slightly different position than the expected one (assuming no visual feedback is available), the only way to successfully perform the grasping task is to rely on force feedback, making sure that the motion tracking does not disturb.

From a control point of view, a model-based motion control scheme can interpret the activity of the force control as an error in its task model. If the motion-based model was ideal, i.e. the contact location and the environment impedance exactly the same as during the recording of the task, there would be no need for a force controller (except for contact with rigid environments). However, force tracking is necessary when such an exact model is not available. Corrections coming from the desired force tracking are potentially produced by environmental variability/uncertainty in contact location and/or changed environment impedance during the reproduction of the task. As the expected force control goal is given by an uncertain normally distributed trajectory, from (5.45), we model the corresponding force input as an uncertain input $\boldsymbol{u}_f \approx \boldsymbol{\epsilon}$ in the plant dynamics for the motion control

$$M_r\ddot{\boldsymbol{x}} + D_r\dot{\boldsymbol{x}} = \boldsymbol{u_p} + \boldsymbol{\epsilon} \ . \tag{5.47}$$

We assume that $\boldsymbol{\epsilon}$ is normally distributed, i.e. $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}|\boldsymbol{u}_0, \boldsymbol{\Sigma}_u)$. It represents potential errors of the motion task model due to environmental variability reflected in deviations from the expected force trajectory. A priori, these are given by previously observed statistical variability, encoded in the *expected* force trajectory variance yielding $\boldsymbol{\epsilon_e} \sim \mathcal{N}(\boldsymbol{\epsilon}|\boldsymbol{0}, \hat{\boldsymbol{\Sigma}}_f)$. Additionally, online deviations are also considered by observing the *current* force control input $\boldsymbol{u_f}$. These are characterized by $\boldsymbol{\epsilon_c} \sim \mathcal{N}(\boldsymbol{\epsilon}|\boldsymbol{u}_0, \boldsymbol{\Sigma}_{u_f})$, where $\boldsymbol{u_0}$ and $\boldsymbol{\Sigma}_{u_f}$ are calculated online as the average first and second order moments of $\boldsymbol{u}_f$ of a time window of W s. The overall variability is given by $\boldsymbol{\epsilon} = \max(\boldsymbol{\epsilon}_e, \boldsymbol{\epsilon}_c)$, computed as in Section 5.3.3.

Due to the discrete time nature of the implementation, we discretize the system from (5.47) with a sampling time interval Δt yielding a discretized plant dynamics in the form $\xi_{k+1} = A\xi_k + B\boldsymbol{u}_{p,k}$ given by

$$\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \dot{\boldsymbol{x}}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 - \boldsymbol{M}_r^{-1}\boldsymbol{D}_r\Delta t \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \dot{\boldsymbol{x}}_k \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \boldsymbol{M}_r^{-1}\Delta t \end{bmatrix} (\boldsymbol{u}_{p,k} + \boldsymbol{\epsilon}_k) \tag{5.48}$$

where $\boldsymbol{x}_k$, $\dot{\boldsymbol{x}}_k$ and $\boldsymbol{u}_{p,k} \in \mathbb{R}^3$ are the discrete time position, velocity and motion control input at time k in Cartesian space.

Given plant dynamics (5.48) and a desired trajectory to follow $\{\hat{\mu}_\xi, \hat{\Sigma}_\xi\}$, the aim of the controller is to generate the corresponding control input that tracks the given trajectory considering potential errors.

Remark 5.4. If the behavior model is constantly updated in an online fashion, e.g. the model presented in Section 3.3, the expected variability $\boldsymbol{\Sigma}_{\boldsymbol{u}}$ does not requires the empirical estimation given by the first and second order moment of $\boldsymbol{u}_f$.

Reference Adaptation

The motion control tracks the trajectory given by $\{\hat{\boldsymbol{\mu}}_\xi, \hat{\boldsymbol{\Sigma}}_\xi\}$. The desired reference $\xi_{\boldsymbol{d}}$ is then given by the mean of the demonstrated trajectories

$$\mathrm{E}[\xi_{\boldsymbol{d},k}] = \hat{\boldsymbol{\mu}}_{\xi,k}. \tag{5.49}$$

However, in case of diverging forces, the desired motion reference must be adapted to accommodate for potential mismatches in the expected motion trajectory.

Considering the bias of uncertain input $\boldsymbol{\epsilon}_k$, dynamics (5.48) are expressed as

$$\xi_{k+1} = A\xi_k + B(\boldsymbol{u}_{\boldsymbol{p},k} + \boldsymbol{u}_{\boldsymbol{0},k} + \boldsymbol{\epsilon}_k) \ . \tag{5.50}$$

with unbiased input $\boldsymbol{\epsilon}_k = \mathcal{N}(\boldsymbol{\epsilon}|\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{u},k})$. The reference adaptation $\xi_{\boldsymbol{a}}$ is calculated as the motion difference produced by the bias $\boldsymbol{u}_{\boldsymbol{0}}$ applying the system dynamics

$$\xi_{\boldsymbol{a},k+1} = A\xi_{\boldsymbol{a},k} + B\boldsymbol{u}_{\boldsymbol{0},k} \ .$$

If we add it to the desired tracking reference from (5.49), the resulting desired trajectory is adapted to the observed force controller divergence as

$$\xi_{\boldsymbol{d},k} = \hat{\boldsymbol{\mu}}_{\xi,k} + \xi_{\boldsymbol{a},k} \ . \tag{5.51}$$

With this model, dynamics (5.50) take again the same form as in (5.48) with unbiased input $\boldsymbol{\epsilon}_k$, as the bias's effect is taken into account in the considered reference.

Risk-Sensitive Optimization

From (5.48) and the desired reference (5.51) the computation of the motion control input $\boldsymbol{u}_{\boldsymbol{p}}$ can be formulated from an optimality point of view as the minimization of the distance to the desired trajectory. Due to the continuous reestimation of $\boldsymbol{\epsilon}$, an MPC scheme must be adopted and the optimization must be constantly recalculated as the problem parameters change. A quadratic cost function at sample time k for this problem takes the form

$$J_k = \sum_{i=k}^{k+T} ||(\xi_{\boldsymbol{d},i} - \xi_i)||_Q^2 + ||\boldsymbol{u}_{\boldsymbol{p},i}||_R^2 \ , \tag{5.52}$$

where T is the time horizon and Q and R are weighting factors that allow a trade-off between control cost and tracking error minimization. Similar to Section 5.3.2, weighting factors Q can be also chosen proportional to the inverse of the expected motion covariance $\hat{\boldsymbol{\Sigma}}_\xi$ in order to account for potential task constraints encoded into the variability between the learned trials.

The minimization of the expected cost $\mathbb{E}[J_k]$ for dynamics (5.48) leads to a solution providing optimal tracking. However, the influence of $\boldsymbol{\epsilon}$ is ignored, i.e. uncertainty due to environmental variability does not influence the motion control. In contrast, a risk-sensitive optimization directly considers the uncertainty in the dynamics, adapting the manipulator compliance depending on a risk-sensitivity parameter θ. In this case the cost function takes the form

$$\gamma(\theta) = -2\theta^{-1} \ln \mathbb{E}[\exp^{-\frac{1}{2}\theta J_k}] \,.$$

The exponential form of this cost function makes the optimization sensitive not only to the expected cost $\mathbb{E}[J_k]$ but also to higher order moments of it, which are directly related to uncertain input $\boldsymbol{\epsilon}$. If $\theta = 0$ the controller is risk-neutral and $\boldsymbol{\epsilon}$ has no influence on the resulting gain. For $\theta < 0$ and $\theta > 0$ the controller becomes risk-averse and risk-seeking, respectively.

The solution to this problem is detailed in Chapter 4. The resulting feedback term can be written in the form

$$\boldsymbol{u}_{\boldsymbol{p},k} = -V_i(\xi_{\boldsymbol{d},i} - \xi_i) \,, \tag{5.53}$$

where V_i is the feedback matrix given by a modified form of the Ricatti recursion [172]

$$V_i = -R^{-1}B'(BR^{-1}B' + \theta\boldsymbol{\Sigma}_{\boldsymbol{u},k} + \Pi_{i+1}^{-1})^{-1}A \,,$$

and

$$\Pi_i = Q_i + A'(BR^{-1}B' + \theta\boldsymbol{\Sigma}_{\boldsymbol{u},k} + \Pi_{i+1}^{-1})^{-1}A \,,$$

with $\Pi_T = Q_T$. The resulting feedback gain V_i represents both the damping D and stiffness K parameters from (5.46) as

$$V_i = \begin{pmatrix} 0 & 0 \\ K_i & D_i \end{pmatrix}.$$

Note that, due to the MPC scheme, while the recursion for the optimization is calculated for a time horizon T, the only feedback matrix applied in the motion control from (5.53) is the one for the simulated step $i = k$.

As a result, in the risk-averse case, $\theta < 0$, the feedback gain becomes higher interpreting the noise in a pessimist manner as it if was directing the state in the wrong direction. For the risk-seeking case, $\theta > 0$, the feedback gain becomes lower adopting an optimist attitude as it assumes that the noise is already doing part of the job and therefore directing the state in the right direction. Applied to our scenario, a risk-seeking policy arises as the most intuitive solution. In this case, the stiffness decreases when the force control needs to correct the desired force trajectory in order to avoid disturbances. The risk-averse case is a suitable option for pure motion tasks as the stiffness rises under force control corrections, increasing the tracking precision. Note that while the presented solution only considers linear dynamics, the approach is also applicable to nonlinear dynamics by means of the methods explained in Chapter 4.

5.4.3 Evaluation

In order to evaluate the performance and demonstrate the applicability of the proposed control scheme, a full-scale experiment is conducted in our laboratory. Using a robotic manipulator two different tasks involving contact with the environment are performed for four

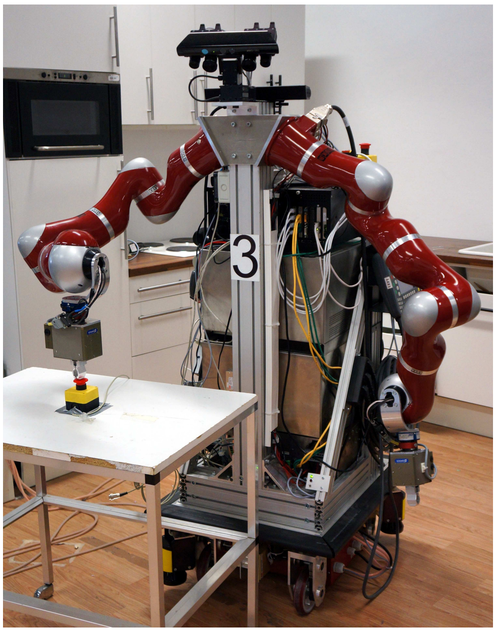


Figure 5.19: Task consisting of pushing a button followed by pointing out text written on a whiteboard.

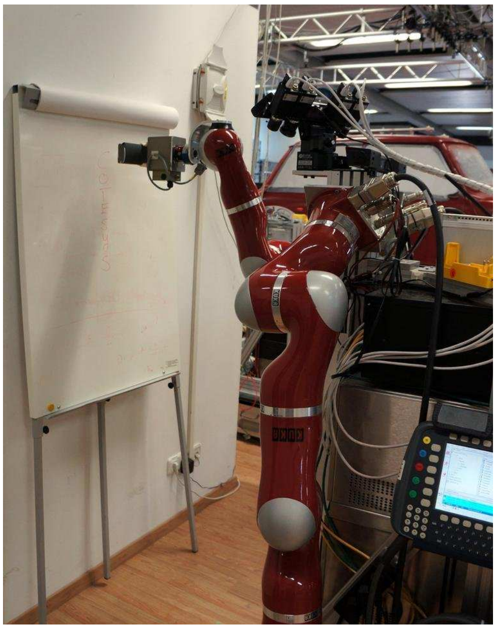


Figure 5.20: Task consisting of cleaning a whiteboard with an eraser grasped with a two-finger gripper.

different motion control strategies. The presented results show the advantages of the proposed approach testing the adaptation capabilities to different environmental variations.

Experimental Setup

The robotic platform employed for this experiment described in Appendix A.3 is shown in Fig. 5.19. In our implementation rotational motions are ignored so that the system dynamics remain decoupled and linear for each Cartesian DoF. To acquire a generalized model of a task given a set of exemplary demonstrations, a probabilistic task model λ given by a time-based HMM is acquired by the methods described in Section 3.1. For evaluating combinations of risk-sensitivity and motion reference, we tested four different motion controllers:

(a) Risk-neutral optimization, i.e. $\theta = 0$ and motion reference (5.49).

(b) Risk-seeking optimization, i.e. $\theta = 10^{-5}$ and motion reference (5.49).

(c) Risk-neutral optimization, i.e. $\theta = 0$ and motion reference (5.51).

(d) Risk-seeking optimization, i.e. $\theta = 10^{-5}$ and motion reference (5.51).

Note that controllers a) and c) are not influenced by the variability of the force control $\boldsymbol{\Sigma}_u$ in contrast to b) and d), that adopt a risk-seeking optimization. Similarly, controllers b) and d) adapt their motion reference depending on the bias $\boldsymbol{u}_0$ while a) and c) do not modify the learned references.

Task	Pushing Button and Pointing			Erasing Whiteboard	
Controller	Initial Distance	Button	Pointing Distance	Initial Distance	Whiteboard
(a)	0 cm	○	0 cm	- 5 cm	○
	5 cm	×	0 cm	0 cm	●
	10 cm	×	0 cm	+5 cm	●
(b)	0 cm	○	0 cm	- 5 cm	●
	5 cm	○	0 cm	0 cm	●
	10 cm	○	0 cm	+5 cm	●
(c)	0 cm	●	8.5 cm	- 5 cm	○
	5 cm	●	9.5 cm	0 cm	○
	10 cm	●	7.8 cm	+5 cm	○
(d)	0 cm	●	8.5 cm	- 5 cm	○
	5 cm	●	9.5 cm	0 cm	○
	10 cm	●	7.8 cm	+5 cm	○

Tabular 5.3: Experimental results for pushing button and pointing task and for whiteboard erasing task. ● success, ○ partial success, × failure

For our experiments, the parameters of Eq. (5.45) are

$$
\begin{aligned}
M_r &= \text{diag}\,(10\text{ kg}, 10\text{ kg}, 10\text{ kg}\ )\\
D_r &= \text{diag}\,(80\text{ Ns/m}, 80\text{ Ns/m}, 80\text{ Ns/m}\ )
\end{aligned}
$$

and the window size W from (5.48) is set to 0.3s. For simplicity of presentation, the tracking error precision Q in (5.26) is chosen such as errors with respect to the expected velocity trajectory are ignored and therefore only position errors are considered resulting in a stiffness adaptation, i.e. $D = 0$ in (5.46).

Experimental Design

Two tasks are considered for the reproduction performance evaluation. A model of each task is acquired after teaching 12 different demonstrations of each task keeping the same environmental conditions.

The first task, shown in Fig. 5.19, consists of pushing a button followed by pointing out text written on a whiteboard. Three different environmental conditions are tested. The button height is decreased with respect to the teaching environmental conditions for 0, 5 and 10 cm

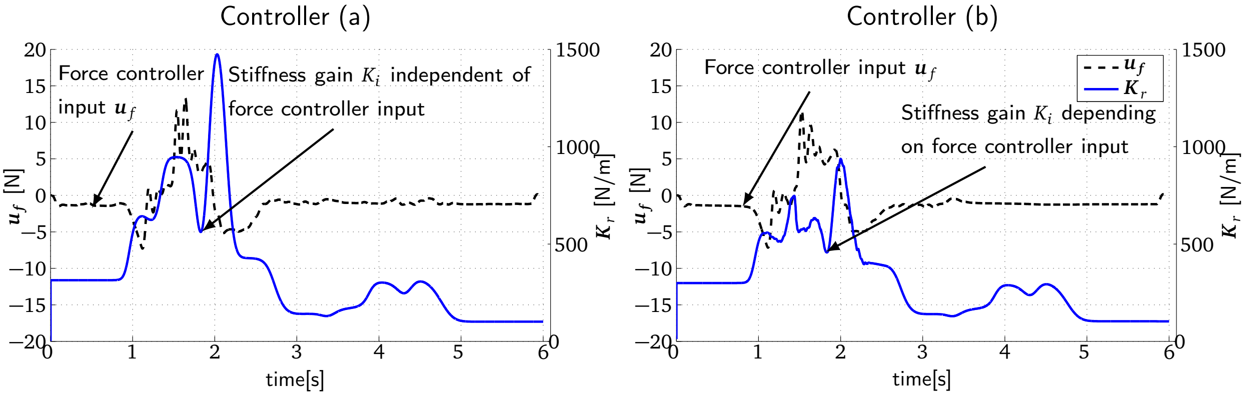


Figure 5.21: Control gains independent of the force error (a) and dependent of the force error (c) together with the force controller input during the button pushing and pointing out task for 10cm.

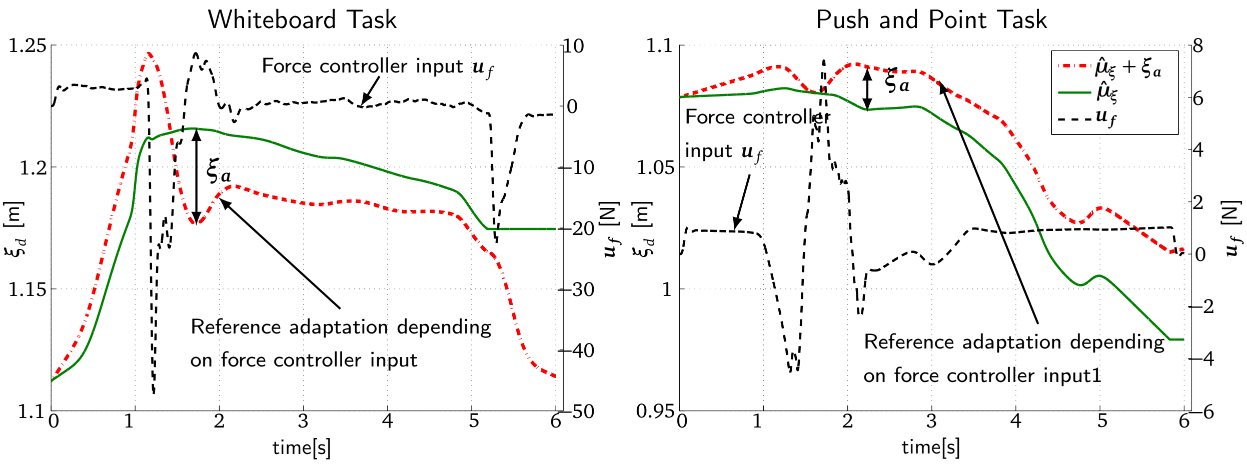


Figure 5.22: Force control input, desired trajectory and adapted desired trajectory with motion reference adaptation depending on force controller input (c) for both tasks for 5cm initial distance.

respectively. A task execution is considered successful when the button is correctly pushed and the predefined place at the board is exactly pointed out. The button pressing procedure is considered a success when the button is correctly pushed and clicked, a half success when contact is partially pushed but did not clicked and other cases are considered as failure.

The second task, shown in Fig. 5.20, consist of cleaning a whiteboard with an eraser which is assumed to be already grasped. Three different environmental conditions are tested, including an initial end-effector position differing in −5, 0 and 5 cm to the teaching environmental conditions. The task is considered successfully executed when the text written on the whiteboard is erased. A half success is considered when the text is only partially erased.

Results: Pushing and pointing out

As shown in Fig. 5.21 for the task reproduction placed 10 cm lower with respect to the initial conditions, the resulting stiffness depicted by the solid blue line is adapted depending on the force control input represented by the black dashed line for the risk-seeking controllers, i.e. (b). The variable stiffness profile from (a) is only a result from the optimization and are not affected by the force controller activity. As no reference adaptation is considered in controller (b), the stiffness decreases more with respect to (a) as it needs a stronger correction from the force control. As a result of the adaptive stiffness with respect to force errors produced by the risk-seeking optimization, the force controller receives less disturbances from the motion controller and can achieve a better force tracking performance which translates in a better task performance: as shown in Table 5.3, the risk-seeking controller (b) performs better than the risk-neutral one (a) when pushing the button.

The reference adaptation for controller (c) and (d) for the 5 cm case is shown in Fig. 5.22. The adapted reference depicted by the dashed-dotted red line deviates from the task model's desired trajectory represented by the solid green line accommodating for the force controller input represented by the black dashed line. As a consequence, a better task performance for the button pressing phase is achieved as shown in Table 5.3 for controllers (c) and (d). However, a drift from the original reference is present at the end of the task execution producing an undesired tracking error effect for the pointing out motion. This effect is not present in the other two controllers that do not adapt their references.

Results: Erasing

The results for the erasing task are shown in Table 5.3. For the two controllers not adapting their motion reference, the risk-seeking controller (b) arises again as a better alternative as it correctly erases the whiteboard in all conditions, in contrast to the risk-neutral case (a) that ignores the force control input. While this task suggests that an adaptation of the reference of the motion controller is beneficial if an environmental offset is present, the results show worse performance for the controllers adapting their reference, i.e. (c) and (d). As shown by Fig. 5.22, the adapted reference for controller (c) depicted by the dashed-dotted red line overcompensates the force controller contribution. This undesired effect may be caused by a too fast accommodation of the observed bias, suggesting that an increased window size for process noise estimation could improve the controller performance by capturing only very clear bias trends.

In summary, the risk-seeking optimization is as a suitable option as it provides variable stiffness depending on environmental force deviations. As a result, increased generalization capabilities in varying environmental conditions is achieved. Similarly, the reference adaptation also improves task success accommodating to diverging forces by adapting the motion reference. Depending on the task, the resulting drift of the reference trajectory w.r.t the desired one increases or decreases the generalization capabilities of the control scheme. This issue requires further investigation. The addition of a forgetting factor to the reference adaptation term is a possible alternative to be addressed in future works.

5.5 Assistance with Multiple Models

As exposed in Section 2.1, the characterization of behavior models for anticipatory physical assistance is very diverse, comprising analytical, learning and planning approaches. Depending on the a priori available environmental and task information, the predictive performance of the different models may vary significantly. When several models are available, the selection of the most promising one arises as an additional control issue. As an example, consider a scenario where various models are available but they only represent partially the task to be fulfilled. The necessity of a dynamic *selection* of the most appropriate strategy between all available models becomes evident. Additionally, when human behavior diverges from any of the possible task specific plans known to the robot, a suitable anticipation strategy remains an open issue. This section studies solutions for these two problems. First, a wrench-based assistance scheme relying entirely on sensed forces is presented, becoming a suitable alternative when no specific task model coincides with human intentions. Second, a method to select the most promising strategy following a short-term retrospective evaluation methodology is proposed. By estimating the applicability of both a model-based and the presented wrench-based control schemes based on recent prediction performance, a strategy selection scheme decides to either retain the current assistance model or switch to a recently more successful strategy. A user study in a 2D virtual scene shows the applicability and the benefits the proposed approach in terms of human effort minimization.

The remainder of this section is organized as follows: after we confine the problem in Section 5.5.1, Section 4.1.1 describes the overall approach. Detailed explanations of our proposed dynamic strategy selection scheme is found in Section 5.5.3. The experimental evaluation with human users is described in Section 5.5.4.

5.5.1 Problem Formulation

The task considered in this section consists of the physically coupled movement of a human and a robot from an initial to a final configuration. However, instead of assuming cooperation towards a common goal known to both partners [25, 181], we consider the possibility that the human diverges from the robot's assumed final configuration or path to the goal. No information on the desired trajectory of the human or the robot is provided to the partner other than through haptic interaction. As in Section 5.3, we assume the robot renders dynamics

$$M_r\ddot{\boldsymbol{x}} + D_r\dot{\boldsymbol{x}} = \boldsymbol{u_h} + \boldsymbol{u}_r \ , \tag{5.54}$$

with inertia M_r and viscous friction D_r, $\boldsymbol{u_h}$ the applied force by the human, $\boldsymbol{u}_r$ the assistive control input of the robot, and $\xi = \left[\boldsymbol{x}^{\mathrm{T}}\ \dot{\boldsymbol{x}}^{\mathrm{T}}\right]^{\mathrm{T}}$ the state of the system, where $\boldsymbol{x}$ denotes the shared object's configuration.

The anticipatory robot input $\boldsymbol{u}_r$ is computed based on a behavior model λ in terms of the human desired path. In contrast to Section 5.3, we now assume that λ can be close to the human intentions, incomplete, partially or completely wrong, i.e. we consider the possibility of divergence from the human side. The goal of this section is the synthesis of the anticipatory robot's proactive contribution $\boldsymbol{u}_r$ taking into account that the human may aim for a different goal or path than the one represented in the behavior model λ.

5.5.2 Approach

From the problem setting, two different cases can be intuitively identified depending on the task model λ's similarity to the human intentions: a) when λ coincides with the human

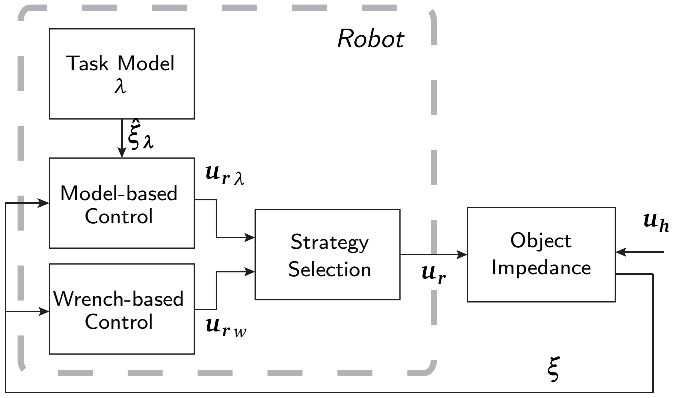


Figure 5.23: General control scheme. The strategy selection switches between a model-based control scheme and a wrench-based control scheme depending on their performance in terms of human effort minimization.

plan, or, in contrast b) the human intended goal or path differs from λ. This binary case separation suggests the situation-dependent application of two different control schemes. On one side, an *model-based* control scheme relying on the task model's predictions performs satisfactorily when both partners have a similar plan (case a)). On the other, interpreting the human force as an indicator for the desired movement direction, a *wrench-based* control scheme based on the currently observed human control input is preferred when the human goal is unknown (case b)). Note that an intermediate strategy between both cases is not considered as a useful solution since merging different goals can lead to undesired or even potentially unsafe configurations.

We consequently propose a strategy selection approach that switches between the model-based and the wrench-based control based on their performance in terms of human effort minimization, as shown in Fig. 5.23. The resulting scheme supports the human partner in case of correct anticipation of the human higher-level intention and also in unexpected situations. Each control scheme relies on a different human desired state trajectory ξ_d. On one side a model-based assistance relies on the task model's λ predictions and assumes that the deviations coming from the human side are not diverging from λ's predictions, i.e. the human goal coincides with the model's description. On the other side, a wrench-based control assumes that the human diverges from any known task models discarding their predictions. The human desired state trajectory is then estimated based only on the human control input.

Model-based control scheme

The model-based scheme assumes that the the behavior represented by the task model λ coincides with the human motion intention and therefore follows predictions given by λ, i.e. $\xi_d = \hat{\xi}_\lambda$. To compute the robot anticipatory input $u_{r\lambda}$ we rely on the optimal control method described in Section 5.3.

Wrench-based control scheme

The wrench-based control scheme assumes different intended goals between the human and task model λ. During physical interaction, this can be detected based on the measured interaction force, where a continuous level of human force input indicates a disagreement to the current motion. In order to represent this divergence, predictions rely on the observed human wrench $\boldsymbol{u_h}$. Applying discretized system dynamics (5.36) the wrench-based prediction of the desired state trajectory $\xi_d = \hat{\xi}_c$ is given by

$$\hat{\xi}_{w,k+1} = A\xi_k + B\boldsymbol{u}_{\boldsymbol{h},k}. \tag{5.55}$$

The wrench-based control considers the observed divergence as the human desired trajectory, i.e. $\xi_d = \hat{\xi}_w$, tracked by means of an optimal regulator as described in Section 5.3.3 and yielding $\boldsymbol{u}_{r_w}$.

5.5.3 Dynamic Strategy Selection

The principle that governs the control design in this work is the minimization of the human force contribution during interaction, which is related to motion prediction accuracy. A higher prediction accuracy results in a diminished required and applied human force contribution, $\boldsymbol{u_h}$, because it allows the robot to contribute a larger load share $\boldsymbol{u_r}$ of the required total force input $\boldsymbol{u}$ to achieve the desired state trajectory ξ_d. This motivates us to develop a strategy selector that aims at optimizing prediction accuracy. By evaluating the models in their short-term retrospective behavior over the window length H, we obtain an estimate for the instantaneous future prediction error. The measure α_i used here to evaluate the prediction accuracy for the control scheme $i \in \{\lambda, w\}$ is defined in the mean squared error sense, i.e.

$$\alpha_{i,k} = \frac{1}{H} \sum_{j=k-H+1}^{k} ||\xi_j - \hat{\xi}_{i,j}||^2, \tag{5.56}$$

where both the model-based and the wrench-based control strategy simultaneously provide predictions $\hat{\xi}_{\lambda/w}$ of the human desired trajectory. Assuming the prediction error to be stationary within the interval H, α_{i_k} defined in (5.56) serves as a good estimate for the mean squared error of the future prediction based on past observations until time step k, i.e.

$$\mathbb{E}\left[||\xi_{k+1} - \hat{\xi}_{i,k+1}||^2 | k\right] \approx \alpha_{i,k}, \quad i \in \{\lambda, w\}, \tag{5.57}$$

where $\mathbb{E}[x|k]$ is the conditional expectation of x conditioned on known data until k. Due to the fact that higher prediction accuracy reduces the human force $\mathbb{E}[\boldsymbol{u_h}]$ and having the estimates for the current prediction accuracy given by (5.57), we select the applied control scheme $\boldsymbol{u}_{r,i}$ used in the next time step $k+1$ according to the measure $\alpha_{i,k}$ by

$$i = \arg\min_{j \in \{\lambda, w\}} \alpha_{j,k}.$$

When choosing a value for the window length H, it can be seen that a trade-off needs to be found between the sensitivity to noise and the ability for fast adaptation to changes of the human behavior. A small value for the window length leads to a greater sensitivity with respect to noise, whereas a large value of H may violate the stationarity assumption needed to assert (5.57).

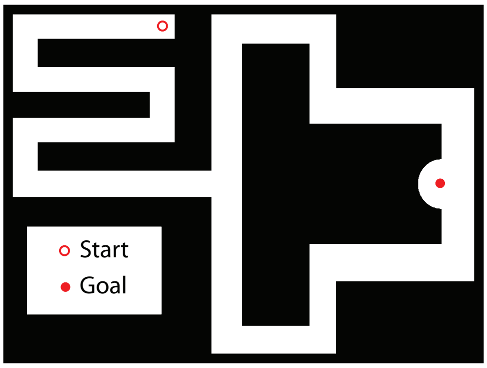


Figure 5.24: Example task: Moving a point mass object from start to goal position through a maze

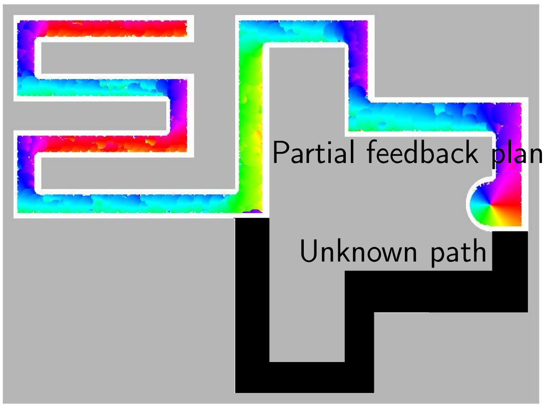


Figure 5.25: Feedback motion plan computed with SNG method. The hue indicates the direction towards the goal.

Finally, it should be noted that the proposed selection strategy can also be applied to the more general case for N control schemes with $1 \leq i \leq N$, when several model-based schemes are available.

5.5.4 Evaluation

The proposed control and strategy selection schemes are evaluated in a human-robot cooperative setup in virtual reality. A user study evaluates different assistive control strategies for a simple task consisting of jointly carrying a virtual object from an initial position until a final goal through different possible paths in a 2D maze.

Experimental Setup

The virtual reality interface used for this experiment is described in Appendix A.1. The displayed task to transport a virtual object is visually represented by a filled red circle and the target position in the upper left corner of the maze (blue dot) depicted in Fig. 5.24. Collisions with the virtual walls should be avoided. Table 5.4 exhibits the constants used to parameterize the experiment.

Constant	Equation	Value
Simulated object mass m	(5.54)	$100\,\mathrm{kg} \cdot I_2$
Simulated viscous friction ν	(5.54)	$400\,\frac{\mathrm{Ns}}{\mathrm{m}} \cdot I_2$
Optimization's time horizon T	(5.26)	$0.1\,\mathrm{s}$
Tracking error weighting Q	(5.26)	$10^8 I_4$
Control cost weighting R	(5.26)	I_4
Filter window length H	(5.56)	$25\,\mathrm{ms}$

Tabular 5.4: Control parameters used in 2-DoF experiment

Quantitative Measures

We evaluate the following criteria in order to rate the performance of the proposed approaches:

- Mean completion time T_{mean} is a task-related performance measure and serves as an indicator of the increase of efficiency to accomplish a task through interaction.
- Mean absolute human force input $\frac{1}{t}\int_0^t \|\boldsymbol{u_h}\|\,d\tau$.
- Mean energy contributed by the human as measure of effort $E = \int_0^t \boldsymbol{u_h}^T \dot{\boldsymbol{x}}\,d\tau$ is an indicator for the capability of the robotic assistant to take over the overall work load to complete the task.
- Number of collisions with the virtual environment serves as a measure for safety and controllability during task execution.

Experimental Design

We conducted a small pilot study in the presented VR scenario to evaluate the performance of our proposed approach. Twelve non-paid participants (age mean: 27.5, std: 2.7) were instructed to move a virtual object through the simple maze used above from a starting configuration to a final configuration through the scene without colliding with the virtual obstacles visually and haptically displayed.

A total of 6 different conditions were evaluated depending on the controllers used and the paths chosen to solve the maze

(i) No active assistance, i.e. $\boldsymbol{u_r} = 0$ following always the same path.

(ii) Wrench-based control scheme following always the same path.

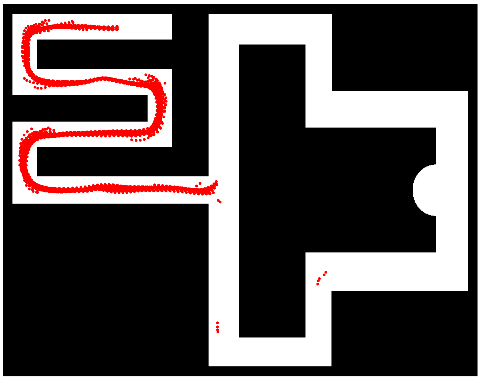


Figure 5.26: Spatial distribution of the strategy selection scheme for all participants for the partially known model condition (v): red dots indicate where the model-based strategy was selected.

(iii) Model-based control scheme with risk-seeking optimization, i.e. $\theta = 10^{-5}$, following always the same path.

(iv) Dynamic strategy selection scheme between controllers (ii) and (iii), following always the same path.

(v) Dynamic strategy selection scheme between controllers (ii) and (iii), following the opposite path with incomplete task model λ.

Each participant repeated each condition 5 times. The underlying task model λ is given by a feedback motion plan towards the goal.

A feedback motion planning algorithm generates a feedback function $\mathscr{K}(\boldsymbol{x})$ for all accessible positions $\boldsymbol{x}$. The sampling-based neighborhood graph (SNG) [182] is a very comprehensible method, efficiently covering the accessible region with the feedback function that indicates the direction towards the goal. Therefore, the entire configuration space is randomly clustered into overlapping circles (or hyperballs in higher dimensions), and Dijkstra's algorithm is applied to plan on the connected graph from the circle containing the initial configuration to the circle containing the final configuration. Finding the shortest path within circles is straightforward. The feedback motion plan for our experiment is depicted in Fig. 5.25. The hue of the colors encodes the direction to the goal. Note, that the lower path is excluded in the feedback plan to render an incomplete task model λ.

Experimental Results

The quantitative results of the experiment are depicted in Fig. 5.28 for all considered conditions. The condition of no assistance (i) has longer completion times and significantly higher mean force and mean absolute power while the mean number of collisions remains on average. While the model-based condition (iii) and the complete strategy selection condition (iv) have similar mean completion times, the mean force and the mean absolute power are higher for condition (iv) as the trial numbers increase, but the mean number of collisions is lower. The model-based condition (iii) has a fixed model and can not adapt its behavior between

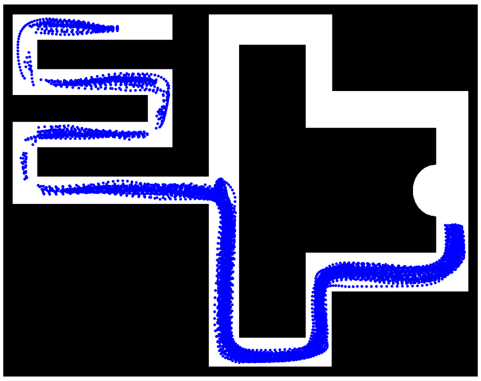


Figure 5.27: Spatial distribution of the strategy selection scheme for all participants for the partially known model condition (v): blue dots indicate where the wrench-based strategy was selected.

trials while the complete strategy selection condition (iv) switches from the model-based to the wrench-based scheme in order to accommodate for higher velocities. Although the path to follow is only partially known, it is remarkable that the incomplete strategy selection condition (v) performs always close to the complete strategy selection condition (iv) for all quantitative measures. At the same time and w.r.t. the wrench-based scheme (iii), the mean completion time is shorter and the the effort needed lower.

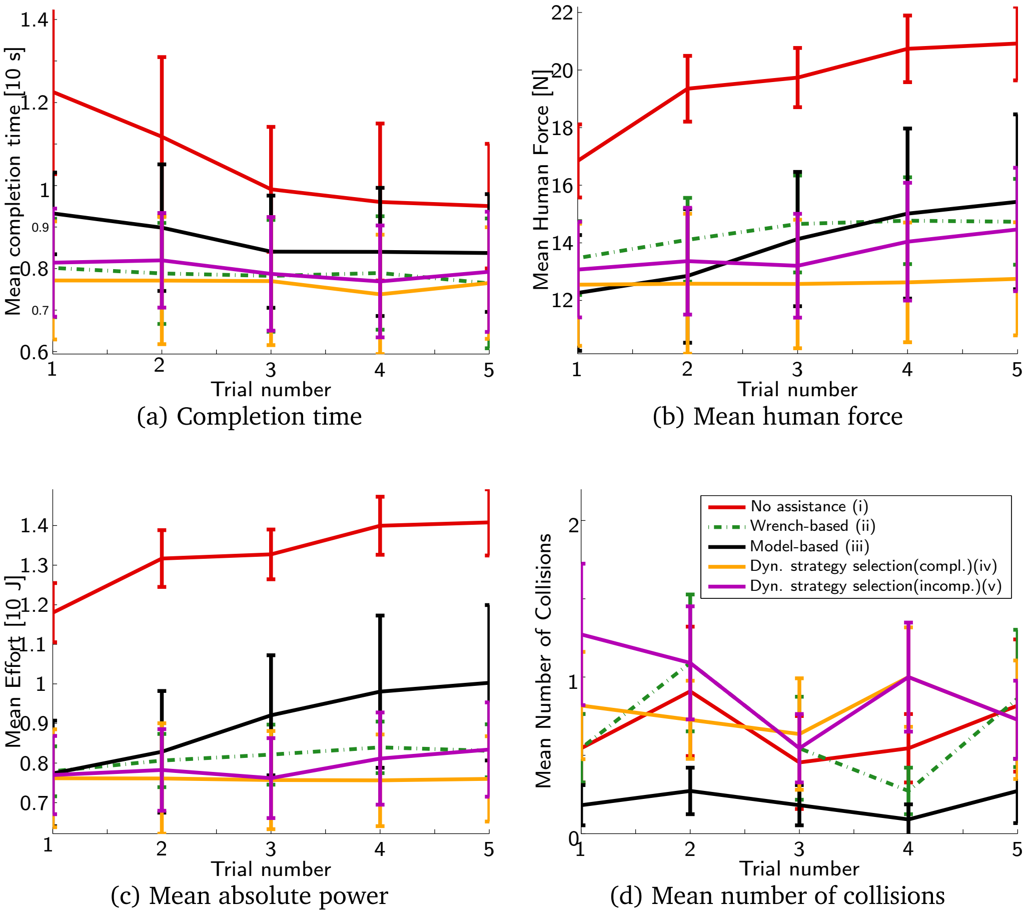
The spatial distribution of the dynamic strategy selection results for the incomplete model case (v) is presented in Fig. 5.27 and Fig. 5.26. Figure 5.27 shows that the wrench-based control scheme is primarily selected in the unknown area of the map, where the model-based approach has no prior knowledge, and during motion along straight lines, where the model-based approach does not adapt its velocity profile to different execution speeds. The model-based strategy is primarily selected during corner turns, where predictions purely based on the human control input are not successful, see Fig. 5.26. It is also remarkable that any potentially undesired switching effect (jerk) in the system dynamics was not noticeable for the participants due to the rendered compliance in the linear-actuated device.

The results indicate that our proposed strategy selection scheme combines the advantages of model-based control in case of a suitable task model and the advantages of our proposed wrench-based control scheme when the task model is far from human motion intention. Model-based control leads to higher execution speed as the trajectory prediction and assistance is accurate not only along straight lines but also during turns. While achieving a similar level of velocity, the proposed dynamic strategy selection scheme requires a significantly lower effort induced by the human partner, an indicator for the efficiency of the strategy.

5.6 Summary

This chapter addressed relevant challenges in the design of model-based anticipatory physical assistants with special emphasis on learned models.

The benefits and disadvantages of anticipation based on models learned during interaction



(a) Completion time

(b) Mean human force

(c) Mean absolute power

(d) Mean number of collisions

Figure 5.28: Mean and standard deviation evolution of quantitative parameters over trials.

were presented in Section 5.1. Although effective human behavior prediction significantly reduces human effort, mispredictions imply disturbing counteracting wrenches. These insights showed the necessity of a deeper understanding of causes and consequences of counteracting wrenches.

Section 5.2 studied internal and effective wrench components arising between goal-oriented agents. This analysis follows from understanding possibly non-uniform wrench allocation policies and load shares. Internal wrench indicate errors in terms of differing desired acceleration directions thereby showing its relevance for efficient anticipatory control designs.

Besides compensating for undesired wrench components, the assistance control scheme introduced in Section 5.3 for navigation tasks also adapt its anticipatory input depending on prediction uncertainty. Formulated as an optimal control problem targeting human force minimization, prediction uncertainty from both the expected trajectory and expected disagreements become decisive components of robot assistance. The Mahalanobis distance modifies the robotic contribution under high prediction uncertainty. In addition, a risk-sensitive optimization governing the robot's behavior synthesizes a spectrum of attitudes in case of disagreements, from recessive to dominant. The proposed approach was objectively and subjectively evaluated in a psychological study, indicating superior performance in terms of perceived helpfulness and human effort minimization.

When tasks are not limited to navigation but also comprise contacts with the environment, the regulation of environmental wrenches is also necessary. Section 5.4 presented an optimal control scheme for manipulation where robot motion takes uncertain environmental force deviations into account. A risk-sensitive optimization provides variable compliance improving generalization capabilities. Additionally, the motion reference adapts to unexpected environmental force divergence yielding higher adaptability in uncertain and variable environments, as validated in experiments with a robotic manipulator.

Envisaging robotic partners fulfilling multiple tasks in cooperation with a human, Section 5.5 introduced a dynamic strategy selection method that decides either to retain a currently selected model or switch to another one depending on recent prediction performance. In addition, a predictive model relying on observed human wrench was also presented, becoming a suitable alternative when no available model coincides with human intentions. An experiment with human users highlighted the superior performance of a dynamic selection strategy when multiple models are available.

6

Conclusions and Outlook

The capability to physically interact and cooperate with humans is an essential requirement for robots sharing their workspace with humans. This scenario encompasses relevant applications such as rehabilitation, mobility aids, elderly care, manufacturing, education or training. A helpful cooperative physical robotic helper is realized by means of suitable control design, requiring: (i) human behavior prediction, (ii) intelligent decision-making and (iii) intuitive reactions to disagreements. This thesis addressed relevant steps towards control algorithms fulfilling such prerequisites.

In order to be able to anticipate human actions, Chapter 3 presented suitable online model learning methods. Due to physical coupling, human models for physical assistance require smooth, fast and conservative predictions. We first studied time-series models of motion and force signals by means of time-based HMMs. Exploiting the explicitly learned time information, time-based HMMs provide smoother predictions in contrast to traditional HMMs. To enable the acquisition of behavior models during interaction, we presented a primitive-based incremental learning framework, successfully validated in an experiment with human users. To represent the correlation of speech commands and environmental variables on motions, we presented parametric time-based HMMs. External parameters modify output densities and transition probabilities of the standard HMM, yielding parametric motion models. By means of the presented parametric HMMs in combination with a natural language model, the influence of environmental properties and natural language descriptions on motion primitives is successfully captured. As a result, the proposed framework delivers motion primitives that adapt to quality descriptors and specific environmental configurations. Although the presented time-series models effectively capture time constraints, their representation of spatial constraints is limited. As an alternative, we studied the applicability of dynamical systems by means of GPs in order to learn observed task dynamics during interaction. We introduced novel impedance-based GP models, where the human is assumed to behave as a mechanical impedance. The unobservable human-desired trajectory is modeled as a latent variable tracked by means of a PD control scheme. The resulting model has desirable properties: in previously unobserved configurations it behaves as a passive mass-damper system. In addition, increased prediction performance is achieved w.r.t naive GP models as validated in the presented experiments.

When robot decisions depend on uncertain human models, the consideration of uncertainty in robot actions becomes instrumental. Chapter 4 presented a novel a systematic approach for the design of uncertainty-dependent decision-makers. We studied the stochastic optimal feedback control problem considering high order cost statistics. In contrast to classical approaches which only evaluate the expected performance, the proposed optimization criterion is given by an arbitrary weighted sum of cost cumulants. In a similar way to statistical moments, high-order cost cumulants capture the influence of uncertainty on the cost. We presented locally optimal solutions to this problem for nonlinear systems with non-quadratic costs. Numerical solutions are computed by approximating the problem around a nominal trajectory, solving the local problem, updating the trajectory and repeating the process until convergence. We showed the potential and the applicability of the proposed approach in various prototypical problems. The effect of multiple sources of uncertainty, present in the robot dynamics and cost parameters, such as a goal or an obstacle position is illustrated in simulations of nonlinear non-quadratic problems.

Besides human behavior models and flexible uncertainty-dependent decision-making methods, the design of intuitive physical assistants requires an adequate understanding of human intentions by means of force signals. In Chapter 5, we first identified internal and effective wrenches arising during interaction by previously analyzing possibly non-uniform load decomposition policies. Based on these findings, we proposed a novel anticipatory control scheme for navigation tasks adapting to prediction uncertainty and disagreements. Formulated as an optimal control problem minimizing human force, prediction uncertainty from both the expected trajectory and disagreement become decisive components of robot assistance. The presented scheme demonstrated superior performance in terms of human effort minimization and perceived helpfulness in an experiment with human users. For tasks requiring contacts with the environment, the regulation of environmental forces becomes a necessary component of the robot control scheme. We presented an optimal motion control scheme that adapts robot motions to model uncertainty and environmental force deviations. Increased robustness and generalization capabilities were achieved in the presented experiments. Envisaging more complex settings with multiple behavior models and tasks, we introduced a dynamic strategy selection method that either retains a currently selected model or switches to another one depending on prediction performance. In addition, a predictive model relying on observed human wrench was also presented, becoming a suitable alternative when no available model coincides with human intentions. The presented user study indicated superior performance of the dynamic selection scheme when multiple models are available in terms of human effort.

6.1 Outlook

This thesis is an endeavor towards intuitive and intelligent physical cooperative partners. Although relevant issues regarding their design were addressed here, the deployment of such systems in daily life scenarios requires further investigation. In the following, we expose interesting open issues related to this problem to be studied in future works.

Interaction Behavior Models

Chapter 3 presented models representing human behavior in terms of motion and wrench signals. However, a more specific and potentially more accurate option considers explicitly the overall interaction behavior dynamics of all agents. Applying the presented GP impedance model to multi-agent systems appears as a compelling alternative. This setting gives rise to models comprising not only motion and resulting wrenches but also internal wrenches and effective load shares.

In addition, learning dynamical systems with GPs entails encoding potentially unstable dynamics, especially when sparse approximations are considered. Studying local and global properties of GP models is a relevant issue that requires further investigation.

Although this thesis limits its scope to time-series and dynamical systems models, other options presented in Chapter 2 should be also studied. We believe that a comprehensive model acquisition architecture must consider behavior modeling options comprising several levels of abstraction. Not only naive descriptors of motion, but also goal-oriented and optimal strategies w.r.t relevant features are needed. Crucial in this context is also a seamless interplay between models. These are necessary steps in order to deliver robots in realistic settings, where multiple complex tasks during long interaction sessions must be fulfilled.

Haptic Negotiation Models

The acquisition of haptic negotiation models is an interesting subproblem of interaction behavior models. The wrench decomposition results from Chapter 5 enable the observation of the joint dynamics of motion, internal and resulting wrenches and effective loads. An insight of the role of each of these signals can be acquired for instance studying human dyads by means of inverse optimal control techniques. We believe that the way each agent reacts to internal wrenches is an important aspect of the negotiation process to be addressed in future works.

Uncertainty-dependent Optimal Control with Multiplicative Noise

The uncertainty-dependent optimal control solutions presented in Chapter 4 comprise problems with uncertainty either in cost parameters or as additive noise in the plant dynamics. However, the problem where uncertainty appears as a factor of states or controls is also a relevant problem. As an example, consider a control problem with learned dynamics, represented for instance by a GP or an HMM. In these cases, predicted dynamics present variance/process noise dependent on inputs, i.e. state and controls. A linear approximation yileds multiplicative noise settings, which appear as a challenging but promising problem with application not only in robotics but also in many other domains such as neuroscience or mathematical finance. Recent results on risk-sensitive control present closed-from solutions for plants with multiplicative noise in continuous time [183] but a solution for discrete-time settings remains an open issue. Additionally, model predictive control schemes enable constant adaptation in dynamic settings and close to real-time computation times [184]. The implementation of these family of controllers based on our uncertainty-dependent approach, as well as belief-space variants [145] are also matter of future research.

Load Share Control

Chapter 5 presented optimal control schemes correcting observed internal wrench components and adapting to model uncertainty. In all presented solutions the robot was always assumed to be accounting for all necessary load. The regulation of load shares was not addressed in this work but appears as an interesting extension. A suitable alternative for approaching this problem is the study of the aforementioned negotiation process in human dyads for later application in human-robot settings. Such models enable first the acquisition of desired load shares and their posterior regulation during interaction.

A

Apparatus

A.1 2 DoF Linear Actuator

The haptic interaface consists of a two degrees-of-freedom (anteroposterior and mediolateral plane of the user standing in front) linear-actuated device (*ThrustTube*) with a free-spinning handle (superoinferior direction of the user) at the grasp point. Each actuator is equipped with a position encoder with precision of 1 μm. Attached to the handle is a 6 DoF force/torque sensor (*JR3*), which measures the human wrench input.

If necessary, a virtual scene is visually represented on a display placed on top of the interface, see Fig. A.1. Virtual obstacles are easily visually and haptically displayed by means of a visuo-haptic rendering framework. For more details, see [185]. Arbitrary two-dimensional scenes are imported from anystandard monochrome pixel or vector graphics file and are automatically rendered as stiff environment. All control algorithms are implemented in *Matlab's Simulink Coder* and executed on *Linux Preempt/RT* at a sampling rate of 1 kHz running on an external PC.

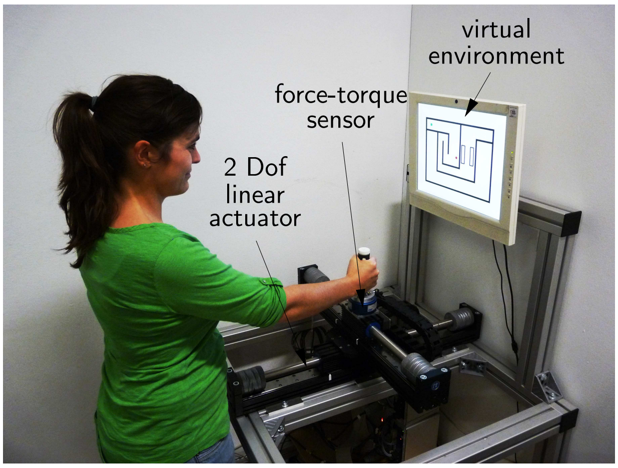


Figure A.1: 2 DoF linear actuator setup with an attached force/torque sensor at the handle and a virtual scene displayed on the monitor.

A.2 Robotic Platform 1

The human-sized robot(see Fig. A.2) stands on a four-wheeled omni-directional mobile platform (f) which offers roughly human-like maneuverability and smooth motion [186]. Two identical anthropomorphic 7-degrees-of-freedom (DoF) arms (c) are front-mounted on the top of the main chassis to provide a human-like working space [187]. Mounted onto a JR3 wrench sensor, the manipulator is equipped with a Schunk PG70 two-finger parallel gripper (a) which allows a tight grasp of the object and force/torque sensing enabling force feedback control. Lithium-ion polymer batteries (e) power the system for long periods without recharging and expands the reachable space. More information on the robot used can be found in [188].

For computational power, the robot carries three PCs (d). The first is a an *Intel Core i7 920* running at 2.66 GHz executing high-level reasoning tasks exploiting the multi-thread *OpenMP* library. The entire manipulator control scheme is implemented in *MATLAB/Simulink* and executed on the *RTAI* using *Matlab's Real-Time Workshop* on a second PC. Low-level control algorithms run at a frequency of 1 kHz. The third PC is identical to the first and ensures real-time control of the mobile platform and a synchronized data management utilizing the *KogmoRTDB* real-time database [189] available at [190] and the *PREEMPT_RT Linux* real-time kernel patch [191].

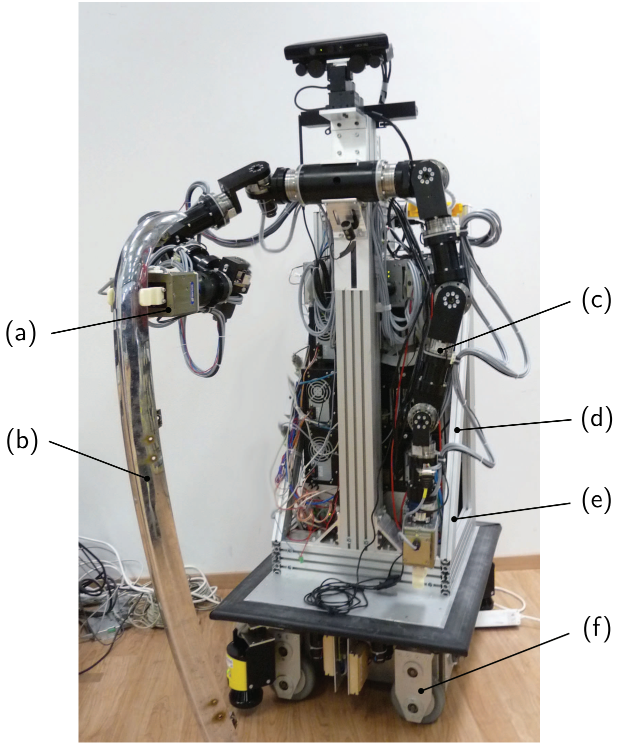


Figure A.2: The human-sized mobile robot with anthropomorphic arms equipped with two-finger grippers and an omni-directional mobile platform.

A.3 Robotic Platform 2

The second human-sized robotic platform shown in Fig. 5.19 consists of a four-wheeled omni-directional mobile platform only used for repositioning the robot and two identical commercially available KUKA LWR (light-weight robot) 4+ manipulators. Desired angular movements for desired movements in task space are computed by an instantaneous inverse kinematics. For measuring resultant forces independent of the configuration and of the human guiding force during kinesthetic teaching of the manipulation task, a 6 DoF force/torque sensor (*JR3*) is attached to the wrist. As end-effector, a Schunk PG70 two-finger parallel gripper is used for simple grasping and object pushing with the fingers. Lithium-ion polymer batteries mounted at the omni-directional platform power the system increasing its autonomy.

The robot carries two KUKA control boxes, one for each manipulator and a single PC for computational power. The PC consists of an *Intel Core i7 920* running at 2.66 GHz executing control schemes, implemented in *MATLAB/Simulink* and executed on the *RTAI* using *Matlab's Real-Time Workshop*. It also ensures real-time control of the mobile platform and a synchronized data management utilizing the *KogmoRTDB* real-time database [189]. In order to operate the KUKA LWRs we use *Fast Research Interface* (FRI), which allows user control and status monitoring of the manipulators based on UDP protocol. This software package is integrated in the real-time robot control framework [192]. All low-level control algorithms runs at a frequency of 1 kHz.

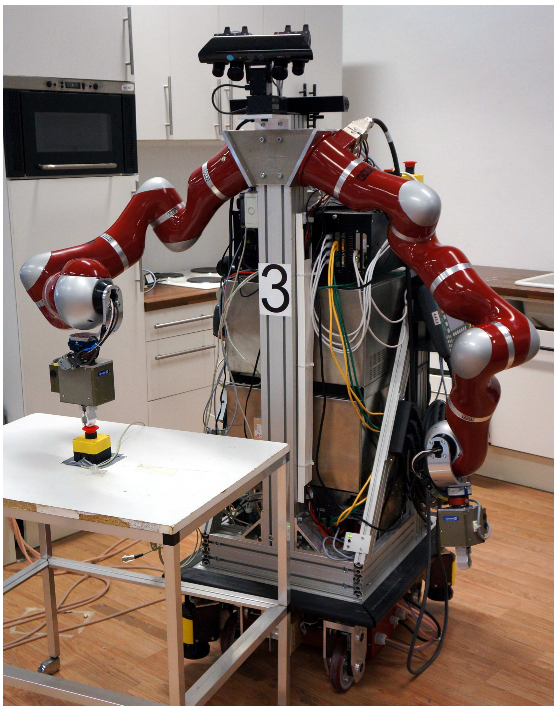


Figure A.3: The robotic platform consisting on two KUKA LWR 4+ manipulators equipped with two-finger grippers and an omni-directional mobile platform.

Bibliography

[1] N. Hogan, “Impedance control: An approach to manipulation,” in *Proc. ACC*, pp. 304–313, 1984.

[2] M. Lawitzky, A. Mörtl, and S. Hirche, “Load Sharing in Human-Robot Cooperative Manipulation,” in *Proc. IEEE Ro-Man*, pp. 185–191, 2010.

[3] N. Sebanz and G. Knoblich, “Prediction in Joint Action: What, When, and Where,” *Topics in Cognitive Science*, no. 2, pp. 353–367, 2009.

[4] N. Jarrassé, J. Paik, V. Pasqui, and G. Morel, “How Can Human Motion Prediction Increase Transparency?,” in *Proc. IEEE ICRA*, pp. 2134–2139, 2008.

[5] J. Medina, M. Lawitzky, A. Mörtl, D. Lee, and S. Hirche, “An Experience-Driven Robotic Assistant Acquiring Human Knowledge to Improve Haptic Cooperation,” in *Proc. IEEE/RSJ IROS*, pp. 2416–2422, 2011.

[6] J. Medina, M. Shelley, D. Lee, W. Takano, and S. Hirche, “Towards Interactive Physical Robotic Assistance: Parameterizing Motion Primitives Through Natural Language,” in *Proc. IEEE Ro-Man*, pp. 1097–1102, 2012.

[7] J. Medina and S. Hirche, “Impedance-based Gaussian Processes for Modeling Human Behavior,” vol. , pp. – , 2015. in preparation.

[8] J. Medina and S. Hirche, “Uncertainty-dependent Locally-Optimal Feedback Control Considering High-order Cost Statistics,” vol. , pp. – , 2015. in preparation.

[9] J. Medina, D. Lee, and S. Hirche, “Risk Sensitive Optimal Feedback Control for Haptic Assistance,” in *Proc. IEEE ICRA*, 2012.

[10] J. Medina, D. Lee, and S. Hirche, “Disagreement-Aware Physical Assistance Through Risk-Sensitive Optimal Feedback Control,” in *Proc. IEEE/RSJ IROS*, 2012.

[11] J. Medina, D. Sieber, and S. Hirche, “Risk-Sensitive Interaction Control in Uncertain Manipulation Tasks,” in *Proc. IEEE ICRA*, pp. 502–507, 2013.

[12] J. Medina, M. Lawitzky, A. Molin, and S. Hirche, “Dynamic Strategy Selection for Physical Robotic Assistance in Partially Known Tasks,” in *Proc. IEEE ICRA*, pp. 1180–1186, 2013.

[13] J. Medina, T. Lorenz, and S. Hirche, “Synthesizing Anticipatory Haptic Assistance Considering Human Behavior Uncertainty,” *IEEE Trans. Robot.*, vol. , pp. – , 2015. to appear.

[14] A. D. Santis, B. Siciliano, A. D. Luca, and A. Bicchi, “An Atlas of Physical Human-Robot Interaction,” *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2007.

[15] D. Prattichizzo and J. C. Trinkle, “Grasping,” *Springer handbook of robotics*, pp. 671–700, 2008.

[16] F. Caccavale and M. Uchiyama, “Cooperative manipulators,” *Springer handbook of robotics*, pp. 701–718, 2008.

[17] S. Schneider and R. Cannon, “Object Impedance Control for Cooperative Manipulation: Theory and Experimental Results,” *IEEE Trans. Robot. Automat.*, vol. 8, no. 3, pp. 383–394, 1992.

[18] L. Villani and J. De Schutter, “Force control,” *Springer Handbook of Robotics*, pp. 161–185, 2008.

[19] J. Szewczyk, F. Plumet, and P. Bidaud, “Planning and controlling cooperating robots through distributed impedance,” *Journal of Robotic Systems*, vol. 19, no. 6, pp. 283–297, 2002.

[20] S. Erhart and S. Hirche, “Adaptive Force/Velocity Control for Multi-Robot Cooperative Manipulation under Uncertain Kinematic Parameters,” in *Proc. IEEE/RSJ IROS*, pp. 307–314, 2013.

[21] K. Kosuge, H. Yoshida, and T. Fukuda, “Dynamic control for robot-human collaboration,” in *Proc. IEEE Workshop on Robot and Human Communication*, pp. 398–401, 1993.

[22] Y. Hirata, Y. Kume, Z.-D. Wang, and K. Kosuge, “Decentralized Control of Multiple Mobile Manipulators based on Virtual 3-D Caster Motion for Handling an Object in Cooperation with a Human,” in *Proc. IEEE ICRA*, vol. 1, pp. 938 – 943 vol.1, 2003.

[23] J. E. Colgate and N. Hogan, “Robust control of dynamically interacting systems,” *Int. J. Control*, vol. 48, no. 1, pp. 65–88, 1988.

[24] G. Niemeyer and J.-J. Slotine, “Stable adaptive teleoperation,” *IEEE Journal of Oceanic Engineering*, vol. 16, no. 1, pp. 152–162, 1991.

[25] L. Parker, “Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems,” *Jo. Ph. A.*, vol. 2, pp. 5–14, 2008.

[26] K. Yokoyama, H. Handa, T. Isozumi, Y. Fukase, K. Kaneko, F. Kanehiro, Y. Kawai, F. Tomita, and H. Hirukawa, "Cooperative Works by a Human and a Humanoid Robot," in *Proc. IEEE ICRA*, pp. 2985–2991, 2003.

[27] Z. Wang, A. Peer, and M. Buss, "An HMM approach to realistic haptic human-robot interaction," in *Proc. IEEE EuroHaptics*, pp. 374–379, 2009.

[28] N. Stefanov, A. Peer, and M. Buss, "Online intention recognition for computer-assisted teleoperation," in *Proc. IEEE ICRA*, pp. 5334–5339, 2010.

[29] V. Duchaine and C. Gosselin, "General Model of Human-Robot Cooperation Using a Novel Velocity Based Variable Impedance Control," in *Proc. IEEE WHC*, pp. 446–451, 2007.

[30] V. Duchaine and C. Gosselin, "Safe, Stable and Intuitive Control for Physical Human-Robot Interaction," in *Proc. IEEE ICRA*, pp. 3383–3388, 2009.

[31] T. Flash and N. Hogan, "The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model," *J. Neurosci.*, vol. 5, pp. 1688–1703, 1985.

[32] Y. Maeda, T. Hara, and T. Arai, "Human-robot Cooperative Manipulation with Motion Estimation," in *Proc. IEEE/RSJ IROS*, vol. 4, pp. 2240–2245, 2001.

[33] B. Corteville, E. Aertbelien, H. Bruyninckx, J. D. Schutter, and H. V. Brussel, "Human-inspired Robot Assistant for Fast Point-to-point Movements," in *Proc. IEEE ICRA*, pp. 3639–3644, 2007.

[34] S. Miossec and A. Kheddar, "Human motion in cooperative tasks: Moving object case study," in *Proc. IEEE ROBIO*, pp. 1509–1514, 2008.

[35] J. Rosell, C. Vázquez, A. Pérez, and P. Iñiguez, "Motion Planning for Haptic Guidance," *J. Intell. Robotic. Syst.*, vol. 53, no. 3, pp. 223–245, 2008.

[36] C. Vázquez, J. Rosell, L. Chirinos, and O. Domínguez, "Haptic Primitives Guidance Based on the Kautham Path Planner," in *Proc. IEEE/RSJ IROS*, pp. 4686–4691, 2010.

[37] M. Lawitzky, J. Medina, D. Lee, and S. Hirche, "Feedback Motion Planning and Learning from Demonstration in Physical Robotic Assistance: Differences and Synergies," in *Proc. IEEE/RSJ IROS*, pp. 3646–3652, 2012.

[38] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot Programming by Demonstration," in *Handbook of Robotics* (B. Siciliano and O. Khatib, eds.), pp. 1371–1394, Springer, 2008.

[39] V. Fernandez, C. Balaguer, D. Blanco, and M. Salichs, "Active Human-Mobile Manipulator Cooperation Through Intention Recognition," in *Proc. IEEE ICRA*, pp. 2668–2673, 2001.

[40] P. Evrard, E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar, "Teaching Physical Collaborative Tasks: Object-Lifting Case Study with a Humanoid," in *Proc. IEEE Humanoids*, pp. 399–404, 2009.

[41] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard, and A. Kheddar, "Learning collaborative manipulation tasks by demonstration using a haptic interface," in *Proc. ICAR*, pp. 1–6, 2009.

[42] E. Gribovskaya, A. Kheddar, and A. Billard, "Motion Learning and Adaptive Impedance for Robot Control during Physical Interaction with Humans," in *Proc. IEEE ICRA*, pp. 4326–4332, 2011.

[43] S. Ikemoto, H. Amor, T. Minato, B. Jung, and H. Ishiguro *IEEE Robot. Autom. Mag.*, 2012.

[44] R. Groten, D. Feth, A. Peer, and M. Buss, "Shared Decision Making in a Collaborative Task with Reciprocal Haptic Feedback - an Efficiency-Analysis," in *Proc. IEEE ICRA*, 2010.

[45] D. Feth, R. Groten, A. Peer, S. Hirche, and M. Buss, "Performance Related Energy Exchange in Haptic Human-Human Interaction in a Shared Virtual Object Manipulation Task," in *Proc. EHS EuroHaptics'09*, pp. 338–343, 2009.

[46] K. Reed and M. Peshkin, "Physical collaboration of human-human and human-robot teams," *IEEE Trans. Haptics*, vol. 1, no. 2, pp. 108–120, 2008.

[47] P. Evrard and A. Kheddar, "Homotopy Switching Model for Dyad Haptic Interaction in Physical Collaborative Tasks," in *Proc. EHS EuroHaptics*, pp. 45–50, 2009.

[48] A. Thobbi, Y. Gu, and W. Sheng, "Using Human Motion Estimation for Human-Robot Cooperative Manipulation," in *Proc. IEEE/RSJ IROS*, pp. 2873–2878, 2011.

[49] A. Mörtl, M. Lawitzky, A. Küçükyılmaz, M. Sezgin, C. Basdogan, and S. Hirche, "The Role of Roles: Physical Cooperation between Humans and Robots," *Int. J. Robot. Res.*, 2012.

[50] C. Passenberg, R. Groten, A. Peer, and M. Buss, "Towards real-time haptic assistance adaptation optimizing task performance and human effort," in *Proc. IEEE WHC*, pp. 155–160, 2011.

[51] D. A. Abbink, M. Mulder, and E. R. Boer, "Haptic shared control: smoothly shifting control authority?," *Cognition, Technology & Work*, vol. 14, no. 1, pp. 19–28, 2012.

[52] R. K. Groten, *Haptic Human-Robot Collaboration: How to Learn from Human Dyads*. Dissertation, Technische Universität München, München, 2011.

[53] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Rob Auton Syst*, vol. 57, no. 5, pp. 469–483, 2009.

[54] T. Lozano-Perez, "Robot programming," *Proc. IEEE*, vol. 71, no. 7, pp. 821–841, 1983.

[55] H. Asada and H. Izumi, "Automatic program generation from teaching data for the hybrid control of robots," *IEEE Trans. Robot. Autom.*, vol. 5, pp. 166–173, Apr 1989.

[56] A. Ude, "Trajectory generation from noisy positions of object features for teaching robot paths," *Rob Auton Syst*, vol. 11, no. 2, pp. 113–127, 1993.

[57] N. Delson and H. West, "Robot programming by human demonstration: adaptation and inconsistency in constrained motion," in *Proc. IEEE ICRA*, vol. 1, pp. 30–36 vol.1, Apr 1996.

[58] K. Ogawara, J. Takamatsu, H. Kimura, and K. Ikeuchi, "Extraction of essential interactions through multiple observations of human demonstrations," *IEEE Trans. Ind. Electron*, vol. 50, pp. 667–675, Aug 2003.

[59] J. Yang, Y. Xu, and C. S. Chen, "Human action learning via hidden markov model," *IEEE Trans. Syst., Man, Cybern A, Syst.,Humans*, vol. 27, pp. 34–44, Jan 1997.

[60] D. Lee and Y. Nakamura, "Mimesis scheme using a monocular vision system on a humanoid robot," in *Proc. IEEE ICRA*, pp. 2162–2168, IEEE, 2007.

[61] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[62] J. Bilmes, "A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," tech. rep., U.C. Berkeley, 1997.

[63] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Trans. Syst., Man, Cybern. B*, vol. 37, no. 2, pp. 286 –298, 2007.

[64] H. G. Sung, *Gaussian mixture regression and classification*. PhD thesis, Rice University, 2004.

[65] D. Kulic, D. Lee, C. Ott, and Y. Nakamura, "Incremental learning of full body motion primitives for humanoidrobots," in *IEEE Humanoids*, pp. 326–332, 2008.

[66] W. Takano, A. Matsushita, K. Iwao, and Y. Nakamura, "Recognition of human driving behaviors based on stochastic symbolization of time series signal," in *Proc. IEEE/RSJ IROS*, pp. 167 –172, 2008.

[67] S. Okada, Y. Kobayashi, S. Ishibashi, and T. Nishida, "Incremental learning of gestures for human–robot interaction," *AI & Soc.*, 2009.

[68] D. Vasquez, T. Fraichard, and C. Laugier, "Growing hidden markov models: An incremental tool for learning and predicting human and vehicle motion," *Int. J. Rob. Res.*, vol. 28, pp. 1486–1506, November 2009.

[69] H. Zen, K. Tokuda, and T. Kitamura, "An Introduction of Trajectory Model into HMM-based Speech Synthesis," in *Proc. ISCA*, pp. 191–196, 2004.

[70] D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Auton. Robot.*, pp. 1–17, 2011.

[71] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," tech. rep., 2002.

[72] K. R. Dixon and P. K. Khosla, "Trajectory representation using sequenced linear dynamical systems," in *Proc. IEEE ICRA*, vol. 4, pp. 3925–3930, 2004.

[73] S. Calinon, F. D'Halluin, E. Sauser, D. Caldwell, and A. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robot. Autom. Mag.*, vol. 17, no. 2, pp. 44 –54, 2010.

[74] E. Gribovskaya, S. Khansari-Zadeh, and A. Billard, "Learning Nonlinear Multivariate Dynamics of Motion in Robotic Manipulators," *Int. J. Rob. Res.*, 2010.

[75] D. Nguyen-Tuong and J. Peters, "Using model knowledge for learning inverse dynamics," in *Proc. IEEE ICRA*, pp. 2677–2682, 2010.

[76] D. Nguyen-Tuong and J. Peters, "Online Kernel-based Learning for Task-Space Tracking Robot Control," no. 9, pp. 1417–1425, 2012.

[77] S. M. Khansari-Zadeh and A. Billard, "Learning Stable Non-Linear Dynamical Systems with Gaussian Mixture Models," *IEEE Transaction on Robotics*, 2011.

[78] C. E. Rasmussen, "Gaussian processes for machine learning," 2006.

[79] M. A. Alvarez and N. D. Lawrence, "Computationally efficient convolved multiple output gaussian processes," *The Journal of Machine Learning Research*, vol. 12, pp. 1459–1500, 2011.

[80] D. H. Grollman and O. C. Jenkins, "Sparse incremental learning for interactive robot control policy estimation," in *Proc. IEEE ICRA*, pp. 3315–3320, 2008.

[81] P. Pastor, M. Kalakrishnan, J. Binney, J. Kelly, L. Righetti, G. Sukhatme, and S. Schaal, "Learning Task Error Models for Manipulation," in *Proc. IEEE ICRA*, 2013.

[82] A. Deshpande, J. Ko, D. Fox, and Y. Matsuoka, "Control strategies for the index finger of a tendon-driven hand," *Int. J. Robot Res.*, vol. 32, no. 1, pp. 115–128, 2013.

[83] M. Deisenroth, D. Fox, and C. Rasmussen, "Gaussian Processes for Data-Efficient Learning in Robotics and Control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, no. 99, pp. 1–1, 2013.

[84] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," *Advances in neural information processing systems*, vol. 16, pp. 329–336, 2004.

[85] J. Wang, D. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007.

[86] R. Rosipal and N. Krämer, "Overview and recent advances in partial least squares," in *Subspace, Latent Structure and Feature Selection*, pp. 34–51, Springer, 2006.

[87] S. Vijayakumar, A. D'Souza, and S. Schaal, "Incremental Online Learning in High Dimensions," *Neural Computation*, vol. 17, pp. 2602–2634, 2005.

[88] S. K. D. Mitrovic and S. Vijayakumar, "Learning impedance control of antagonistic systems based on stochastic optimization principles," *IJRR*, no. 5, pp. 556–573, 2010.

[89] J.-A. Ting, S. Vijayakumar, and S. Schaal, "Locally weighted regression for control," in *Encyclopedia of Machine Learning*, pp. 613–624, Springer, 2010.

[90] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," in *Robotics Research*, pp. 561–572, Springer, 2005.

[91] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance," in *Proc. IEEE ICRA*, pp. 2587–2592, IEEE, 2009.

[92] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Proc. IEEE ICRA*, pp. 763–768, 2009.

[93] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, "Online movement adaptation based on previous sensor experiences," in *Proc. IEEE/RSJ IROS*, pp. 365–371, 2011.

[94] A. Gams, A. J. Ijspeert, S. Schaal, and J. Lenarčič, "On-line learning and modulation of periodic movements with nonlinear dynamical systems," *Autonomous robots*, vol. 27, no. 1, pp. 3–23, 2009.

[95] 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.

[96] G. Gergely, Z. Nádasdy, G. Csibra, and S. Biro, "Taking the intentional stance at 12 months of age," *Cognition*, vol. 56, no. 2, pp. 165–193, 1995.

[97] A. Y. Ng, S. J. Russell, *et al.*, "Algorithms for inverse reinforcement learning.," in *Icml*, pp. 663–670, 2000.

[98] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proc. ICML*, p. 1, 2004.

[99] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning," *Urbana*, vol. 51, p. 61801, 2007.

[100] B. D. Ziebart, J. A. Bagnell, and A. K. Dey, "The principle of maximum causal entropy for estimating interacting processes," *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 1966–1980, 2013.

[101] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning.,” in *AAAI*, pp. 1433–1438, 2008.

[102] S. Levine and V. Koltun, “Continuous inverse optimal control with locally optimal examples,” in *Proc. ICML*, 2012.

[103] S. Levine, Z. Popovic, and V. Koltun, “Nonlinear inverse reinforcement learning with gaussian processes,” in *Advances in Neural Information Processing Systems*, pp. 19–27, 2011.

[104] E. Todorov, “Optimal control theory,” 2006.

[105] R. E. Bellman and S. E. Dreyfus, “Applied dynamic programming,” 1962.

[106] D. Jacobson, “Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic differential games,” *IEEE Trans. Autom. Control*, vol. 18, pp. 124 – 131, apr 1973.

[107] P. Whittle, “Risk-sensitive linear/quadratic/gaussian control,” *Advances in Applied Probability*, vol. 13, no. 4, pp. pp. 764–777, 1981.

[108] A. Bensoussan and J. Van Schuppen, “Optimal control of partially observable stochastic systems with an exponential-of-integral performance index,” *SIAM Journal on Control and Optimization*, vol. 23, no. 4, pp. 599–613, 1985.

[109] C. Won, “Cost distribution shaping: the relation between bode integral, entropy, risk-sensitivity, and cost cumulant control,” in *Proc. ACC*, vol. 3, pp. 2160–2165 vol.3, 2004.

[110] S. Kuindersma, R. Grupen, and A. Barto, “Variational bayesian optimization for runtime risk-sensitive control,” *Robotics*, p. 201, 2013.

[111] K. Rawlik, M. Toussaint, and S. Vijayakumar, “On stochastic optimal control and reinforcement learning by approximate inference,” in *Proc. IJCAI*, pp. 3052–3056, AAAI Press, 2013.

[112] K. Rawlik, “On probabilistic inference approaches to stochastic optimal control,” 2013.

[113] L. Cosenza, *On the minimum variance control of discrete-time systems.* PhD thesis, University of Notre Dame, 1969.

[114] M. K. Sain, C.-H. Won, and B. Spencer, “Cumulant minimization and robust control,” in *Stochastic Theory and Adaptive Control*, pp. 411–425, Springer, 1992.

[115] M. Sain, C. Won, J. Spencer, and S. Liberty, “Cumulants and risk-sensitive control: a cost mean and variance theory with application to seismic protection of structures,” in *Advances in Dynamic Games and Applications*, pp. 427–459, Springer, 2000.

[116] K. Pham, *Linear-Quadratic Controls in Risk-Averse Decision Making: Performance-Measure Statistics and Control Decision Optimization*. Springer, 2012.

[117] F. Qian, J. Gao, and D. Li, "Complete statistical characterization of discrete-time lqg and cumulant control," *IEEE Trans. Autom. Control*, vol. 57, no. 8, pp. 2110–2115, 2012.

[118] K. Pham, M. Sain, and S. Liberty, "Cost cumulant control: state-feedback, finite-horizon paradigm with application to seismic protection," *Journal of optimization theory and applications*, vol. 115, no. 3, pp. 685–710, 2002.

[119] D. H. Jacobson, "New second-order and first-order algorithms for determining optimal control: A differential dynamic programming approach," *Journal of Optimization Theory and Applications*, vol. 2, no. 6, pp. 411–440, 1968.

[120] B. Janus and Y. Nakamura, "Unsupervised Probabilistic Segmentation of Motion Data for Mimesis Modeling," in *Proc. ICAR*, pp. 411 –417, 2005.

[121] J. Kohlmorgen and S. Lemm, "A Dynamic HMM for On-line Segmentation of Sequential Data," in *Advances in Neural Information Processing Systems 14*, pp. 793–800, MIT Press, 2002.

[122] W. Takano, K. Yamane, T. Sugihara, K. Yamamoto, and Y. Nakamura, "Primitive communication based on motion recognition and generation with hierarchical mimesis model," in *ICRA*, pp. 3602–3609, 2006.

[123] T. Ogata, S. Matsumoto, J. Tani, K. Komatani, and H. G. Okuno, "Human-robot cooperation using quasi-symbols generated by rnnpb model," in *ICRA*, pp. 2156–2161, 2007.

[124] T. Miyashita, T. Tajika, H. Ishiguro, K. Kogure, and N. Hagita, "Haptic communication between humans and robots," in *ISRR*, pp. 525–536, 2005.

[125] Y. Sugita and J. Tani, "Learning semantic combinatoriality from the interaction between linguistic and behavioral processes," *ADAPTIVE BEHAVIOR*, vol. 13, pp. 33–52, 2005.

[126] K. Sugiura, N. Iwahashi, H. Kashioka, and S. Nakamura, "Active learning of confidence measure function in robot language acquisition framework," in *IROS*, pp. 1774–1779, 2010.

[127] W. Takano, K. Yamane, and Y. Nakamura, "Capture database through symbolization, recognition and generation of motion patterns," in *ICRA*, pp. 3092–3097, 2007.

[128] W. Takano and Y. Nakamura, "Statistically integrated semiotics that enables mutual inference between linguistic and behavioral symbols for humanoid robots," in *Proc. IEEE ICRA*, pp. 646–652, 2009.

[129] W. Takano and Y. Nakamura, "Associative processes between behavioral symbols and a large scale language model," in *Proc. IEEE ICRA*, pp. 2404–2409, 2010.

[130] A. Wilson and A. Bobick, "Parametric hidden markov models for gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 9, pp. 884–900, 1999.

[131] D. Herzog and V. Krüger, "Statistical and geometrical approaches to visual motion analysis," ch. Recognition and Synthesis of Human Movements by Parametric HMMs, pp. 148–168, Springer-Verlag, 2009.

[132] W. Takano and Y. Nakamura, "Bigram-based natural language model and statistical motion symbol model for scalable language of humanoid robots," in *Proc. IEEE ICRA*, 2012.

[133] J. D. Schutter, "Invariant Description of Rigid Body Motion Trajectories," *ASME Journal of Mechanisms and Robotics*, vol. 2, pp. 011004–1–011004–9, 2010.

[134] A. Van Der Vaart and H. Van Zanten, "Information rates of nonparametric gaussian process methods," *The Journal of Machine Learning Research*, vol. 12, pp. 2095–2119, 2011.

[135] T. Tsuji, P. G. Morasso, K. Goto, and K. Ito, "Human hand impedance characteristics during maintained posture," *Biological cybernetics*, vol. 72, no. 6, pp. 475–485, 1995.

[136] H. Gomi and M. Kawato, "Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement," *Science*, vol. 272, no. 5258, pp. 117–120, 1996.

[137] M. Alvarez and N. D. Lawrence, "Sparse convolved gaussian processes for multi-output regression," in *Advances in neural information processing systems*, pp. 57–64, 2009.

[138] E. Snelson, C. E. Rasmussen, and Z. Ghahramani, "Warped gaussian processes," *Advances in neural information processing systems*, vol. 16, pp. 337–344, 2004.

[139] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," 2006.

[140] E. Snelson and Z. Ghahramani, "Local and global sparse gaussian process approximations," in *Proc. AISTATS*, pp. 524–531, 2007.

[141] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Real-time local gp model learning," in *From Motor Learning to Interaction Learning in Robots*, pp. 193–207, Springer, 2010.

[142] S. V. Vaerenbergh, M. Lázaro-Gredilla, and I. Santamaría, "Kernel recursive least-squares tracker for time-varying regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1313–1326, 2012.

[143] A. Faisal, L. Selen, and D. Wolpert, "Noise in the nervous system," *Nature Reviews Neuroscience*, vol. 9, no. 4, pp. 292–303, 2008.

[144] E. Todorov, "Optimality principles in sensorimotor control.," *Nature Neuroscience*, vol. 7, pp. 907–15, 2004.

[145] J. Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1263–1278, 2012.

[146] D. Mitrovic, S. Klanke, and S. Vijayakumar, “Adaptive optimal feedback control with learned internal dynamics models,” in *From Motor Learning to Interaction Learning in Robots* (O. Sigaud and J. Peters, eds.), vol. 264 of *Studies in Computational Intelligence*, pp. 65–84, Springer Berlin Heidelberg, 2010.

[147] D. A. Braun, A. J. Nagengast, and D. M. Wolpert, “Risk-sensitivity in sensorimotor control,” *Frontiers in Human Neuroscience*, vol. 5, no. 1, 2011.

[148] J. Grau-Moya, P. Ortega, and D. A. Braun, “Risk-sensitivity in bayesian sensorimotor integration,” *PLoS Comput Biol*, vol. 8, p. e1002698, 09 2012.

[149] J. Grau-Moya, E. Hez, G. Pezzulo, and D. Braun, “The Effect of Model Uncertainty on Cooperation in Sensorimotor Interactions,”

[150] J. Müller and G. Sukhatme, “Risk-aware trajectory generation with application to safe quadrotor landing,” in *Proc. IEEE/RSJ IROS*, 2014.

[151] E. Todorov and W. Li, “A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems,” in *Proc. ACC*, pp. 300–306, 2005.

[152] D. Jacobsen and M. D, *Differential Dynamic Programming*. Elsevier.

[153] V. Huynh and R. N, “iclqg: Combining local and global optimization for control in information space,” in *Proc. IEEE ICRA*, pp. 2851–2858, 2009.

[154] T. Erez and W. Smart, “A scalable method for solving high-dimensional continuous pomdps using local approximation,” *arXiv preprint arXiv:1203.3477*, 2012.

[155] P. Whittle, *Risk-Sensitive Optimal Control*. Wiley, 1990.

[156] D. Mitrovic, S. Klanke, R. Osu, M. Kawato, and S. Vijayakumar, “A computational model of limb impedance control based on principles of internal model uncertainty,” *PLoS ONE*, vol. 5, p. e13601, 10 2010.

[157] G. Aoude, B. Luders, J. Joseph, N. Roy, and J. How, “Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns,” *Autonomous Robots*, vol. 35, no. 1, pp. 51–76, 2013.

[158] C. Won, “Cost moment control and verification theorem for nonlinear stochastic systems,” in *Proc. IEEE CDC*, pp. 2583–2588, IEEE, 2006.

[159] F. Lewis and D. Vrabie, “Reinforcement learning and adaptive dynamic programming for feedback control,” *Circuits and Systems Magazine, IEEE*, vol. 9, no. 3, pp. 32–50, 2009.

[160] S. Yakowitz, “Algorithms and computational techniques in differential dynamic programming,” *Control and Dynamic Systems*, vol. 31, pp. 75–91, 1989.

[161] S. Calinon and A. Billard, “Statistical learning by imitation of competing constraints in joint space and task space,” *Advanced Robotics*, vol. 23, pp. 2059–2076, 2009.

[162] U. Unterhinninghofen, T. Schauß, and M. Buss, “Control of a Mobile Haptic Interface,” in *Proc. IEEE ICRA*, pp. 2085–2090, 2008.

[163] N. Nitzsche, U. Hanebeck, and G. Schmidt, “Design Issues of Mobile Haptic Interfaces,” *J. Robotic Syst.*, vol. 20, no. 9, pp. 549–556, 2003.

[164] W. Feiten, P. Atwal, R. Eidenberger, and T. Grundmann, “6D Pose Uncertainty in Robotic Perception,” in *Proc. GWR*, pp. 1–10, 2009.

[165] K. L. Doty, C. Melchiorri, and C. Bonivento, “A theory of generalized inverses applied to robotics,” *The International Journal of Robotics Research*, vol. 12, no. 1, pp. 1–19, 1993.

[166] I. D. Walker, R. A. Freeman, and S. I. Marcus, “Analysis of motion and internal loading of objects grasped by multiple cooperating manipulators,” *The International journal of robotics research*, vol. 10, no. 4, pp. 396–409, 1991.

[167] M. Uchiyama and P. Dauchez, “A symmetric hybrid position/force control scheme for the coordination of two robots,” in *Proc. IEEE ICRA*, pp. 350–356, 1988.

[168] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*, vol. 3. Wiley New York, 2006.

[169] K. Kronander and A. Billard, “Online learning of varying stiffness through physical human-robot interaction,” in *Proc. IEEE ICRA*, 2012.

[170] C. E. Garcia, D. M. Prett, and M. Morari, “Model predictive control: theory and practice–a survey,” *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[171] S. Boyd, L. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994.

[172] A. Shaiju and I. Petersen, “Formulas for discrete time LQR, LQG LEQG and minimax LQG optimal control,” in *Proc. IFAC*, 2008.

[173] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*, vol. 1. Wiley-interscience New York, 1972.

[174] A. Jain and C. Kemp, “Pulling open doors and drawers: Coordinating an omni-directional base and a compliant arm with equilibrium point control,” in *Proc. IEEE ICRA*, pp. 1807 –1814, 2010.

[175] M. Rambow, S. Hirche, and M. Buss, “Autonomous manipulation of deformable objects based on teleoperated demonstrations,” in *Proc. IEEE/RSJ IROS*, 2012.

[176] A. Saxena, J. Driemeyer, and A. Ng, “Robotic grasping of novel objects using vision,” *Int. J. Robot. Res.*, vol. 27, no. 2, pp. 157–173, 2008.

[177] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, “Online movement adaptation based on previous sensor experiences,” in *Proc. IEEE/RSJ IROS*, pp. 365 –371, 2011.

[178] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou, and S. Schaal, "Skill learning and task outcome prediction for manipulation," in *Proc. IEEE ICRA*, pp. 3828 –3834, 2011.

[179] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 126–133, 1981.

[180] S. Chiaverini and L. Sciavicco, "The parallel approach to force/position control of robotic manipulators," *IEEE Trans. Robot. Autom.*, vol. 9, pp. 361 –373, aug 1993.

[181] R. Olfati-Saber, A. Fax, and R. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proc. IEEE*, vol. 95, pp. 215–233, 2007.

[182] L. Yang and S. LaValle, "The Sampling-Based Neighborhood Graph: An Approach to Computing and Executing Feedback Motion Strategies," *IEEE Trans. Robot. Automat.*, vol. 20, no. 3, pp. 419–432, 2004.

[183] P. Date and B. Gashi, "Risk-sensitive control for a class of nonlinear systems with multiplicative noise," *Systems & Control Letters*, vol. 62, no. 10, pp. 988 – 999, 2013.

[184] T. Erez, Y. Tassa, and E. Todorov, "Infinite-horizon model predictive control for periodic tasks with contacts," *Robotics: Science and Systems VII*, p. 73, 2012.

[185] M. Lawitzky, J. Medina, and S. Hirche, "Rapid prototyping of planning, learning and control in physical human-robot interaction," in *Proc. International Symposium on Experimental Robotics*, pp. 819–824, 2012.

[186] U. Hanebeck, N. Saldic, and G. Schmidt, "A Modular Wheel System for Mobile Robot Applications," in *Proc. IEEE/RSJ IROS*, pp. 17–22, 1999.

[187] B. Stanczyk and M. Buss, "Development of a Telerobotic System for Exploration of Hazardous Environments," in *Proc. IEEE/RSJ IROS*, pp. 2532–2537, 2004.

[188] D. Brscic, M. Eggers, F. Rohrmüller, O. Kourakos, S. Sosnowski, D. Althoff, M. Lawitzky, A. Mörtl, M. Rambow, V. Koropouli, J. Medina, X. Zang, W. Wang, D. Wollherr, K. Kühnlenz, C. Mayer, T. Kruse, A. Kirsch, J. Blume, A. Bannat, T. Rehrl, F. Wallhoff, T. Lorenz, P. Basili, C. Lenz, T. Röder, G. Panin, W. Maier, S. Hirche, M. Buss, M. Beetz, B. Radig, A. Schubö, S. Glasauer, A. Knoll, and E. Steinbach, "Multi Joint Action in CoTeSys - Setup and Challenges," tech. rep., Technische Universität München and Ludwig-Maximilians-Universität München, 2010.

[189] M. Goebl and G. Färber, "A Real-Time-capable Hard- and Software Architecture for Joint Image and Knowledge Processing in Cognitive Automobiles," in *Proc. IEEE IV*, pp. 734–740, 2007.

[190] M. Goebl, "KogMo-RTDB - Real-time Database for Cognitive Automobiles," 2011.

[191] The Preempt RT community, "Real-Time Linux Wiki," 2011.

[192] D. Althoff, O. Kourakos, M. Lawitzky, A. Mörtl, M. Rambow, F. Rohrmüller, D. Brščić, D. Wollherr, S. Hirche, and M. Buss, “An architecture for real-time control in multi-robot systems,” *Human Centered Robot Systems*, pp. 43–52, 2009.