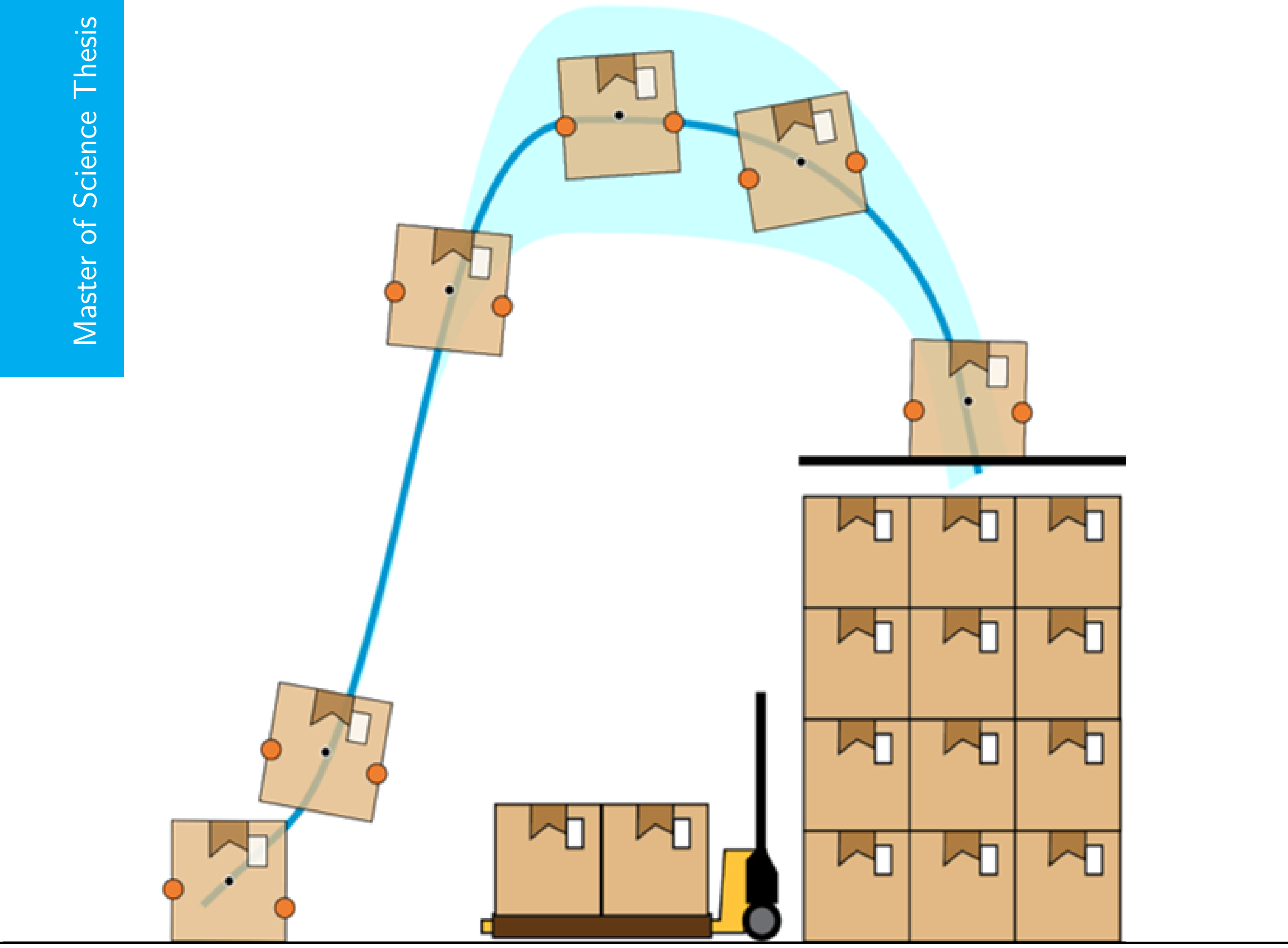
Impact-Aware
Learning from Demonstration

S.W. de Zwart

Master of Science Thesis



Delft University of Technology

Impact-Aware
Learning from Demonstration

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

S.W. de Zwart

November 27, 2019

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

TU Delft Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Impact-Aware
Learning from Demonstration

by

S.W. de Zwart

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: November 27, 2019

Supervisor(s):

dr.ir. Jens Kober

dr.ir. Alessandro Saccon (TU/e)

Reader(s):

dr.ir. Jens Kober

dr.ir. Alessandro Saccon (TU/e)

dr. ir. René van de Molengraft (TU/e)

Abstract

We often establish contact with our environment at non-zero speed. Grabbing and pushing objects without the need to stop our hands at the moment of impact is an example of this. Although humans learn and execute such tasks with relative ease, robots cannot. The difficulty in executing such tasks lies in the complexity of control at the moment of impact. Traditional control approaches avoid contact at non-zero speed by a so called transition phase in which the relative velocity is reduced to zero near contact. Learning from demonstration refers to the process used to transfer new skills to a machine through human demonstrations instead of traditional, time consuming, robotic programming. The goal of this research is to develop a learning strategy that is able to learn and execute tasks in which contact is made at non-zero speed.

The new learning strategy is an adaptation of the state of the art learning from demonstration method, probabilistic movement primitives, combined with the impact-aware robot control strategy, reference spreading. Probabilistic movement primitives translate demonstration data into a trajectory distribution. Reference spreading tackles the problem of having a different time of impact than expected by defining a new error which compares the current state to an extended reference trajectory, switching to the extended trajectory of another mode upon impact. In this work, these methods are combined by extending the demonstration data, to subsequently fit the probabilistic movement primitives resulting in extended trajectory distribution for multiple modes. This trajectory, in conjunction with the reference spreading error can be used for control. The proposed method is numerically validated by simulating two end effectors, dynamically picking up a box to then put it on top of a shelf. The task is successfully learned and executed, showing the effectiveness of the impact-aware learning strategy.

Table of Contents

Acknowledgements

I would like to take the opportunity to thank the people that helped me during my graduation project and the rest of my studies. Jens, you are an inspiration because of your knowledge of machine learning in robotics. Your lectures and research have motivated me to pursue a career in this field. Thank you for the friendly and honest supervision throughout the research. Alessandro, thank you for your enthusiasm and critical view on my work. I appreciate our meetings and discussions throughout the research. Also, thank you for taking the time to introduce me to new people and giving me the opportunity to present the research to others.

I would like to thank my fellow students in the robotics lab. Being in the robotics lab made my graduation project much more enjoyable. I found great joy in the lab activities, lunch, and discussions we had. I want to express my gratitude to my parents for their support through the ups and downs during my studies. My friends, thank you for taking my mind of the graduation project every now and then.

Most importantly, thank you Eline for your love and support during my entire studies. Also, thank you for your patience when I decided to continue my studies in a different direction.

Lastly, I would like to thank you the reader, for taking the time to read this thesis.

Delft, University of Technology S.W. de Zwart
November 27, 2019

Chapter 1

Introduction

In everyday life, we often establish contact with our environment at non-zero speed. In particular, we can grab and push objects without the need to stop our hands at the moment of contact, or kick a ball with our feet, causing intentional impacts to occur between our body and the objects to be picked up or moved. Learning and executing such a task comes naturally to us after a certain age. Not for robots. To avoid complexity, common control approaches circumvent contact at non-zero speed by reducing the relative velocity to zero in proximity of contact to avoid impact in a so called transition phase [1]. This strategy is effective but also limits the ideally achievable performance, especially when time is of importance, such as for pick and place robots (Figure 1-1). Another limiting factor of these control approaches is that a reference trajectory has to be defined manually. Much of the difficulty involved in this traditional robotic programming can be alleviated through the use of learning from demonstration. Learning from demonstration refers to the process used to transfer new skills to a machine through demonstrations [2]. State of the art methods in the field of robotic learning however currently avoid impact [3]. Being able to make contact at non-zero speed without the difficulty of defining a reference trajectory can be beneficial in industrial settings such as dynamically picking and placing or pushing objects of varying shapes and sizes. Therefore, it can be beneficial to combine a control strategy applicable for making contact at non-zero speed and learning from demonstration.

Figure 1-1: The IRB 360 FlexPicker of ABB [4].

1-1 Related work

This section provides an overview of the research relevant to this thesis. To the best of our knowledge, no prior research has been conducted regarding this combination. The research conducted on the control approaches for contact at non-zero speed, is briefly described in Section 1-1-1. The research conducted on learning from demonstration is covered in Section 1-1-2.

1-1-1 Control approaches for contact at non-zero speed

Controlling a system which makes contact at non-zero speed is complicated due to a jump of the velocity. Such systems, more often than not, make contact at a different time than expected. This jump of the velocity, in combination with a different jump time, results in a peak in the error when using a traditional error definition. Such a peak in the error can result in poor tracking performance or the destabilization of the system. There are several approaches that tackle this problem. In [5], a method is described in which the velocity based feedback of the controller is discarded in a small time interval at the expected impact. This approach only deals with periodic trajectories, similar to the approach of using hybrid zero dynamics [6]. In [7], a new distance measure is defined such that the tracking error does not change at state-triggered jumps. However, this method can only be applied for partially inelastic impacts. In [8], the reference trajectory is mirrored at the expected impact. Reference spreading, which was first introduced in 2014 by Saccon et al. [9], defines a new tracking error. This error compares the current state to a reference trajectory that is extended about its impact time. This extension allows for the comparison of the current state to the reference in the same mode [10]. Unlike the methods of [5, 6], reference spreading is applicable for both periodic and non-periodic trajectories. Also, it is applicable on both elastic and inelastic impacts unlike [7]. Reference spreading is able to handle both single impacts and simultaneous impacts [11]. Due to these properties, the reference spreading error definition is used for control in this research.

1-1-2 Learning from Demonstration

One way for a robot to learn a certain task is through learning from demonstration. The demonstration can take various forms such as kinesthetic learning or observational learning. Observational learning observes a teacher, either through visual observation [12] or through wearable motion sensors [13]. Kinesthetic learning (depicted in Figure 1-2) is a method in which the teacher physically maneuvers the robot [14]. An advantage of kinesthetic learning is that it does not suffer from the correspondence problem [2]. The correspondence problem is the difference in the kinematic model of the teacher and the learner [15]. In this research it is assumed that the kinematic models of the teacher and the learner are equal. An important part of learning a new skill is the ability to generalize. In other words, a learned task should still be able to be successfully completed even if the conditions differ from the conditions in which the task was demonstrated. For instance, if the teacher demonstrated grabbing an object, the learner should ideally be able to grab it even if the position of the object is slightly different.

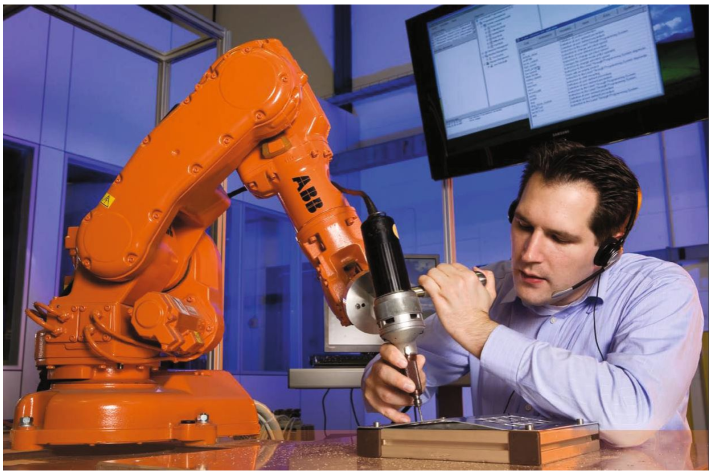


Figure 1-2: Teaching a robot a new skill through kinesthetic teaching [16].

A large field of research within learning from demonstration is to design a compact and adaptive movement representation, such a representation is referred to as a *movement primitive* [2]. Through the use of movement primitives, a movement can be split into smaller motions, which may or may not be repeated. These smaller parts can be seen as building blocks which can be parameterized. These parameters can be learned through, among others, learning from demonstration. A wide variety of movement primitives have been developed and a number of survey papers on the topic is available [17, 18, 19]. Out of the many movement primitives popular ones are Dynamic Movement Primitives (DMPs) [20], Gaussian Mixture Regression (GMR) [21], Stable Estimator of Dynamical Systems (SEDS) [22], and Probabilistic Movement Primitives (ProMPs) [23]. These movement primitives are popular due to their effectiveness at representing movement from demonstration data whilst also embedding the ability to generalize to situations which differ from the demonstrations.

At the time of doing this research, reference spreading has only been applied to time driven trajectories. Out of the movement primitives listed above, ProMP and GMR are able to learn time driven trajectories from demonstration data. The movement primitive GMR proved to be less suited for learning a time driven trajectory as it treats the temporal (time) information of the data equal to the spatial (position) information of the data. The comparison between ProMP and GMR is detailed in Appendix A. This research explores the possibility of combining ProMP with reference spreading.

1-2 Research goal

The goal of this research is to develop a learning strategy that is able to learn and execute robot tasks in which contact is made at non-zero speed.

This goal is achieved by combining ProMP and reference spreading into a novel Impact-Aware Learning from Demonstration (IA-LfD) method. This learning approach is numerically validated through the simulation of two end effectors dynamically picking and placing a box.

1-3 Report outline

The report is structured as follows. In Chapter 2, the mathematical preliminaries and background material are provided. This chapter contains background material on reference spreading for both single and simultaneous impacts, ProMP, a task-based Quadratic Programming (QP) controller and compliant contact and friction models. Thereafter in Chapter 3, the proposed learning method IA-LfD is detailed. Then, in Chapter 4, the simulation in which the proposed method is numerically validated is discussed. Lastly, in Chapter 5 conclusions are drawn and recommendations for further research are given.

Chapter 2

Mathmatical preliminaries and background material

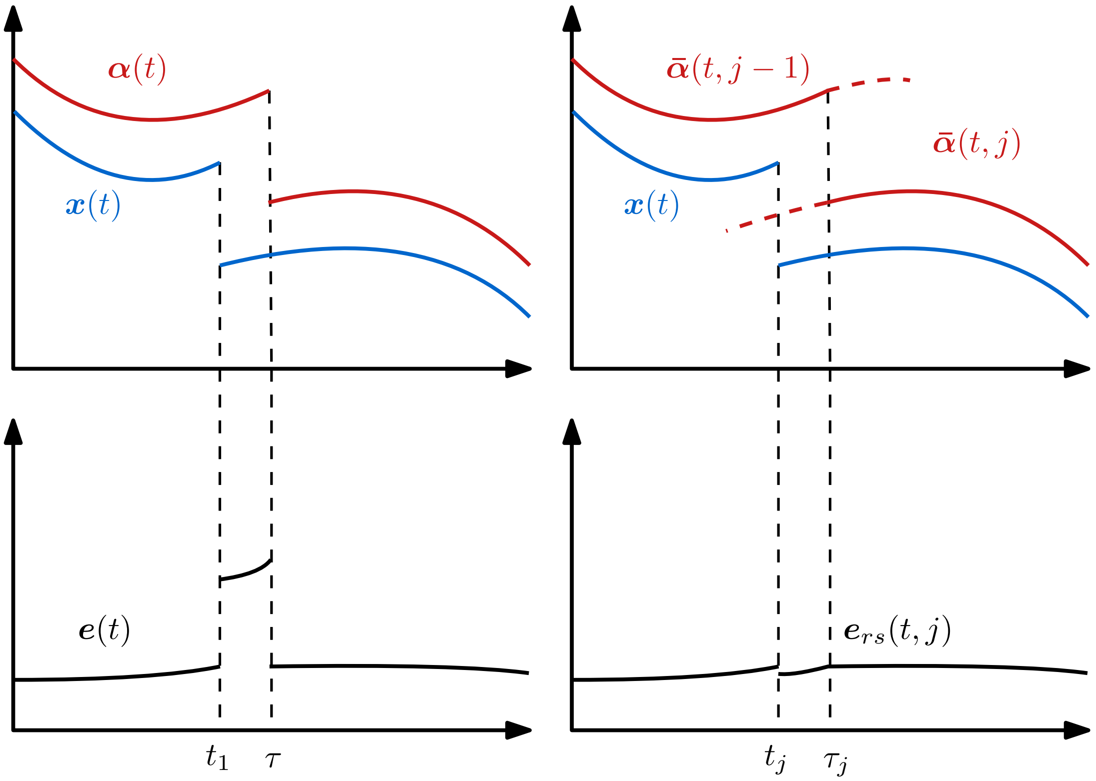
In this chapter an overview is provided of the preliminary information relevant to this thesis. First, the control strategy reference spreading is explained. Thereafter, the used movement primitive Probabilistic Movement Primitive (ProMP), is described. Thirdly, background information is provided on task-based optimal constrained robot control based on Quadratic Programming (QP). Finally the contact models, used for the simulation, are reviewed. The reader is suggested to skim through this chapter at first read, using its content as a reference while reading Chapter 3 and Chapter 4.

2-1 Reference Spreading

This section describes reference spreading. First, the difficulty of tracking tasks in which impact occurs is explained. Thereafter, reference spreading for single impacts is reviewed along with a quick description of hybrid time. Lastly, reference spreading is explained for simultaneous impacts along with an extension of hybrid time. Namely, multi-scale hybrid time. Reference spreading was first introduced by Saccon et al. in 2014 [9]. Reference spreading is a control strategy used for the control of dynamical systems with hybrid dynamics with state-triggered jumps, such as velocity jumps occurring in mechanical systems experiencing physical impact with their surroundings. Reference spreading tackles the peak in the error occurring when the time of impact differs from the expected time of impact.

2-1-1 Reference Spreading for single impacts

When using traditional feedback controllers, desired motions with hard impacts can be difficult to control. The actual time of impact is, more often than not, unequal to the expected time of impact. This time mismatch results in large tracking errors when employing the standard error definition [24]. This large tracking error due to the time mismatch is named the peaking phenomenon and is illustrated in Figure 2-1a.



(a) The peak in the error trajectory $\boldsymbol{e}(t)$.

(b) No peak in the reference spreading error trajectory $\boldsymbol{e}_{rs}(t, j)$.

Figure 2-1: The classical error $\boldsymbol{e}(t)$ and reference spreading error $\boldsymbol{e}_{rs}(t, j)$ resulting from a mismatch between the expected time of impact τ of the reference trajectory $\boldsymbol{\alpha}(t)$ and actual time of impact t_1 of the state trajectory $\boldsymbol{x}(t)$.

The peak in the error is an unwanted occurrence which can result in poor tracking performance or destabilization of the system. To circumvent the peaking phenomenon, a new error is defined. This new error compares the current state with a nominal trajectory known in advance that is extended about its nominal impact times. This extension of the nominal trajectory can be seen as the trajectory the system would take if no impact were to take place. For example, for a ball hitting a floor, the reference trajectory of the ball can be extended as if the ball goes through the floor. Switching from one section of the trajectory to another is triggered by the detection of an impact [10].

To describe such an extended trajectory $\bar{\boldsymbol{\alpha}}$, hybrid time (t, j) is used. The variable t denotes continuous time and j discrete time. For reference spreading, the discrete time variable j is used as an event counter. An event refers to the occurrence of a state triggered jump. In this thesis the events that cause the state triggered jumps are the making of contact at non-zero velocity. The hybrid time domain of a reference trajectory $\boldsymbol{\alpha}$ is given by

$$\text{dom}\,\boldsymbol{\alpha} := \bigcup_{j=0}^{N} I_j^{\alpha} \times \{j\}, \tag{2-1}$$

where N denotes the total number of events, $I_j^{\alpha} = [t_j, t_{j+1}]$ the time interval between the j-th and $(j+1)$-th event. For reference spreading, this domain is extended such that $\bar{I}_j^{\alpha} = [t_j - \Delta_{rs}, t_{j+1} + \Delta_{rs}]$ with $\Delta_{rs} > 0$. This allows for the definition of the extended reference trajectory $\bar{\boldsymbol{\alpha}}(t,j)$, as is illustrated in Figure 2-1b. The reference trajectory can be extended through forward and backward integration. However, this method of extension is not unique. One could, for example, choose to mirror the reference trajectory as in [8], or extend by maintaining a constant velocity.

The extended reference trajectory $\bar{\boldsymbol{\alpha}}(t,j)$ allows for a new error definition. This new error $\boldsymbol{e}_{rs}$, further referred to as the reference spreading error, is formulated as

$$\boldsymbol{e}_{rs}(t,j) := \bar{\boldsymbol{\alpha}}(t,j) - \boldsymbol{x}(t,j). \tag{2-2}$$

This new error notion allows us to compare a state trajectory $\boldsymbol{x}$ that has experienced j jumps to a reference trajectory with the same jump counter [10]. Looking back at the example given in Figure 2-1a, using this error definition results in an error trajectory without the peaking phenomenon (Figure 2-1b). This error definition can be used to define a state feedback with a feedforward, such as

$$\mathbf{u}(\boldsymbol{x},t,j) = \bar{\mathbf{u}}^{ff}(t,j) + \bar{\mathbf{K}}(t,j)(\bar{\boldsymbol{\alpha}}(t,j) - \boldsymbol{x}(t,j)), \tag{2-3}$$

where $\bar{\mathbf{u}}^{ff}(t,j)$ denotes the (hybrid) time-varying feedforward term and $\bar{\mathbf{K}}(t,j)$ the (hybrid) time-varying feedback gain. It is important to note that the control law (2-3) is not the only choice. It is possible to use the newly defined error in combination with with different control laws such as the QP control (Section 2-3), as is done in this thesis.

2-1-2 Reference Spreading for simultaneous impacts

For single impacts, as is described in the previous subsection, it is assumed that contact is made at a single point. However, in reality, contact can be made at multiple points simultaneously. Such simultaneous impacts occur, for example, when a box is placed on the floor with its bottom surface being parallel to the ground. If a reference trajectory is defined such that it contains simultaneous impacts, a small disturbance can result a different time of impact then expected, but also in the impacts not taking place simultaneously. For the example of the box, this would result in one corner of the box making contact before the other. This is illustrated in Figure 2-2.

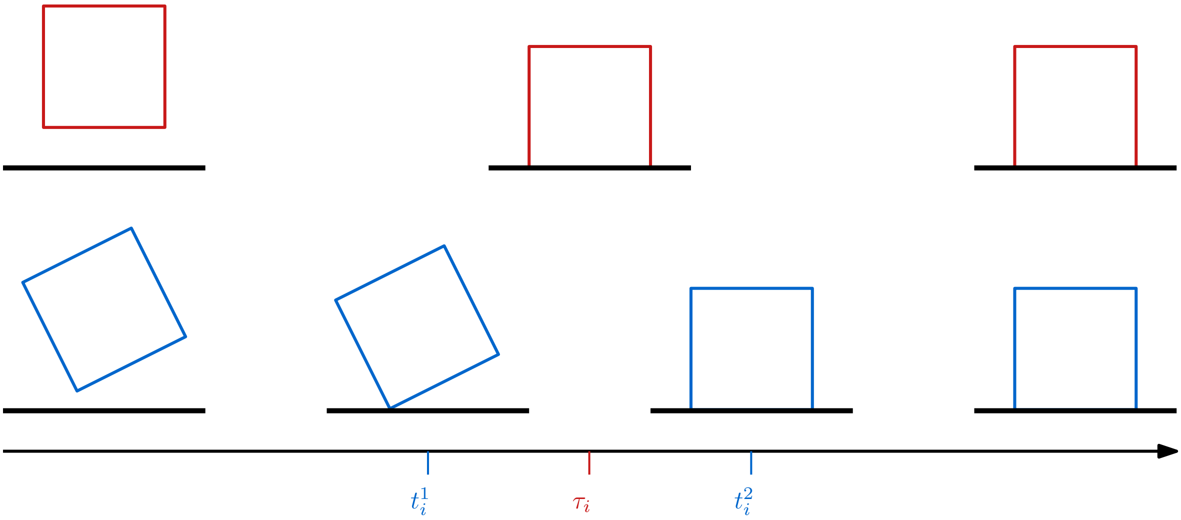


Figure 2-2: Illustration of one simultaneous impact of the red box at τ_i, whilst the blue box makes two near simultaneous impacts at t_i^1 and t_i^2.

Whilst, for this example, the reference trajectory $\boldsymbol{\alpha}$ only has one simultaneous impact, the state trajectory $\boldsymbol{x}$ has multiple isolated impacts. The state trajectory $\boldsymbol{x}$ encounters multiple jumps, while the reference trajectory $\boldsymbol{\alpha}$ encounters one. Such trajectories are depicted in Figure 2-3.

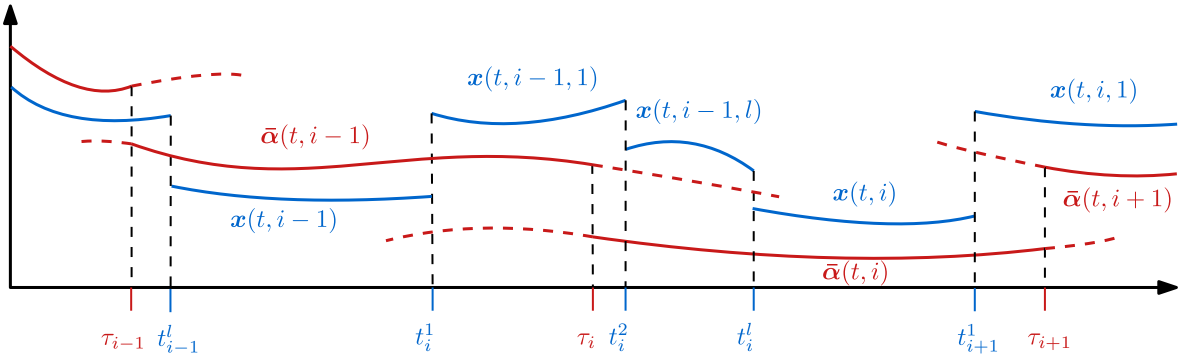

$\boldsymbol{x}(t, i-1, 1)$
$\boldsymbol{x}(t, i-1, l)$
$\boldsymbol{x}(t, i, 1)$
$\bar{\boldsymbol{\alpha}}(t, i-1)$
$\boldsymbol{x}(t, i)$
$\bar{\boldsymbol{\alpha}}(t, i+1)$
$\boldsymbol{x}(t, i-1)$
$\bar{\boldsymbol{\alpha}}(t, i)$
τ_{i-1}
t_{i-1}^l
t_i^1
τ_i
t_i^2
t_i^l
t_{i+1}^1
τ_{i+1}

Figure 2-3: Illustration of the extended reference trajectory $\bar{\boldsymbol{\alpha}}$, with multiple simultaneous impacts and the state trajectory $\boldsymbol{x}$ with multiple non-simultaneous impacts.

To apply reference spreading for simultaneous impacts, the notion of multiscale hybrid time is introduced, through which each section of the trajectory can be uniquely defined [10]. Multiscale hybrid time is denoted with (t, i, k) in which i denotes the *macro* event counter and k denotes the *micro* event counter. The *micro* counter k starts at zero and is increased by one with every discrete event. The *macro* counter i is incremented by one when the system reaches the next mode, which resets the *micro* counter k to zero. The *micro* event times t_i^k are denoted as t_i^1, t_i^2,...,$t_i^{l_i}$, in which l_i denotes the total amount of *micro* events for *macro* event i. For ease of notation, $t_i^{l_i}$ is equal to t_{i+1}^0 and $(t, i, 0) = (t, i)$. This notation is also illustrated in Figure 2-3. Due to this mismatch in number of events, the state trajectory $\boldsymbol{x}$ enters a mode that is not defined for the reference trajectory $\boldsymbol{\alpha}$ for $t \in [t_i^1, t_i^{l_i}]$. For the control of tasks which contain simultaneous impacts, the authors of [10] propose to solely use the feedforward $\bar{\mathbf{u}}^{ff}(t,i)$ during the unspecified mode. Resulting in the control law given by

$$\mathbf{u}(\boldsymbol{x},t,i,k) = \begin{cases} \bar{\mathbf{u}}^{ff}(t,i) + \bar{\mathbf{K}}(t,i)(\bar{\boldsymbol{\alpha}}(t,i) - \boldsymbol{x}(t)), & t \in [t^0_{i+1}, t^1_{i+1}], \\ \bar{\mathbf{u}}^{ff}(t,i), & t \in \left(t^1_{i+1}, t^{l_i}_{i+1}\right). \end{cases} \tag{2-4}$$

The combination of the feedforward and the systems dynamics results in the stabilization of the contact. The extended reference trajectory, used by the controller, is in this work obtained through the use of a movement primitive. The chosen movement primitive, ProMP, is covered in the next section.

2-2 Probabilistic Movement Primitives

This section covers a method to translate demonstration data to a reference trajectory. This is done by using a movement primitive which, in this work, is ProMP [23]. The method fits a weighted sum of Radial Basis Functions (RBFs) on each demonstrated trajectory. It then combines these demonstrations by fitting a Gaussian distribution on the weights, resulting in a trajectory distribution. The RBF fits on demonstration data of two demonstrations and the resulting trajectory distribution is illustrated in Figure 2-4. Obtaining such a trajectory distribution from demonstration data is explained below.

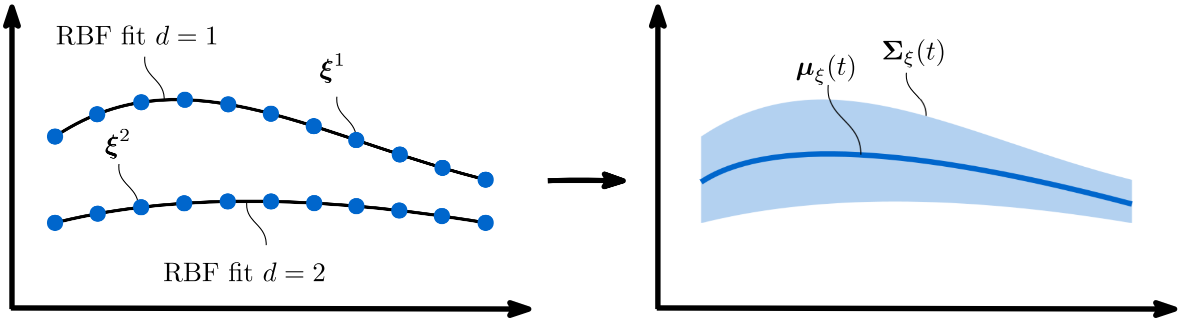


Figure 2-4: The RBF fits on the demonstration data $\boldsymbol{\xi}$ for two demonstrations (left) and the resulting trajectory distribution with mean trajectory $\boldsymbol{\mu}_\xi(t)$ and covariance $\boldsymbol{\Sigma}_\xi(t)$ (right).

The approximation of a Q dimensional trajectory through the use of Z RBFs can be written as

$$\boldsymbol{\xi}(t) = \boldsymbol{\psi}(t)\boldsymbol{w} + \boldsymbol{\epsilon}(t), \tag{2-5}$$

where $\boldsymbol{\xi}(t) \in \mathbb{R}^Q$ denotes the function to be approximated, $\boldsymbol{w} \in \mathbb{R}^{QZ}$ the vector of weights, and $\boldsymbol{\epsilon}(t) \in \mathbb{R}^Q$ the inaccuracy of the model. The matrix $\boldsymbol{\psi}(t) \in \mathbb{R}^{Q \times QZ}$ contains RBFs and is given by

$$\boldsymbol{\psi}(t) = \begin{bmatrix} \phi_1(t)\boldsymbol{I} & \phi_2(t)\boldsymbol{I} & \dots & \phi_Z(t)\boldsymbol{I} \end{bmatrix}, \tag{2-6}$$

where $\boldsymbol{I} \in \mathbb{R}^{Q \times Q}$ denotes the identity matrix and $\phi_v(t)$ the v-th RBF. The choice of the RBFs depends on the trajectory. The authors of [25] propose to use normalized Gaussian basis functions for stroke-based movement and Von-Mises basis functions for rhythmic movements. In this thesis we consider only stroke-based movement and thus the RBFs are defined as normalized Gaussians, formulated as

$$\phi_v(t) = \frac{b_v(t)}{\sum_{z=1}^{Z} b_z(t)}, \quad b_v(t) = \exp\left(-\frac{(t-c_v)^2}{2h}\right), \tag{2-7}$$

where h denotes the width parameter and c_v the center of radial basis function b_v. The centers of the radial basis functions are uniformly placed in the time domain of $\boldsymbol{\xi}(t)$. The RBF can be fit on L data points by the least squares estimate

$$\boldsymbol{w}^d = (\boldsymbol{\Psi}^T\boldsymbol{\Psi} + \lambda_{reg}\boldsymbol{I})^{-1}\boldsymbol{\Psi}^T\boldsymbol{\gamma}^d, \tag{2-8}$$

where λ_{reg} denotes the regularization parameter used to prevent overfitting, $\boldsymbol{\gamma}^d \in \mathbb{R}^{QL}$ is a vector containing all the measured data points $\boldsymbol{\xi}^d \in \mathbb{R}^Q$ of demonstration d and the RBF matrix $\boldsymbol{\Psi} \in \mathbb{R}^{QL \times QZ}$ contains the corresponding $\boldsymbol{\psi}(t)$ values. More specifically, the data vector and $\boldsymbol{\gamma}^d$ and RBF matrix $\boldsymbol{\Psi}$ are defined as

$$\boldsymbol{\gamma}^d = \begin{bmatrix} \boldsymbol{\xi}_0^d \\ \boldsymbol{\xi}_1^d \\ \vdots \\ \boldsymbol{\xi}_{L-1}^d \end{bmatrix}, \qquad \text{and} \qquad \boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\psi}(s_0) \\ \boldsymbol{\psi}(s_1) \\ \vdots \\ \boldsymbol{\psi}(s_{L-1}) \end{bmatrix}, \tag{2-9}$$

where $\boldsymbol{\xi}_a^d := \boldsymbol{\xi}(s_a^d)$ denotes the a-th measured data point and s_a^d the corresponding time value. The measured data points $\boldsymbol{\xi}_a$ can contain only joint positions $\mathbf{q}$ or, as in this work contain contact force data $\boldsymbol{\lambda}$ and input data $\mathbf{u}$ in addition. Which, assuming an equal sampling frequency can be denoted as $\boldsymbol{\xi} = [\mathbf{q}, \boldsymbol{\lambda}, \mathbf{u}]$. The result of this fitting process (2-8) is a weight vector $\boldsymbol{w}^d$ for each of the D demonstrations. To represent the trajectory, a Gaussian distribution is fit on the weight vectors. The mean of this Gaussian distribution is formulated as

$$\boldsymbol{\mu}_w = \frac{1}{D}\sum_{d=1}^{D} \boldsymbol{w}^d, \tag{2-10}$$

and the covariance matrix $\boldsymbol{\Sigma}_w$ as

$$\boldsymbol{\Sigma}_w = \frac{1}{D}\sum_{d=1}^{D}(\boldsymbol{w}^d - \boldsymbol{\mu}_w)(\boldsymbol{w}^d - \boldsymbol{\mu}_w)^T. \tag{2-11}$$

The weights and covariance matrix can be used to define a trajectory distribution [17], given by

$$\mathcal{N}(\boldsymbol{\mu}_\xi(t), \boldsymbol{\Sigma}_\xi(t)), \tag{2-12}$$

where $\boldsymbol{\mu}_\xi(t) \in \mathbb{R}^Q$ denotes the mean trajectory, formulated as

$$\boldsymbol{\mu}_\xi(t) = \boldsymbol{\psi}(t)\boldsymbol{\mu}_w, \tag{2-13}$$

and $\boldsymbol{\Sigma}_\xi \in \mathbb{R}^{Q\times Q}$ the covariance of the trajectory distribution, defined by

$$\boldsymbol{\Sigma}_\xi(t) = \boldsymbol{\psi}(t)\boldsymbol{\Sigma}_w\boldsymbol{\psi}(t)^T. \tag{2-14}$$

The mean trajectory $\boldsymbol{\mu}_\xi(t)$ can be used as a reference when the demonstration data contains join position $\mathbf{q}$. The derivative, given by

$$\dot{\boldsymbol{\mu}}_\xi(t) = \dot{\boldsymbol{\psi}}(t)\boldsymbol{\mu}_w, \tag{2-15}$$

can be used to define a velocity reference, in which $\dot{\boldsymbol{\psi}}(t)$ contains the derivative of the basis functions defined as

$$\dot{\phi}_v(t) = \frac{\dot{b}_v(t)\sum_{z=1}^{Z} b_z(t) - b_v(t)\sum_{z=1}^{Z}\dot{b}_z(t)}{(\sum_{z=1}^{Z} b_z(t))^2}, \quad \text{and} \quad \dot{b}_v(t) = -\frac{(t-c_v)}{h}\exp\left(-\frac{(t-c_v)^2}{2h}\right). \tag{2-16}$$

The position and velocity reference obtained through ProMP can be used for control. In this work, a QP controller is used for reference tracking. The QP controller is detailed in the next section.

2-3 Task-based QP robot control

This section explains QP control. Task-based QP robot control is an approach in which a control task is formulated as a quadratic optimization problem. This approach allows for a straightforward formulation of the tasks for complex systems such as humanoid robots [26, 27]. The control approach has been extended to be applied to multiple robots [28]. The control approach defines the weighted sum of the tasks as the cost function, whilst including equations of motion in the constraints, assuring that the solution holds true to the modelled dynamics. This work includes a simulation of a box being lifted from the ground by two end effectors. To do so, a constraint needs needs to be defined on the contact forces applied to the box such that it does not slip. The QP formulation allows for defining such constraints.

Using a similar notation as in [29], the cost function of the optimization problem is formulated as

$$\min_{\ddot{\mathbf{q}},\mathbf{u},\boldsymbol{\lambda}} \quad \sum_{m=1}^{M} w_m \left|\left|\ddot{g}_m^{des}(t) - \ddot{g}_m(t)\right|\right|^2, \tag{2-17}$$

where the joint acceleration $\ddot{\mathbf{q}}$, the input $\mathbf{u}$, and the contact forces $\boldsymbol{\lambda}$ are the free variables. The cost function (2-17) is quadratic in the free variables and can thus be solved using QP. The weights w define the relative importance between M tasks. The variable g_m denotes the task output and $\ddot{g}_m^{des}$ the desired task acceleration. The desired task acceleration includes a proportional and derivative feedback term, and is defined as

$$\ddot{g}_m^{des}(t) = \ddot{g}_m^{ref}(t) + k_m^p e_m(t) + k_m^d \dot{e}_m(t), \tag{2-18}$$

in which $\ddot{g}_m^{ref}$ denotes the reference acceleration, e_m the position error, $\dot{e}_m$ the velocity error,k_m^p the proportional gain, and k_m^d the derivative gain. The constraints of the QP problem can vary between tasks and can be subdivided into different categories.

Equations of motion. To assure that the solution of the QP problem holds true to the modelled dynamics, the equations of motion are included as equality constraints. Using the Euler-Lagrange formulation, the equations of motion are given by

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{g}(\mathbf{q}) + \mathbf{J}_c^T(\mathbf{q})\boldsymbol{\lambda} + \mathbf{S}\mathbf{u}, \tag{2-19}$$

where $\mathbf{M}(\mathbf{q})$ denotes the mass matrix, $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}})$ the Coriolis matrix, $\mathbf{g}$ the gravity vector, $\mathbf{J}_c^T(\mathbf{q})$ the Jacobian of the contact points and $\mathbf{S}$ the actuation matrix for underactuated systems. Equation (2-19) clearly shows that the equations of motion are linearly dependent on the free variables $\ddot{\mathbf{q}}$, $\mathbf{u}$, and $\boldsymbol{\lambda}$ and can thus be used as a equality constraint of the QP problem.

Kinematic loop. A kinematic loop [27] links the velocity of two points in the system. This can be formulated as

$$(\mathbf{J}_1(\mathbf{q}) - \mathbf{J}_2(\mathbf{q}))\dot{\mathbf{q}} = \mathbf{0}, \tag{2-20}$$

where $\mathbf{J}_1$ and $\mathbf{J}_2$ denote the Jacobians of the two points. However, this equation is not linearly dependent on the free variables of the QP problem. In this work, the acceleration of two points are linked, formulated as

$$(\mathbf{J}_1(\mathbf{q}) - \mathbf{J}_2(\mathbf{q}))\ddot{\mathbf{q}} + (\dot{\mathbf{J}}_1(\mathbf{q}) - \dot{\mathbf{J}}_2(\mathbf{q}))\dot{\mathbf{q}} = \mathbf{0}, \tag{2-21}$$

which is linearly dependent on the free variable $\ddot{\mathbf{q}}$ and can thus be used as an equality constraint of the QP problem.

Technological limits. The QP formulation allows for defining limits on the joint position, velocity, contact forces and input. This can be formulated as

$$\mathbf{q}_{min} \leq \mathbf{q} \leq \mathbf{q}_{max}, \tag{2-22}$$

$$\dot{\mathbf{q}}_{min} \leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_{max}, \tag{2-23}$$

$$\ddot{\mathbf{q}}_{min} \leq \ddot{\mathbf{q}} \leq \ddot{\mathbf{q}}_{max}, \tag{2-24}$$

$$\mathbf{u}_{min} \leq \mathbf{u} \leq \mathbf{u}_{max}, \tag{2-25}$$

$$\boldsymbol{\lambda}_{min} \leq \boldsymbol{\lambda} \leq \boldsymbol{\lambda}_{max}. \tag{2-26}$$

However, the position $\mathbf{q}$ and velocity $\dot{\mathbf{q}}$ are not linearly dependent on the free variables. The limits can be defined through an approximation, given by

$$\mathbf{q}_{min} \leq \mathbf{q} + \dot{\mathbf{q}}\Delta_{qp} + \ddot{\mathbf{q}}\frac{(\Delta_{qp})^2}{2} \leq \mathbf{q}_{max}, \tag{2-27}$$

$$\dot{\mathbf{q}}_{min} \leq \dot{\mathbf{q}} + \ddot{\mathbf{q}}\Delta_{qp} \leq \dot{\mathbf{q}}_{max}. \tag{2-28}$$

which are linearly dependent on the free variables.

Tangential and normal contact forces. To prevent the slipping of contact points, it is possible to define a relation between the tangential and normal contact forces as constraints. As we only cover the planar scenario and slipping in the numerical validation of this work, in the following only that case will be detailed. The reader is referred to [27, 29] for further reading on the constraints to be applicable for sliding and a 3D scenario. To prevent slipping, an equality should hold that defines a relation between the normal and tangential contact forces. Using the Coulomb friction model this inequality is given by

$$|\lambda_T| \leq \mu\lambda_N, \tag{2-29}$$

where λ_T denotes the tangential contact force, λ_N the normal contact force and μ the friction coefficient. The inequality (2-29) is equal to the following set of inequality constraints

$$\lambda_T \leq \mu\lambda_N, \tag{2-30}$$

$$-\lambda_T \leq \mu\lambda_N, \tag{2-31}$$

which are linearly dependent on the contact forces and can be used as inequality constraints of the QP controller.

This work uses a compliant contact model to define the contact forces in the simulations used to numerically validated the proposed method. The next section covers the used compliant contact model and the friction model.

2-4 Compliant contact and friction models

This section describes compliant contact and friction models. A simulation is done in this research to validate the proposed learning method. In this numerical validation, compliant contact models are used to define the normal contact forces at each contact. The friction forces are defined using friction models.

2-4-1 Compliant contact models

Compliant contact models [30], unlike non-smooth methods [31], allow a small indentation between two objects that make contact. The normal force is modeled to be a function of this small penetration, which is formulated as

$$f_{ccm} = f(\delta, \dot{\delta}), \tag{2-32}$$

where δ denotes the indentation. This function can have different forms, depending on the application for which it is used. The simplest compliant contact model is the Kelvin-Voigt model. The Kelvin-Voigt model is given by

$$f_{kv} = \begin{cases} c\dot{\delta} + k\delta, & \delta \geq 0, \\ 0, & \delta < 0, \end{cases} \tag{2-33}$$

where c and k denote the damping and stiffness coefficients. This model has the drawback that it results in an instantaneous jump in the normal force when impact is made, due to $\dot{\delta}$ being non-zero. The Hunt-Crossley model [32], does not have this deficiency as it models the normal force as

$$f_{hc} = \begin{cases} c\dot{\delta}\delta^n + k\delta^n, & \delta \geq 0, \\ 0, & \delta < 0, \end{cases} \tag{2-34}$$

which will result in a normal force of zero when $\delta = 0$. Although broadly used to model contact, the model has the drawback that it can result in a sticky force if an external force is applied. For example, in this work, the contact of a box between the floor is modeled using compliant contact models. Applying an external force, which results in the box being lifted from the ground, can result in $-c\dot{\delta}\delta^n > k\delta^n$. This inequality results in a negative normal force. The authors of [33] propose a solution through a new model referred to as the exponential contact model. The exponential contact model is used to model the normal force in the simulation of this research. The model is formulated as

$$f_{ccm} = \begin{cases} ke^{(c/k)\dot{\delta}}\delta^n & \delta \geq 0, \\ 0, & \delta < 0, \end{cases} \tag{2-35}$$

which, due to the exponential function, will not become negative. The Taylor series of (2-35) first order matches with (2-34). The damping coefficient k and the stiffness coefficients c are referred to as k_{ccm} and c_{ccm} in the remainder of this work.

2-4-2 Friction models

There are many ways to model friction [34]. Perhaps the most commonly used method is the Coulomb friction model [35]. However, this model has a jump in the friction force at zero velocity. To simplify, different models have been proposed which have a finite slope at zero velocity, and thus no discontinuity at zero velocity [34]. A simple model is the Coulomb friction model with a linear friction modification, defined as

$$f_w(v_T) = \lambda_N \mu_w \min(k_w|v_T|, 1) \operatorname{sign}(v_T), \tag{2-36}$$

where λ_N denotes the normal force, which in this work comes from the compliant contact model described in the previous subsection. The variable v_T denotes the relative tangential velocity, μ_w denotes the friction coefficient and k_w a variable that determines the angle of the slope at zero velocity. The force as a function of v_T is illustrated in Figure 2-5. This linear model is in this work referred to as the Coulomb friction model.

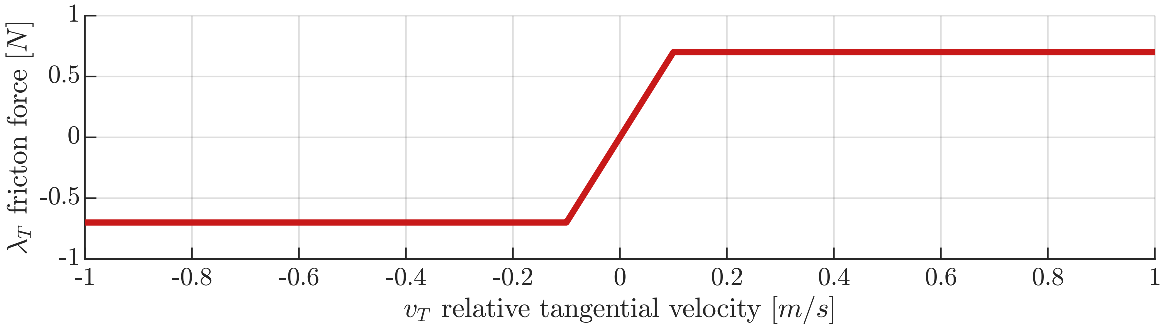


Figure 2-5: The friction force λ_T, plotted as a function of the relative tangential velocity v_T, with $\mu = 0.7$, $k_w = 10$, and $\lambda_N = 1$.

2-5 Summary

In this chapter the background information needed for this research has been explained. First, the control strategy for state triggered jumps named reference spreading has been covered. Secondly, the time driven movement primitive ProMP which translates demonstration data to a trajectory distribution has been presented. Thereafter, the QP controller, which formulates the control task as a minimization problem, has been detailed. Lastly the compliant contact and friction models have been explained. These topics are referred to in the next chapter which explains the proposed impact-aware learning method.

Chapter 3

Reference Spreading based Probabilistic Movement Primitives

This chapter describes the proposed method named Impact-Aware Learning from Demonstration (IA-LfD), which combines reference spreading and learning from demonstration. The main contribution of this thesis is that it proposes a method to translate demonstration data, of a task in which impact occurs, to an extended reference trajectory which can be used by the reference spreading control strategy covered in Section 2-1.

It is proposed to split the demonstration data into multiple modes segmented by events at which contact is established at non-zero speed. This is similar to the modes used by the reference spreading control strategy as is detailed in Section 2-1. It is then proposed to extend the demonstration data of each mode. The proposed method to segment and extend the demonstration data is covered in Section 3-1. This extended data can then be translated to extended reference trajectories through the use of a movement primitive. The chosen movement primitive in this thesis is Probabilistic Movement Primitive (ProMP), which is detailed in Section 2-2. Applying ProMP on the extended data of each mode is covered in Section 3-2. These extended references can then be tracked using varying controllers such a PD controller and Quadratic Programming (QP) controller in combination with the reference spreading error. These controllers in combination with the reference spreading error are covered in Section 3-3.

3-1 Data preconditioning for reference spreading

This section describes the preconditioning applied to the demonstration data. As the proposed method offers a solution to learning tasks which contain impact, it is assumed that the demonstration data is of such a task. As is explained in Section 2-1, reference spreading makes use of different modes in which the trajectories are defined. Furthermore, the reference trajectory $\boldsymbol{\alpha}(t)$ is extended about the time of impact. To define extended trajectories segmented into different modes using demonstration data, the data is preconditioned before applying a movement primitive such as ProMP. Preconditioning the data contains splitting the data into different modes and the extension of demonstration data.

3-1-1 Data segmentation into multiple modes

In this work, the reference spreading error (2-2), in which the current state is compared to a reference of the same mode (Section 2-1), is used for reference tracking. The reference trajectory $\bar{\boldsymbol{\alpha}}$ used in this error is defined for multiple modes, which are segmented through impact events. In this work, such a reference trajectory can be defined by a movement primitive. To do so, the demonstration data has to be segmented into different modes, such that movement primitives can be fitted on data in each mode. This subsection describes the notation of the segmented demonstration data for simultaneous impacts. This notation is an adaptation of the multiscale hybrid time notation of [10], explained in Section 2-1, such that it is applicable for discrete data of multiple demonstrations.

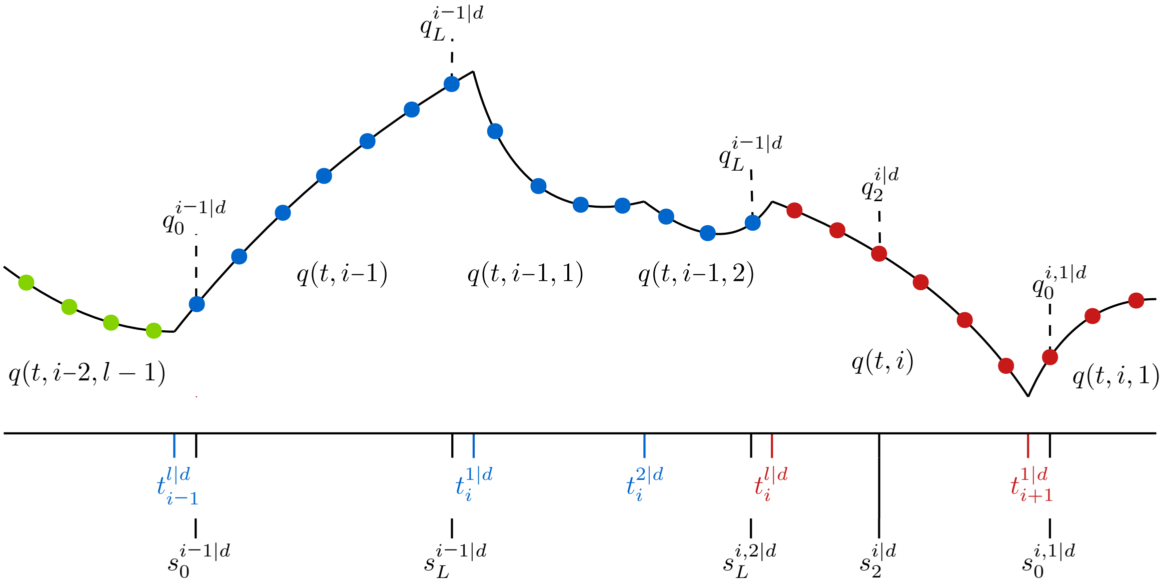


Figure 3-1: Demonstration data of one dimensional joint position q. The real joint position is depicted with the black line, whilst the sampling data points for $i-2$ (green), $i-1$ (blue), and i (red) are depicted by the colored dots.

As is described in Section 2-1, modes are divided by events. The times at which these events occur may vary between demonstrations. In order to describe these event times for different demonstrations, the index d has been added to the event time notation. The event time is denoted as $t_i^{k|d}$ in which i is the macro-event counter and k the micro-event counter as in the multiscale hybrid time notation. These event times have to be determined through impact detection. Detecting impact based on demonstration data is an interesting and important topic for further research. In this thesis, it is assumed that the impact event times are known for single impacts. For simultaneous impacts it is assumed that the first and last impact event times are known. The impact events split the data into different modes. The segmentation of the data into multiple modes is illustrated in Figure 3-1 for the simple case of one dimensional position data q. This simple example can be directly generalized for multidimensional demonstration data.

To mathematically formulate the extension of the data and the fitting of the movement primitives, each data point is labeled. The labeling of the demonstration is explained using the example of one dimensional position data q. One position data point is denoted as $q_a^{i,k|d} := q(s_a^{i,k|d}, i, k)$, belonging to a mode indicated with macro counter i and micro counter k. The counter $a \in \{0, 1, \dots, L^{i,k|d}\}$ is the discrete index of each data point belonging to that mode. The variable $L^{i,k|d}$ indicates the last data point of the mode of that demonstration. To improve readability, the superscripts of $L^{i,k|d}$ are left out in $q_L^{i,k|d}$. The variable $s_a^{i,k|d}$ denotes the time value of the discrete counter a for demonstration d and the mode with macro counter i and micro counter k. The discrete counter a and the time variable s are also used in the ProMP notation described in Section 2-2 and are illustrated in Figure 3-1. For ease of notation, the micro counter k is omitted when $k = 0$, resulting in $(\cdot)^{i,0|d} = (\cdot)^{i|d}$. This is similar to the multiscale hybrid time notation of [10] in which $(t, i, 0) = (t, i)$.

Although this example shows the notation for joint position data q, real demonstration data may in addition also contain contact force data $\boldsymbol{\lambda}$ and input data $\mathbf{u}$. Assuming an equal sampling frequency, this data will be denoted as $\boldsymbol{\xi}_a^{i,k|d} = [\mathbf{q}_a^{i,k|d}; \boldsymbol{\lambda}_a^{i,k|d}; \mathbf{u}_a^{i,k|d}] \in \mathbb{R}^{Q_{i,k}}$ in which $Q_{i,k}$ is the dimension of the used demonstration data belonging to the mode with macro event counter i and micro event counter k.

3-1-2 Extending the demonstration data

To apply the reference spreading error for reference tracking, the reference needs to be extended about its impact time. One could either extend the reference after applying a movement primitive on the data or extend the data and then fit a movement primitive. In this work, the latter approach is chosen because it provides a solution to ProMP specific issues which are the inaccuracy of the fit at the boundaries of the data and the data having varying time domains. These issues are explained in more detail in Section 3-2.

The data obtained through demonstrations is used to define a reference trajectory. Such a reference can represent a desired position and orientation, but it can also represent a desired contact force. The method of the extension depends on the type of data. In this section two types of data are discussed: position and orientation data and force data. The methods below are not the only applicable techniques for data extension.

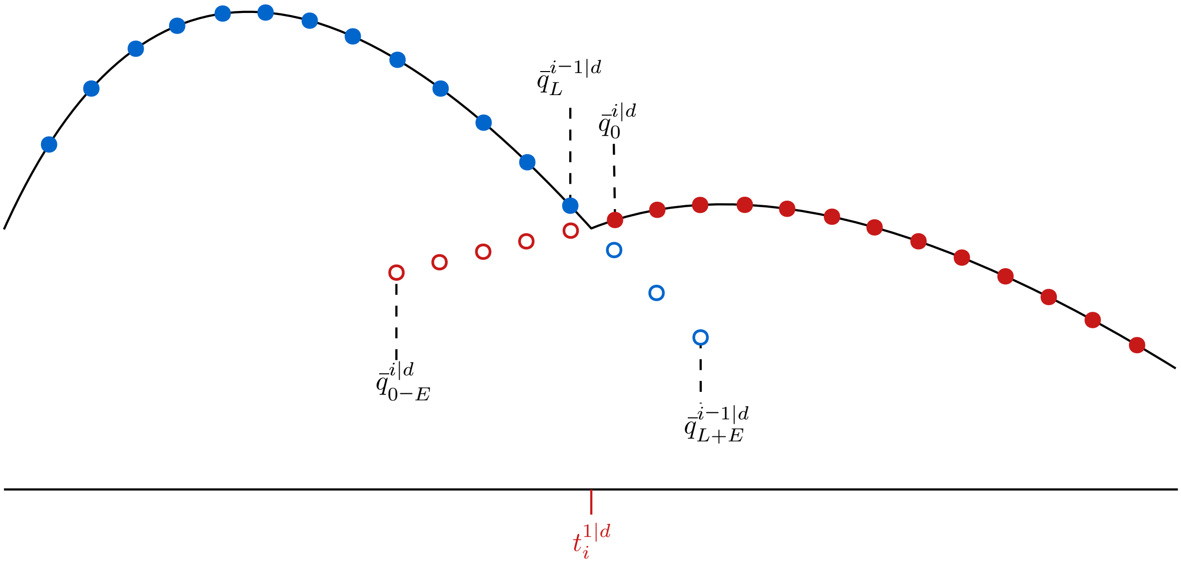


Figure 3-2: Extended demonstration data of one dimensional joint position $\bar{q}$. The demonstration data is presented by disks and the extended data by circles. The position data of mode $i-1$ and its extension is visualised in blue, whilst the data and its extension of mode i is visualised in red. For the sake of readability, $(\cdot)_{fw}$ and $(\cdot)_{bw}$ are not denoted.

Extending position and orientation data. In this work, it is proposed to extend the position and orientation data using a first order hold, as is illustrated in Figure 3-2. The position data is extended using a first order hold because of its simplicity and because it keeps a constant velocity. The constant velocity results in the system establishing contact with the same velocity as expected even if impact happens later in time. Extending the data using a zero order hold, thus keeping a constant position and orientation, may result in the system not establishing contact when tracking the learned trajectory. Extending the data by keeping a constant acceleration may result in an undesirably large velocity.

The time step between the extended data points is equal to the time step between the measured data points Δ_{sa}. This prevents the movement primitive from overfitting on the extended data if the least squares fitting method is used as in this work. To apply reference spreading, the trajectory needs to be defined for different modes about the event time. To do so, the data in each mode needs to be extended both forwards and backwards in time. For readability purposes, the indices $i, k|d$ are taken out of the equations. The backwards extended data can be defined as

$$\bar{\mathbf{q}}_a = \mathbf{q}_{a+1} - (\mathbf{q}_1 - \mathbf{q}_0) \quad \forall \quad a \in \{-E_{bw}, -E_{bw}+1, \ldots, -1\}, \tag{3-1}$$

in which E_{bw} is the (mode and demonstration specific) number of points by which the data is backwards extended. The forward extension can be written as

$$\bar{\mathbf{q}}_a = \mathbf{q}_{a-1} + (\mathbf{q}_L - \mathbf{q}_{L-1}) \quad \forall \quad a \in \{L+1, L+2, \ldots, L+E_{fw}\}, \tag{3-2}$$

in which E_{fw} is the (mode and demonstration specific) number of points by which the data is forwards extended.

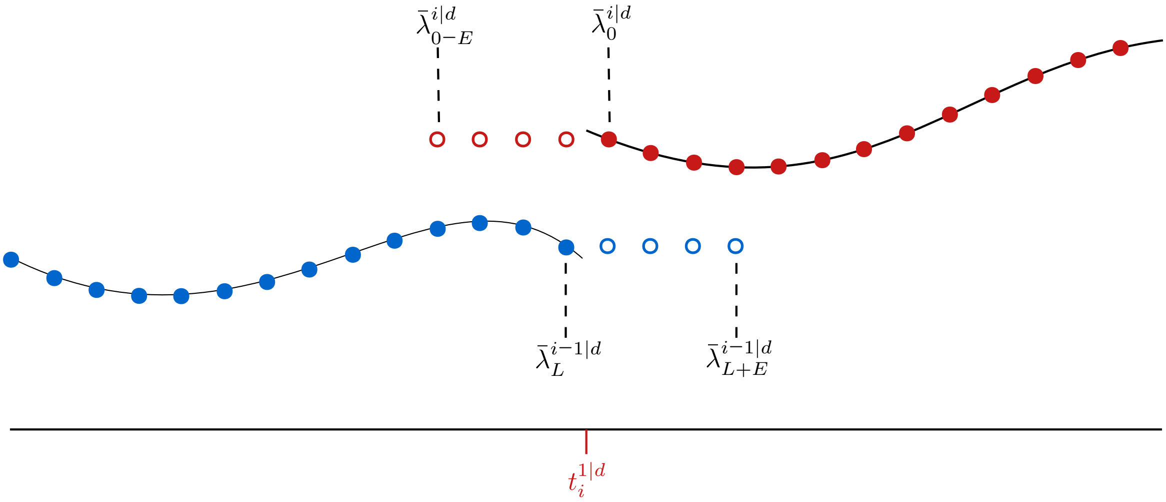

$\bar{\lambda}^{i|d}_{0-E}$
$\bar{\lambda}^{i|d}_{0}$
$\bar{\lambda}^{i-1|d}_{L}$
$\bar{\lambda}^{i-1|d}_{L+E}$
$t_i^{1|d}$

Figure 3-3: Extended demonstration data of contact force $\bar{\lambda}$. The demonstration data is presented by disks and the extended data by circles. The force data of mode $i-1$ and its extension is visualised in blue, whilst the data and its extension of mode i is visualised in red. For the sake of readability, $(\cdot)_{fw}$ and $(\cdot)_{bw}$ are not denoted.

Extending force data. Data obtained through demonstrations can also contain force data. For example, the contact forces between two objects can be measured during a demonstration. It is possible to use this demonstration data to define a reference for the contact forces. Unlike the extension of the position data, it is proposed to extend the force data using a zero order hold. This, to prevent undesired large forces. An example of extended force data is depicted in Figure 3-3. The backwards extension of the force can be formulated as

$$\bar{\boldsymbol{\lambda}}_a = \boldsymbol{\lambda}_0 \quad \forall \quad a \in \{-E_{bw}, -E_{bw}+1, \dots, -1\}. \tag{3-3}$$

The forward extension of the force can be written as

$$\bar{\boldsymbol{\lambda}}_a = \boldsymbol{\lambda}_L \quad \forall \quad a \in \{L+1, L+2, \dots, L+E_{fw}\}. \tag{3-4}$$

The input data in this work is regarded as force data and thus extended using the method explained above.

Extending data of near simultaneous impacts. During demonstrations, near simultaneous impacts, which are also mentioned in Section 2-1, may occur. For example, picking up a box with two end effectors where the end effectors establish contact with the box near simultaneously (Figure 3-4). Another example is putting a box down under a slight angle (which is more likely than not) in which the corners of the box will make near simultaneous impact with the floor. Due to the small time difference between these events, the sequence in which these events occur can differ between demonstrations. The modes between these quick successive events are referred to as the impact modes. Together, these impact modes form the impact phase, which starts with the first impact t_i^1 and ends with the last impact $t_i^{l_i}$ for macro counter i.

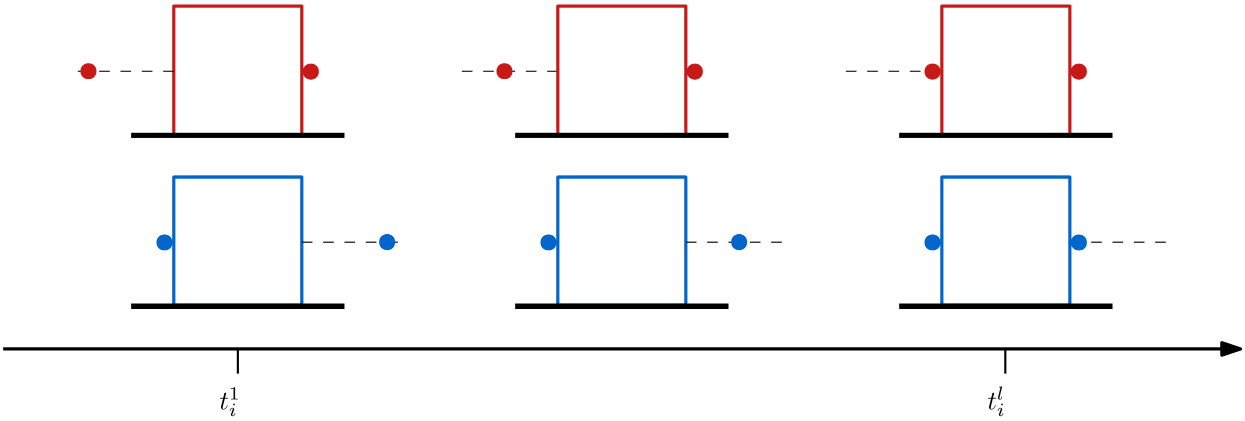


Figure 3-4: Two demonstrations in which two end effectors establish contact with a box near simultaneously. The sequence in which the contact is made varies between the two demonstrations, resulting in different demonstration data.

The data of the impact phase may be unreliable due to the successive impulsive forces applied to the system. Furthermore, due to the variation of the sequence in which these events take place, the data can vary significantly between demonstrations. For example, two end effectors making contact with a box near simultaneously as is depicted in Figure 3-4. The difference between which end effector is in free motion and which one is in contact with the box results in very different demonstration data.

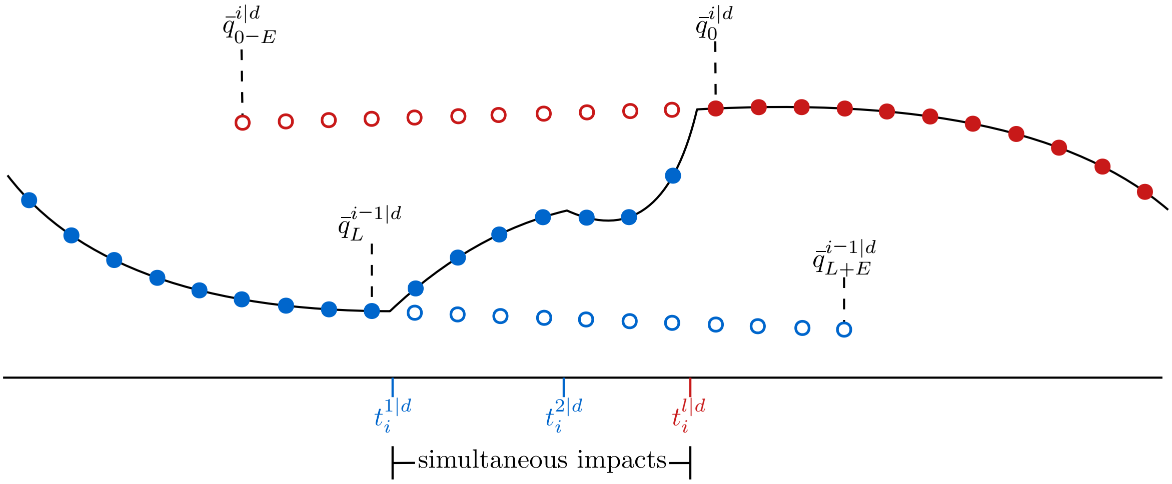


Figure 3-5: Extended demonstration data of joint position $\bar{q}$ with simultaneous impacts. The demonstration data is presented by disks and the extended data by circles. The position data of the mode $i-1$ and its extension is visualised in blue, whilst the data and its extension of mode i is visualised in red.

Due to the data of the impact phase being unreliable, a method of extension is proposed for near simultaneous impacts. Using this method, the data of the impact mode is excluded when a movement primitive is fitted. To overcome the time frame in which there are no data points to be used to learn a trajectory, the data from the modes alongside the impact mode is extended such that they both include the time frame of the impact phase. This is illustrated in Figure 3-5. There is no need to subdivide the impact phase into multiple impact modes as the data is not used in the method proposed in this work. The extended data is in this work used to fit an extended trajectory distribution through the use of ProMP. Applying ProMP on the extended demonstration data is covered in the next section.

3-2 Applying ProMP on extended demonstration data

After extending the segmented data of each mode, movement primitives are fit to translate the demonstration data into extended reference trajectories. It is proposed to fit a movement primitive on the extended data of the individual modes. In this work, the chosen movement primitive is ProMP (Section 2-2). However, any time driven movement primitive can be used for the purpose of translating the extended data to extended reference trajectories. This section first describes the requirements of the data extension such that ProMP can be applied. Thereafter, obtaining the trajectory distributions from demonstration data is covered.

3-2-1 Boundary issue

The movement primitive ProMP fits radial basis functions on the data of each demonstration to define the reference trajectory. A known issue of Radial Basis Function (RBF) approximation is that the approximation is relatively inaccurate near the boundaries if the values at the boundaries are not zero [36]. The behaviour is similar to, and often referred to as, the Runge [37] or Gibbs [36] phenomenon. As is depicted in Figure 3-6, and noted in [38], the approximation error is made prominent when taking the derivative of the radial basis functions.

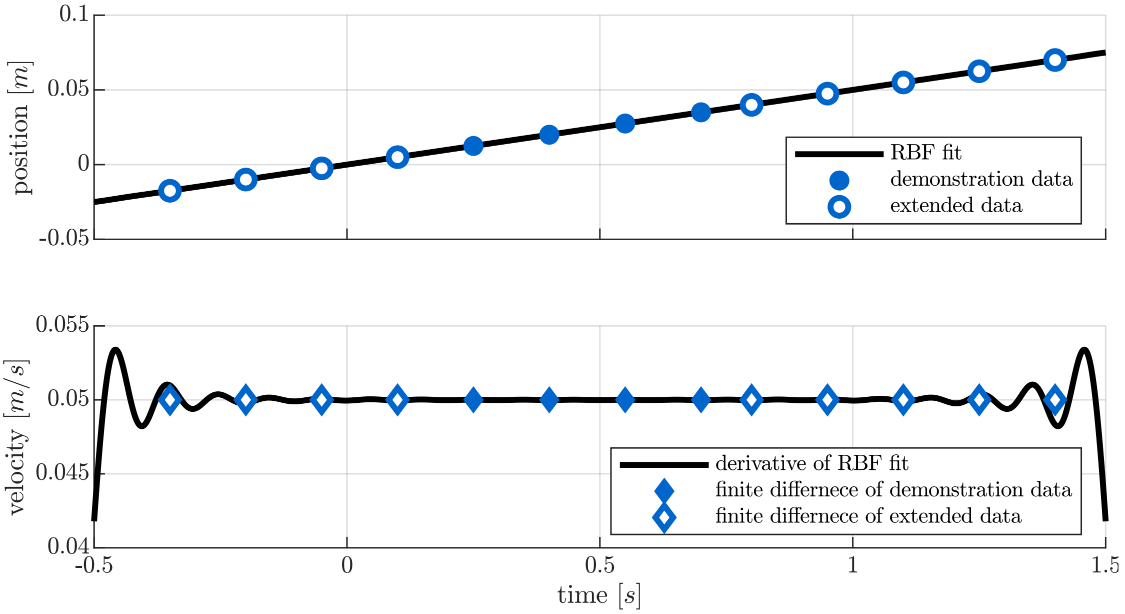


Figure 3-6: An illustration of the inaccuracy of the RBF fit near the boundaries of the data. The inaccuracy becomes most apparent for the velocity trajectory. The plots shown are the result of fitting 40 RBFs on 40 data points, where 14 are shown.

The movement primitive ProMP, uses the derivative of the RBF approximation to define a velocity reference (2-15). In this work, due to how the demonstration data is split, the data at the boundary represents the data near impact. Having an inaccurate fit on this data, and thus an undesired velocity reference when establishing contact, can result in unwanted behaviour of the system.

The authors of [23], suggest to place the centers of the radial basis function slightly outside the time domain of the demonstration data [25]. This is also suggested in [36] and [37]. Other suggestions are to use multi-quadric basis functions [38] or the addition of polynomial terms [38, 39]. The proposed method of this research is to further extend the demonstration data, ensuring that the boundary issue occurs outside the time domain in which the system is expected to be in the corresponding mode.

3-2-2 Time alignment

To apply ProMP on the data of multiple demonstrations, the time domain of the demonstrations should be of equal length. Usually, the demonstration trajectories provided by humans differ in duration. In this work, the issue of the difference in time domain is amplified by splitting the demonstration data at different event times between demonstrations. In the original paper on ProMP [23], this issue is resolved by using the phase variable. By using the phase variable, the time domain of each demonstration is normalized to be between zero and one. The authors of [40] use a slightly modified Dynamic Time Warping (DTW) method to align the demonstration data in time. The drawback of the phase variable and the DTW method, in the context of tasks in which contact is made at non-zero speed, is that they both alter the velocity to align the data. The drawback of altering the velocity, in regard to this work, is that it heavily influences the behaviour of the system upon establishing contact, which results in the system entering a sequential mode with an undesired velocity.

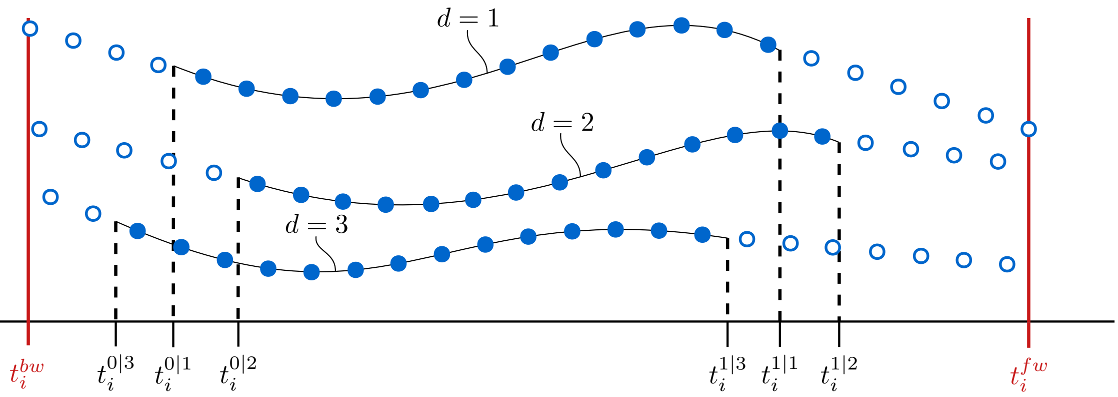


Figure 3-7: An illustration of the extended data of multiple demonstrations. The data is extended backwards up to t_i^{bw} and forwards up to t_i^{fw}. The instants of time up to which the data is extended are not demonstration dependent. The number of data points by which they are extended is demonstration dependent due to the varying event times.

In this research, we propose to extend the data until the same instants of time, for each demonstration. Thus resulting in extended data of each mode with a time domain that is about[1] equal between demonstrations. It is assumed that the demonstrations are aligned in time globally and that the variation in event times is minor compared to the length of the modes. Therefore, the data is only extended and not shifted in time. The extension of the data of multiple demonstrations up to the same instants of time is illustrated in Figure 3-7.

The time values t_i^{bw} and t_i^{fw} define the time until which the data of each demonstration is backwards and forwards extended respectively. These time values determine the time domain in which a reference trajectory is defined after applying ProMP. It is important that a reference trajectory is defined for the entire time the system is in the corresponding mode during tracking. This, plus the boundary issue discussed in Section 3-2-1 should be taken into account when defining the time values up to which the data is extended. The variation in event times during the demonstrations can be used to determine the uncertainty of the event time when tracking the reference. The higher this uncertainty, the further the data should be extended. However, in practise, there is little downside to extending the data further than the time domain the system is expected to be in the corresponding mode.

The number of points E needed to extend the data up to the defined time values varies between demonstration. This is due to the variation in event times and the sampling times not coinciding. For the backwards extension, the number of extended data points $E_{bw}^{i|d}$ is set such that

$$t_i^{bw} \leq s_{0-E}^{i|d} \leq t_i^{bw} + \Delta_{sa}, \tag{3-5}$$

where t_i^{bw} denotes the time value up to which the data of each demonstration is backwards extended, $s_{0-E}^{i|d}$ the time value of the data point corresponding to $-E_{bw}^{i|d}$, and Δ_{sa} the sampling time of the demonstration data. The subscript $(\cdot)_{bw}$ is removed from E in $s_E^{i|d}$ for the sake of readability. For the forwards extension, the number of extended data points $E_{fw}^{i|d}$ is set such that

$$t_i^{fw} - \Delta_{sa} \leq s_{L+E}^{i|d} \leq t_i^{fw}, \tag{3-6}$$

where t_i^{fw} denotes the time value up to which the data of each demonstration is forwards extended. Making the time domains of the data of equal length through the extension methods described in Section 3-1-2 does not alter the velocity of the data, unlike the phase variable and DTW. The extended demonstration data allows for learning an extended reference trajectory through the use of movement primitives.

3-2-3 Mode dependent movement primitives

After extending the segmented data of each mode, movement primitives are fit to translate the demonstration data into extended reference trajectories. It is proposed to fit a ProMP (Section 2-2) on the extended data of each of the individual modes. Fitting ProMP on the extended data of multiple demonstrations for different modes results in mode dependent trajectory distributions. An example of such a mode dependent trajectory distribution is illustrated in Figure 3-8.

[1] Maximum difference of $2\Delta_{sa}$

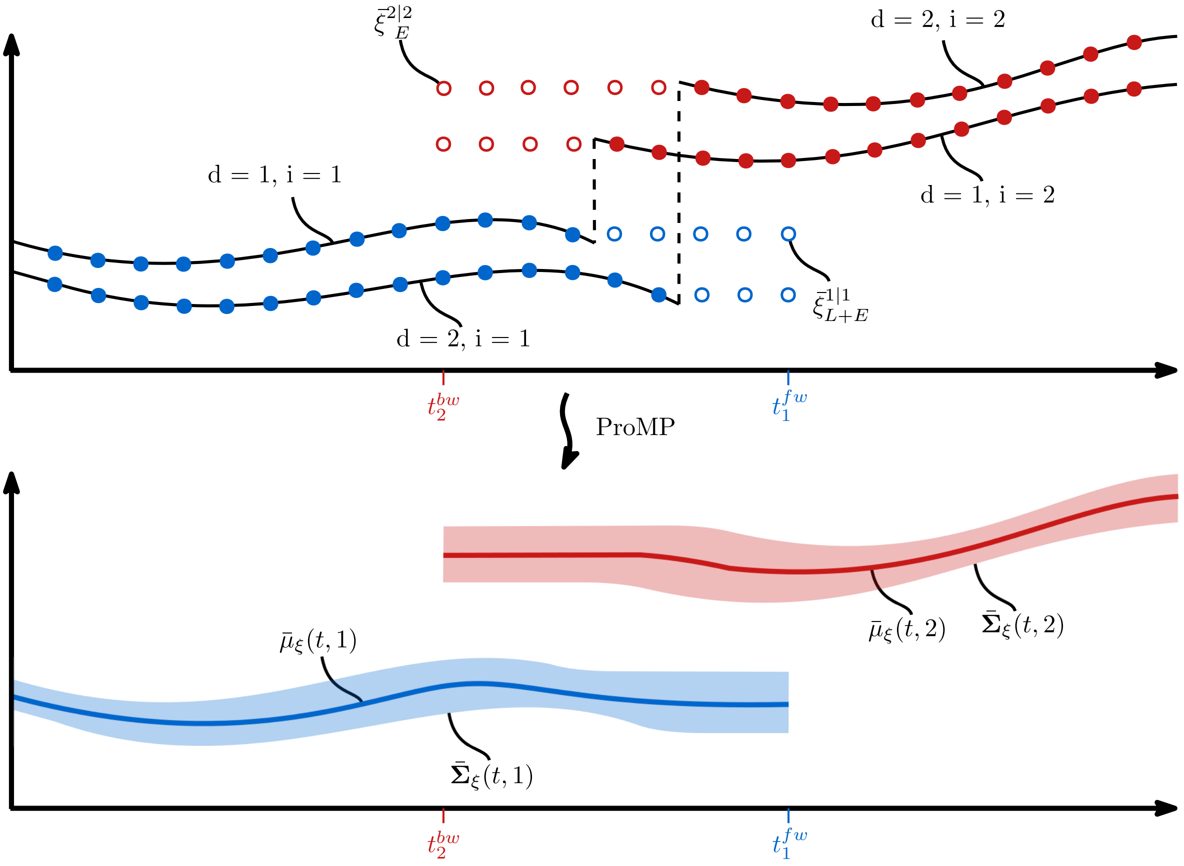


Figure 3-8: Figure illustrating the extended demonstration data $\bar{\boldsymbol{\xi}}$ of two demonstrations (top). The disks represent the original data while the circles represent the extended data. The bottom figure shows the resulting extended trajectory distributions, with mean $\bar{\boldsymbol{\mu}}_\xi(t,i)$ and covariance $\bar{\boldsymbol{\Sigma}}_\xi(t,i)$. The probability distributions and extended data of the mode $i=1$ are illustrated in blue while the probability distribution and extended data of mode $i=2$ are illustrated in red.

Obtaining these mode dependent trajectory distributions from extended demonstration data is explained below. As mentioned in Section 3-1-2, the movement primitives are only fit on the data of the first mode ($k=0$) for each macro event i. The vector containing the extended demonstration data is written as

$$\bar{\gamma}^{i|d} = \begin{bmatrix} \bar{\boldsymbol{\xi}}_{0-E}^{i|d} \\ \bar{\boldsymbol{\xi}}_{1-E}^{i|d} \\ \vdots \\ \bar{\boldsymbol{\xi}}_{L+E}^{i|d} \end{bmatrix}. \tag{3-7}$$

The centers of the radial basis functions are evenly spread over the time domain of the extended data in that mode. Which, as mentioned in Section 3-2-2 is equal to $\begin{bmatrix} t_i^{bw} & t_i^{fw} \end{bmatrix}$. The matrix $\boldsymbol{\psi}^i(t)$ (2-6) contains the RBFs of mode i. The matrix containing the values of $\boldsymbol{\psi}^i(t)$, at the time values of the extended data, is given by

$$\boldsymbol{\Psi}^i = \begin{bmatrix} \boldsymbol{\psi}(s_{0-E}^{i|d}) \\ \boldsymbol{\psi}(s_{1-E}^{i|d}) \\ \vdots \\ \boldsymbol{\psi}(s_{L+E}^{i|d}) \end{bmatrix}. \tag{3-8}$$

Using equations (2-8), (2-10), and (2-11), the mean $\boldsymbol{\mu}_w^i$ and the covariance $\boldsymbol{\Sigma}_w^i$ of the weights can be determined for each mode, which can then be used to define the extended trajectory distribution of each mode with their mean formulated as

$$\bar{\boldsymbol{\mu}}_\xi(t,i) = \boldsymbol{\psi}^i(t)\boldsymbol{\mu}_w^i, \tag{3-9}$$

and their covariance defined as

$$\bar{\boldsymbol{\Sigma}}_\xi(t,i) = \boldsymbol{\psi}^i(t)\boldsymbol{\Sigma}_w^i\boldsymbol{\psi}^i(t)^T. \tag{3-10}$$

The mean of the trajectory distribution can be used as the extended reference trajectory in the reference spreading error (2-2). The next section describes in further detail how the trajectory distribution can be used for control.

3-3 Reference spreading based control

This section covers how the extended trajectory distribution can be used for control. The obtained extended trajectory distributions allow for the use of the reference spreading error (2-2). In this work, two controllers are proposed that make use of the reference spreading error. These controllers allow for the execution of a task in which collision occurs at non-zero speed.

The first proposed controller is a simple PD controller with a feedforward term. This controller is covered to illustrate that input data can be used to define a feedforward. The second proposed controller is an adaption of the QP controller covered in Section 2-3, which is used in the numerical validation of this work. Both of these controllers make use of the reference spreading error.

To use the proposed controllers, the demonstration data should at least contain joint position data $\mathbf{q}$ and input data $\mathbf{u}$. Such a data point can be written as

$$\boldsymbol{\xi}_a^{i|d} = \begin{bmatrix} \mathbf{q}_a^{i|d} \\ \mathbf{u}_a^{i|d} \end{bmatrix}. \tag{3-11}$$

This discrete-time demonstration data, in combination with the proposed extension method (Section 3-1-2) and ProMP (Section 3-2) results in the continuous-time extended mean trajectory, formulated as

$$\bar{\boldsymbol{\mu}}_\xi(t,i) = \begin{bmatrix} \bar{\mathbf{q}}^{ref}(t,i) \\ \bar{\mathbf{u}}^{ff}(t,i) \end{bmatrix}, \tag{3-12}$$

where the extended mean trajectory of the joints $\bar{\mathbf{q}}^{ref}(t,i)$ can be used in conjunction with the reference spreading error and the extended mean trajectory of the input can be used as a feedforward $\bar{\mathbf{u}}^{ff}(t,i)$. The reference spreading error with the mean trajectory of the joints as a reference can be defined as

$$\boldsymbol{e}_{rs}(t,i) = \bar{\mathbf{q}}^{ref}(t,i) - \mathbf{q}(t,i). \tag{3-13}$$

To use the derivative of the references spreading error, given by

$$\dot{\boldsymbol{e}}_{rs}(t,i) = \dot{\bar{\mathbf{q}}}^{ref}(t,i) - \dot{\mathbf{q}}(t,i), \tag{3-14}$$

the derivative of the mean trajectory of the joint position $\dot{\bar{\mathbf{q}}}^{ref}(t,i)$ needs to be defined. This can be done using (2-15), resulting in

$$\dot{\boldsymbol{\mu}}_{\xi}(t,i) = \begin{bmatrix} \dot{\bar{\mathbf{q}}}^{ref}(t,i) \\ \dot{\bar{\mathbf{u}}}^{ff}(t,i) \end{bmatrix}, \tag{3-15}$$

where $\dot{\bar{\mathbf{u}}}^{ff}(t,i)$ denotes the derivative of the feedforward, which is not used in this thesis. The reference spreading error $\boldsymbol{e}_{rs}(t,i)$, its derivative $\dot{\boldsymbol{e}}_{rs}(t,i)$, and the feedforward $\bar{\mathbf{u}}^{ff}(t,i)$ are used for the proposed controllers.

As is explained in Section 2-1, the authors of [10] propose to solely use the feedforward during the unspecified mode. The unspecified mode is in this work referred to as the impact mode. In this research, it is proposed to use the feedforward learned from the extended input data of the mode previous to the impact mode. The control law during the impact mode is formulated as

$$\mathbf{u}(t,i,k) = \bar{\mathbf{u}}^{ff}(t,i) \quad \forall \quad 1 \leq k < l_i. \tag{3-16}$$

During the other modes, the input to the system can be defined through a PD or QP controller as is detailed below.

3-3-1 PD Controller with learned feedforward

It is possible to define a PD controller using the reference spreading error $\boldsymbol{e}_{rs}(t,i)$, its derivative $\dot{\boldsymbol{e}}_{rs}(t,i)$, and the feedforward $\bar{\mathbf{u}}^{ff}(t,i)$. The control law $\mathbf{u}(\mathbf{q},t,i)$ can then be written as

$$\mathbf{u}(\mathbf{q},t,i) = \bar{\mathbf{u}}^{ff}(t,i) + \mathbf{K}^{p}(t,i)\boldsymbol{e}_{rs}(t,i) + \mathbf{K}^{d}(t,i)\dot{\boldsymbol{e}}_{rs}(t,i), \tag{3-17}$$

where the matrices $\mathbf{K}^{p}(t,j)$ and $\mathbf{K}^{d}(t,j)$ denote the proportional and derivative gain, respectively. It is possible to learn the gains from demonstrations as is done in [41, 42, 43]. The main idea behind learning the controller gains is that they are related to the inverse of the variance of the demonstration data. Thus, resulting in higher controller gains in segments in which the demonstration data has a low variance and vice versa. The next section details the QP controller which is used in the simulation of this work.

3-3-2 QP controller combined with the reference spreading error

Generating robot motions which are robust to impact through the use of a QP controller, covered in Section 2-3, is an active and very recent research area. The authors of [44] propose to adapt the constraints of the QP controller, such that they remain feasible when impact occurs. In this work, it is proposed to use the QP controller in combination with the reference spreading error.

Using the reference spreading error, results in a different notation of the desired task acceleration (2-18). Due to the reference spreading error, this desired task acceleration becomes mode dependent. The mode dependent desired task acceleration can be written as

$$\ddot{\bar{g}}_m^{des}(t,i) = \ddot{\bar{g}}_m^{ref}(t,i) + k_m^p e_{rs,m}(t,i) + k_m^d \dot{e}_{rs,m}(t,i). \tag{3-18}$$

The reference spreading error for one task is formulated as

$$e_{rs,m}(t,i) = \bar{g}_m^{ref}(t,i) - g(t,i), \tag{3-19}$$

where the mode dependent task output $g(t,i)$ could for example be a joint position q and the task reference $\bar{g}_m^{ref}(t,i)$ can be the mean trajectory $\bar{q}^{ref}(t,i)$ learned through the proposed IA-LfD method. Other than the mode dependent desired task acceleration, it is also possible to define different tasks between modes. For example, for the scenario of two end effectors grabbing a box to then lift it up, the tasks before impact may define a desired position of the end effectors while the tasks after impact may define a desired position of the box. It is also possible that the weights of the tasks, controller gains, and constraints vary between modes. For the sake of readability we do not stray away from the QP notation covered in Section 2-3, apart from the mode dependent desired task acceleration. An example of a control objective for which the QP controller is used in combination with the reference spreading error is given in Section 4-3. It describes the QP controller used to grab a box with two end effectors simultaneously and lift it up to put it on top of a shelf. The QP controller used, is an example of how the tasks and constraints can vary between modes.

It is recommended to do further research on tuning the controller parameters based on the demonstration data, due to the high amount of controller parameters of the QP problem being increased by the addition of different modes.

3-4 Summary

This chapter covered the impact-aware learning strategy. First, the segmentation of the demonstration data into multiple modes is explained. Thereafter, the extension of the data is detailed. In this work, the position data is extended using a first order hold. The force data is extended using a zero order hold. The chapter also covered the alignment of the time domains of the data in each mode and the boundary issue related to fitting RBFs. After extending the data, ProMPs are fit on the extended data of each mode, resulting in extended trajectory distributions. The mean of the trajectory distributions can be used for control in combination with the reference spreading error, of which some examples are shown. The next chapter describes the numerical validation of the proposed impact-aware learning approach.

Chapter 4

Numerical validation of Impact-Aware Learning from Demonstration

This chapter describes how the Impact-Aware Learning from Demonstration (IA-LfD) method, covered in the previous chapter, is numerically validated by means of dynamical simulations. The simulation consists of two end effectors making contact at non-zero speed with a heavy box, to then swiftly lift it and put it on a shelf. This dynamic pick and place task is covered in detail in the first chapter. This task is to be demonstrated, to then be learned and executed, using the methods described in the previous chapter. The Quadratic Programming (QP) controller, combined with the reference spreading error, covered in Section 3-3-2, is used to generate the demonstration data as well as to track the learned trajectory. The implementation of the QP controller for the dynamic pick and place task is covered in Section 4-3. The demonstration data is generated by tracking manually defined reference trajectories. Generating the demonstration data is detailed in Section 4-4. The extension of the generated demonstration data, using the method detailed in Section 3-1-2, is described in Section 4-5. This section also covers the fitting of the Probabilistic Movement Primitives (ProMPs) on the extended demonstration data as is explained in Section 3-2. Lastly, the tracking results are reviewed.

4-1 The task of putting a box on a shelf

As stated above, the task consists of two end effectors making contact at non-zero speed with a box and subsequently putting it on top of a shelf. The challenge in this task is that contact is established at non-zero speed, with near simultaneous impacts at varying instances of time, both for the end effectors establishing contact with the box, and the box establishing contact with the shelf. The dynamic pick and place task is depicted in Figure 4-1.

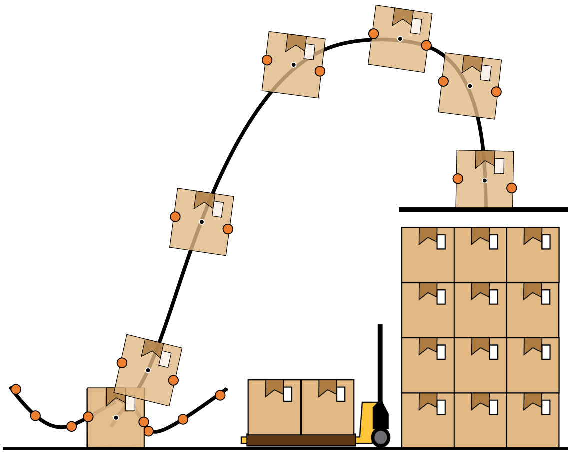


Figure 4-1: Snapshots of the dynamic pick and place task in which two end effectors near simultaneously establish contact with a box to then put it on top of a shelf.

The task is subdivided into different modes. The first mode contains the end effectors moving towards the box until one end effector establishes contact. The second mode is referred to as the first impact phase and starts when an end effector establishes contact with the box and ends when both end effectors have established contact. In the third mode, the box is lifted from the ground and put on top of a shelf, this mode ends when one of the corners of the box establishes contact with the shelf. The last mode, which is referred to as the second impact phase, starts with one corner of the box making contact and ends with both corners of the box establishing contact with the shelf. A timeline of the different modes and the events that separate them, is depicted in Figure 4-2.

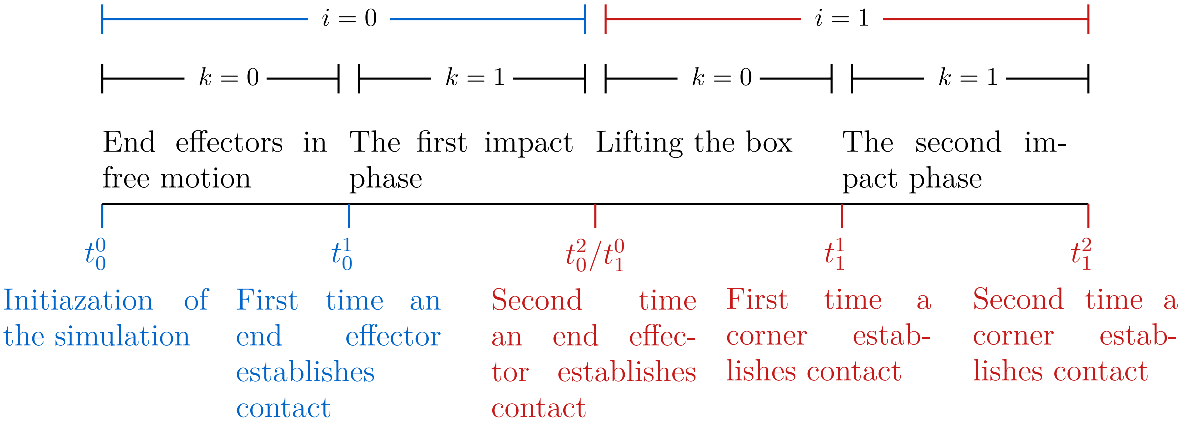

$i = 0$
$i = 1$
$k = 0$
$k = 1$
$k = 0$
$k = 1$
End effectors in free motion
The first impact phase
Lifting the box
The second impact phase
t_0^0
t_0^1
t_0^2/t_1^0
t_1^1
t_1^2
Initiazation of the simulation
First time an end effector establishes contact
Second time an end effector establishes contact
First time a corner establishes contact
Second time a corner establishes contact

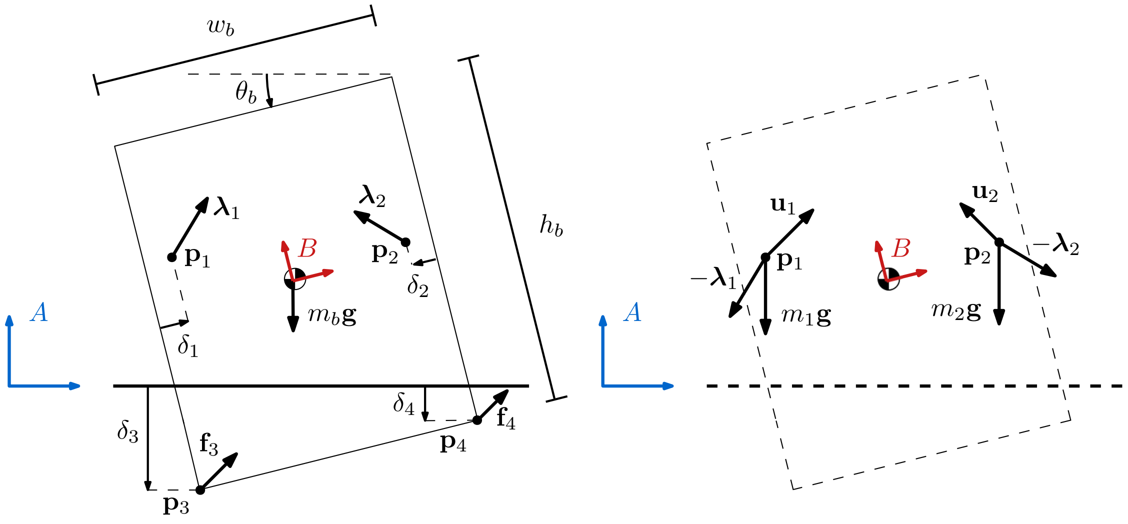
Figure 4-2: Timeline of the different modes of the dynamic pick and place task.

The event times vary between demonstrations and during tracking. Also, the sequence in which the micro events occur can vary. For example, in one demonstration the left corner of the box establishes in contact with the shelf first, while in the other demonstration the right corner establishes contact first. To simulate the dynamic pick and place task, a model is derived. The model is covered in the next section.

4-2 The dynamic model

A model is derived to simulate the dynamic pick and place task. The model describes the dynamics of the end effectors and the box, as well as the interaction of the box between the floor and shelf. This section details how the equations of motion are derived.

The Free Body Diagrams (FBDs) of the box and the effectors are drawn. The FBD of the box and the end effectors are depicted in Figure 4-3. The forces, illustrated in the FBDs, are listed in Table 4-1.



(a) FBD of the box. **(b)** FBD of the end effectors.

Figure 4-3: Free body diagram of the box (left) and the end effectors (right). It illustrates the world frame A and box frame B.

Table 4-1: An overview of the forces illustrated in the FBDs.

$\mathbf{u}_1, \mathbf{u}_2$	The input forces exerted on the end effectors.
$\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2$	The contact forces exerted by the end effectors on the box.
$\boldsymbol{\lambda}_3, \boldsymbol{\lambda}_4$	The contact forces exerted by the floor on the box.

The frames illustrated in the FBDs are the world frame A= $(\mathbf{o}_A, [A])$ and the frame attached to the box B= $(\mathbf{o}_B, [B])$. The origin $\mathbf{o}_A$ of the world frame A coincides with the floor. The origin $\mathbf{o}_B$ and the orientation $[B]$ of frame B are, respectively, equal to the Center of Gravity (CG) and the orientation of the box. The position of the end effectors are denoted by $\mathbf{p}_1$ and $\mathbf{p}_2$ as illustrated in Figure 4-3. The system contains a total of seven degrees of freedom, three for the box and two for each end effector. These are listed in Table 4-2. The box is placed on top of a shelf. The shelf is modeled as an elevated surface.

Table 4-2: The degrees of freedom of the end effectors and the box.

Box

x_b	The x coordinate of the CG of the box expressed in frame A.
y_b	The y coordinate of the CG of the box expressed in frame A.
θ_b	The angle of the box.

End effector 1

x_1	The x coordinate of end effector 1 expressed in frame A.
y_1	The y coordinate of end effector 1 expressed in frame A.

End effector 2

x_2	The x coordinate of end effector 2 expressed in frame A.
y_2	The y coordinate of end effector 2 expressed in frame A.

To derive the equations of motion, some simplifications are made. Firstly, it is assumed that only the bottom of the box comes in contact with the floor, the line contact is simplified to two points of contact. Secondly, the end effectors can only establish contact with the side of the box, not from above or from below. Lastly, the end effectors are assumed to never touch the floor and thus, no reaction forces of the floor on the end effectors are taken into account in the model.

For ease of notation, the cross product is defined for two dimensional vectors. For two vectors in $\mathbb{R}^2$: $\mathbf{a} = \begin{bmatrix} a_1 & a_2 \end{bmatrix}^T$ and $\mathbf{b} = \begin{bmatrix} b_1 & b_2 \end{bmatrix}^T$, this newly defined cross product is equal to $a_1b_2 - a_2b_1$. This can be written as

$$\times : \mathbb{R}^2 \to \mathbb{R}^1, \mathbf{a} \times \mathbf{b} := a_1b_2 - a_2a_1. \tag{4-1}$$

The equations of motion are formulated as

$$\begin{aligned}
\begin{bmatrix} m_b & 0 & 0 \\ 0 & m_b & 0 \\ 0 & 0 & I_b \end{bmatrix} \begin{bmatrix} \ddot{x}_b \\ \ddot{y}_b \\ \ddot{\theta}_b \end{bmatrix} &= \begin{bmatrix} \sum_{z=1}^{2} {}^A\mathbf{R}_B {}^B\boldsymbol{\lambda}_z + \sum_{z=3}^{4} {}^A\boldsymbol{\lambda}_z + m_b\mathbf{g} \\ \sum_{z=1}^{2} {}^B\mathbf{p}_z \times {}^B\boldsymbol{\lambda}_z + \sum_{z=3}^{4} ({}^B\mathbf{p}_z \times {}^B\mathbf{R}_A {}^A\boldsymbol{\lambda}_z) \end{bmatrix}, \\
\begin{bmatrix} m_1 & 0 \\ 0 & m_1 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \end{bmatrix} &= \begin{bmatrix} {}^A\mathbf{u}_1 - {}^A\mathbf{R}_B {}^B\boldsymbol{\lambda}_1 + m_1\mathbf{g} \end{bmatrix}, \\
\begin{bmatrix} m_2 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{x}_2 \\ \ddot{y}_2 \end{bmatrix} &= \begin{bmatrix} {}^A\mathbf{u}_2 - {}^A\mathbf{R}_B {}^B\boldsymbol{\lambda}_2 + m_2\mathbf{g} \end{bmatrix},
\end{aligned} \tag{4-2}$$

where $\mathbf{g} = \begin{bmatrix} 0 & -9.81 \end{bmatrix}^T$ denotes the gravity vector. The points $\mathbf{p}_z$, representing the points of contact of the reaction forces from both the floor and the end effectors on the box, are defined as $\mathbf{p}_z = \begin{bmatrix} p_{z,x} & p_{z,y} \end{bmatrix}^T$. The reaction forces of the end effectors are expressed in frame B as

$${}^B\boldsymbol{\lambda}_1 = \begin{bmatrix} \lambda_{N,1} \\ \lambda_{T,1} \end{bmatrix}, \qquad {}^B\boldsymbol{\lambda}_2 = \begin{bmatrix} -\lambda_{N,2} \\ \lambda_{T,2} \end{bmatrix}, \tag{4-3}$$

where $\lambda_{N,z}$ denotes the normal force at contact point z, modeled through the compliant contact model described in Section 2-4-1, formulated as

$$\lambda_{N,z} = f_{ccm}(\delta_z, \dot{\delta}_z). \tag{4-4}$$

The variable $\lambda_{T,z}$ is the friction force at contact point z defined by the Coulomb friction model described in Section 2-4-2, given by

$$\lambda_{T,z} = f_w(\lambda_{N,z}, {}^{B}\dot{p}_{z,y}). \tag{4-5}$$

The indentation variables δ_1 and δ_2 are illustrated in Figure 4-3a and defined as

$$\delta_1 = {}^{B}p_{1,x} + \frac{w}{2}, \tag{4-6}$$

$$\delta_2 = \frac{w}{2} - {}^{B}p_{2,x}. \tag{4-7}$$

The reaction forces of the floor on the box are defined in frame A by

$$^{A}\boldsymbol{\lambda}_3 = \begin{bmatrix} -\lambda_{T,3} \\ \lambda_{N,3} \end{bmatrix}, \qquad {}^{A}\boldsymbol{\lambda}_4 = \begin{bmatrix} -\lambda_{T,4} \\ \lambda_{N,4} \end{bmatrix}, \tag{4-8}$$

where the normal forces $\lambda_{N,3}$ and $\lambda_{N,4}$ are also defined through the compliant contact model as in (4-4) and the friction forces $\lambda_{T,3}$ and $\lambda_{N,4}$ are defined through the Coulomb friction model as in (4-5). The corresponding variables δ_3 and δ_4 are the indentations of the corners of the box with the floor or shelf, which are defined as

$$\delta_3 = \begin{cases} -{}^{A}p_{3,y} + y_{shelf}, & {}^{A}p_{3,x} \geq x_{shelf}, \\ -{}^{A}p_{3,y}, & {}^{A}p_{3,x} < x_{shelf}, \end{cases} \tag{4-9}$$

$$\delta_4 = \begin{cases} -{}^{A}p_{4,y} + y_{shelf}, & {}^{A}p_{4,x} \geq x_{shelf}, \\ -{}^{A}p_{4,y}, & {}^{A}p_{4,x} < x_{shelf}, \end{cases} \tag{4-10}$$

where x_{shelf} denotes the x value of the left point of the shelf and y_{shelf} denotes the height of the shelf. The rotation matrix ${}^{B}\mathbf{R}_A$ is defined as

$$^{B}\mathbf{R}_A = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{4-11}$$

The model parameters and their values used during the simulation in this work are denoted in Appendix B. For the simulation, the dynamical model is integrated using `ode15s` in MATLAB with time step $\Delta_{ode} = 1e-3$. The input $\mathbf{u}$ is defined through the use of a QP controller. The QP controller is covered in more detail in the next section.

4-3 QP control for lifting a box

The QP controller, combined with the reference spreading error, as is detailed in Section 2-3, is used to perform the dynamic pick and place task. The formulated QP problems are solved using `quadprog` in MATLAB. As mentioned in Section 4-1, the task is split into separate modes. The QP controller is covered for each mode below.

4-3-1 End effectors in free motion

The objective in the mode, in which the end effectors are in free motion, is to move the end effectors towards the box. Therefore, the tasks of the QP controller denote the desired accelerations of the two end effectors. The cost function of the QP controller can be written as

$$\min_{\ddot{\mathbf{q}}_{ee},\mathbf{u}} \quad \sum_{m=1}^{4} w_m \left|\left|\ddot{g}_m^{des}(t,0) - \ddot{g}_m(t)\right|\right|^2, \tag{4-12}$$

where g_1, g_2, g_3, and g_4 denote the position of the end effectors x_1, y_1, x_2, and y_2 respectively. The free variables of the optimization problem are the acceleration of the end effectors $\ddot{\mathbf{q}}_{ee} \in \mathbb{R}^4$ and the input $\mathbf{u} \in \mathbb{R}^4$. The contact forces are not included in the free variables as the end effectors are in free motion. The equality constraints are the equations of motion of the end effectors in free motion and can be formulated as

$$\mathbf{M}\ddot{\mathbf{q}}_{ee} = \mathbf{g} + \mathbf{u}, \tag{4-13}$$

where $\mathbf{M} = \text{diag}(m_1, m_1, m_2, m_2)$ denotes the mass matrix, $\ddot{\mathbf{q}}_{ee} = [\ddot{x}_1, \ddot{y}_1, \ddot{x}_2, \ddot{y}_2]^T$ the acceleration vector of the end effectors, $\mathbf{g} \in \mathbb{R}^4$ a gravity vector, and $\mathbf{u} \in \mathbb{R}^4$ the input.

4-3-2 The first impact phase

The first impact phase is the mode between the first time t_0^1 at which an end effector establishes contact and the time t_0^2 at both end effectors have established contact with the box. The control law of this mode is a feedforward controller, defined as

$$\mathbf{u}(t,0,1) = \bar{\mathbf{u}}^{ff}(t,0), \tag{4-14}$$

where $\bar{\mathbf{u}}^{ff}(t,0)$ denotes the feedforward of the previous mode, in which both end effectors are in free motion, moving towards the box.

4-3-3 Lifting the box

The objective in the third mode is to lift the box from the ground and put it on top of the shelf. The mode is initiated, when both end effectors have established contact with the box, at t_0^2, and ends when one of the corners establishes contact with the shelf, at t_1^1. To achieve the objective of lifting the box and placing it on top of a shelf a task is defined of a desired position and orientation of the box. These tasks result in the cost function of the QP problem given by

$$\min_{\ddot{\mathbf{q}}_b,\mathbf{u},\boldsymbol{\lambda}} \quad \sum_{m=1}^{3} w_m \left|\left|\ddot{\bar{g}}_m^{des}(t,1) - \ddot{g}_m(t)\right|\right|^2, \tag{4-15}$$

where g_1, g_2, and g_3 denote the position of the box x_b, y_b, and the orientation of the box θ_b, respectively. The free variables of the optimization problem are the position and angular acceleration of the box $\ddot{\mathbf{q}}_b \in \mathbb{R}^3$, the input $\mathbf{u} \in \mathbb{R}^4$ and the contact forces of the end effectors on the box $\boldsymbol{\lambda} \in \mathbb{R}^4$. However, the cost function (4-15) does not have a unique solution, even with the added constraints mentioned later in this subsection. The minimum value of the cost function is 0, which is true for $\ddot{\bar{g}}_m^{des}(t,1) = \ddot{g}_m(t) \quad \forall \quad m \in \{1,2,3\}$. Using the equations of motion, this equality can be written as a system of equations, given by

$$\begin{bmatrix} \ddot{\bar{g}}_1^{des}(t,1) \\ \ddot{\bar{g}}_2^{des}(t,1) \\ \ddot{\bar{g}}_3^{des}(t,1) \end{bmatrix} = \mathbf{M}_b^{-1}(\mathbf{g} + \mathbf{J}_b^T(\mathbf{q_b})\boldsymbol{\lambda}, \tag{4-16}$$

where $\mathbf{M}_b = \text{diag}(m_b, m_b, I_b)$ denotes the mass matrix of the box, $\mathbf{g} \in \mathbb{R}^3$ the gravity vector, and $J_b(\mathbf{q_b}) \in \mathbb{R}^{4\times 3}$ the Jacobian of the contact points with respect to the box. This system of three equations has four unknowns $\boldsymbol{\lambda} \in \mathbb{R}^4$ and thus, has infinitely many solutions. To understand this, the reader is kindly asked to lift, and hold still, an object with two fingers (i.e. a water bottle). Applying a higher amount of force on the object with the fingers does not necessarily cause it to move, as the forces applied to the object can cancel each other out. To assure that the system of equations is determined, additional tasks are added. These tasks aim to minimize the difference between the normal forces exerted by the end effectors and their respective extended references. The cost function of the QP problem can then be written as

$$\min_{\ddot{\mathbf{q}},\mathbf{u},\boldsymbol{\lambda}} \quad \sum_{m=1}^{3} w_m^{pos} \left|\left|\ddot{\bar{g}}_m^{des}(t,1) - \ddot{g}_m(t)\right|\right|^2 + \sum_{z=1}^{2} w_z^{force} \left|\left|\bar{\lambda}_{n,z}^{ref}(t,1) - \lambda_{N,z}(t)\right|\right|^2, \tag{4-17}$$

where the weights w_m^{pos} and w_z^{force} denote the relative importance between the position, orientation, and the force tasks. The extended reference of the normal force exerted on the box by end effector z is denoted as $\bar{\lambda}_{N,z}^{ref}(t,1)$.

The equations of motion. The first equality constraints, of the optimization problem, are the equations of motion. The equations of motion used in the QP controller vary from the equations of motion of the simulation. The contact forces between the floor and the box are not known and are excluded from the equations of motion. The first equality constraint of the QP controller is formulated as

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{g} + \mathbf{J}_c^T(\mathbf{q})\boldsymbol{\lambda} + \mathbf{S}\mathbf{u}, \tag{4-18}$$

which includes the mass matrix $\mathbf{M} \in \mathbb{R}^{7\times7}$, the acceleration vector of both the box and the end effectors $\ddot{\mathbf{q}} = [\ddot{x}_b, \ddot{y}_b, \ddot{\theta}_b, \ddot{x}_1, \ddot{y}_1, \ddot{x}_2, \ddot{y}_2]^T$, a gravity vector $\mathbf{g} \in \mathbb{R}^7$, the contact Jacobian $\mathbf{J}_c(\mathbf{q}) \in \mathbb{R}^{4\times7}$, the contact forces of the end effectors on the box $\boldsymbol{\lambda} \in \mathbb{R}^4$, the actuation matrix $\mathbf{S} \in \mathbb{R}^{7\times4}$, and the input $\mathbf{u} \in \mathbb{R}^4$.

Kinematic loops. The next equality constraints of the QP problem are the kinematic loops. The kinematic loops determine the acceleration of the end effectors, given a desired acceleration of the box. As explained in Section 2-3, kinematic loops link the acceleration of two points in the system. Which in this mode, is the location of the end effectors and their initial location on the box. As there are two end effectors, the kinematic loops are given by

$$(\mathbf{J}_{z1}(\mathbf{q}) - \mathbf{J}_{z2}(\mathbf{q}))\ddot{\mathbf{q}} + (\dot{\mathbf{J}}_{z1}(\mathbf{q}) - \dot{\mathbf{J}}_{z2}(\mathbf{q}))\dot{\mathbf{q}} = \mathbf{0} \quad \forall \quad z \in \{1, 2\}, \tag{4-19}$$

where $\mathbf{J}_{z1} \in \mathbb{R}^{2\times7}$ $\forall z$ denotes the Jacobian of the contact point of end effector z on the box and $\mathbf{J}_{z2} \in \mathbb{R}^{2\times7}$ $\forall z$ denotes the Jacobian of end effector z itself, which are straightforward as the position of the end effectors are part of the generalized coordinates $\mathbf{q}$. The Jacobians that link the two points in the system, are related to the contact Jacobian, due to the two points being equal to the points at which the contact forces are applied. This relation is given by

$$\begin{bmatrix} \mathbf{J}_{11}(\mathbf{q}) - \mathbf{J}_{12}(\mathbf{q}) \\ \mathbf{J}_{21}(\mathbf{q}) - \mathbf{J}_{22}(\mathbf{q}) \end{bmatrix} = \mathbf{J}_c(\mathbf{q}). \tag{4-20}$$

The contact forces. To be able to lift the box and prevent slipping between the end effectors and the box, inequality constraints are defined. These inequality constraints are similar to the constraints defined in Section 2-3, with the addition of a boundary on the normal forces for robustness. The first inequality constraints define the relation between the tangential and normal contact forces and are given by

$$\begin{aligned} \lambda_{T,1} &\leq \mu\lambda_{N,1}, \\ -\lambda_{T,1} &\leq \mu\lambda_{N,1}, \\ \lambda_{T,2} &\leq \mu\lambda_{N,2}, \\ -\lambda_{T,2} &\leq \mu\lambda_{N,2}. \end{aligned} \tag{4-21}$$

To assure that the end effectors stay in contact, the normal forces need to stay above a certain boundary. These inequality constraints are formulated as

$$\begin{aligned} \lambda_N^{min} f_N^{step}(t) &\leq \lambda_{N,1}, \\ \lambda_N^{min} f_N^{step}(t) &\leq \lambda_{N,2}, \end{aligned} \tag{4-22}$$

in which λ_N^{min} denotes the desired minimum of the normal force and $f_N^{step}(t)$ a smooth step function. This smooth step function is used, such that the inequality constraint is not violated when the system enters this mode.

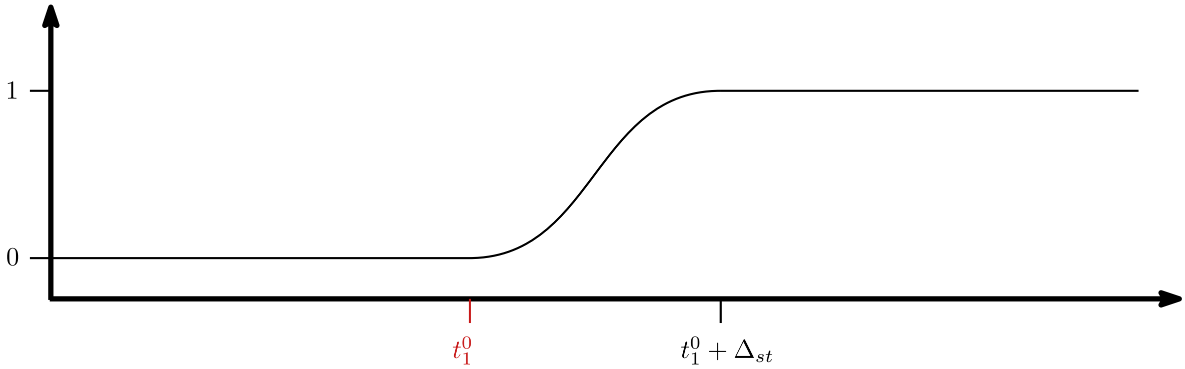


Figure 4-4: Figure illustrating a smooth step function used as a boundary on the normal forces in the QP controller. The instant t_1^0 denotes the event time at which both end effectors have established contact with the box and Δ_{st} a small time step.

The smooth step function is illustrated in Figure 4-4 and formulated as

$$f_n^{step}(t) = \begin{cases} 0, & t \leq t_1^0, \\ 3((t-t_1^0)/\Delta_{st})^2 - 2((t-t_1^0)/\Delta_{st})^3, & t_1^0 < t \leq t_1^0 + \Delta_{st}, \\ 1, & t > t_1^0 + \Delta_{st}, \end{cases} \tag{4-23}$$

where Δ_{st} denotes the small time step that determines the slope of the function. To summarize, the QP problem of the third mode aims at minimizing the cost function (4-17) subject to (4-18), (4-19), (4-21), and (4-22).

4-3-4 The second impact phase

The last mode, named the second impact phase, is initiated at the time t_0^2 at which one of the corners of the box establishes contact with the shelf and ends when both of the corners have established contact with the shelf at t_1^1. The control law of this mode, is the extended feedforward of the previous mode, in which the box is lifted from the ground. Written as

$$\mathbf{u}(t,1,1) = \bar{\mathbf{u}}^{ff}(t,1), \tag{4-24}$$

where $\bar{\mathbf{u}}^{ff}(t,1)$ denotes the extended feedforward of the mode in which the box is lifted. The simulation is ended when both corners of the box have established contact with the shelf.

4-3-5 Switching between modes

To switch between modes, an impact detection tool is needed. During the simulation, contact is straightforwardly detected by observing whether the indentation values $\delta_1, \delta_2, \delta_3$, or δ_4

become positive. As we are using soft-contact models, the impacts are not instantaneous. This requires different definitions of the events.

In this work, the beginning of the first near simultaneous impact, is defined as the event that triggers the mode switch for going from an non-impact mode to an impact mode. The ending of the last near simultaneous impact is defined as the event that triggers the mode switch for going from an impact mode to a non-impact mode. This definition results in the impacts being included in the impact modes for their full duration. Each event, or transition to the next mode, is described separately below.

Transition from end effectors in free motion to the first impact phase. The transition from the first mode to the second mode, takes place with the first event at t_0^1, which is when one of the end effectors establishes contact with the box. The event is enabled when one of the variables that denote the indentation of the end effectors and the box becomes positive, which is written as

$$\delta_1 > 0 \vee \delta_2 > 0. \tag{4-25}$$

Transition from first impact phase to lifting the box. The transition of the second mode to the third mode takes place with the event at t_0^2, which is at the end of the impact caused by the second end effector establishing contact with the box. The event is enabled shortly after both of the variables denoting the indentation between the end effectors and the box become positive. This is defined as

$$\delta_1 > 0 \wedge \delta_2 > 0 \wedge t > t_{init,0}^2 + \Delta_{im}, \tag{4-26}$$

where $t_{init,0}^2$ denotes the instant of time at which $\delta_1 > 0 \wedge \delta_2 > 0$ and Δ_{im} denotes a small time step added such that the transition to the next mode takes place at the end of impact.

Transition from lifting the box to the second impact phase. The transition from the second mode to the third mode takes place with the event at t_1^1, which is when one of the corners of the box establishes contact with the shelf. This event is enabled when one of the variables indicating the indentation between the corners of box and the shelf becomes positive. This can be written as

$$\delta_3 > 0 \vee \delta_4 > 0. \tag{4-27}$$

The end of the second impact phase. The simulation ends with the last event at t_1^2, which is at the end of the impact caused by the second corner establishing contact with the shelf. This event is enabled shortly after both variables denoting the indentation between the box and the shelf become positive. This is defined as

$$\delta_3 > 0 \wedge \delta_4 > 0 \wedge t > t_{init,1}^2 + \Delta_{im}, \tag{4-28}$$

where $t_{init,1}^2$ denotes the initial time at which $\delta_3 > 0 \wedge \delta_4 > 0$. Again, Δ_{im} is added such that the impact is included in the impact phase for its full duration.

The QP controller covered in this section, is used to generate demonstration data as well as tracking the learned extended trajectory. Generating the demonstration data is detailed in the next section.

4-4 Generating demonstration data

This section covers generating the demonstration data. To generate demonstration data, multiple reference trajectories are manually defined. The defined trajectories are tracked using the QP controller described in the previous section. The tracking results are then used as the demonstration data. In this thesis, the number of demonstrations D is set to five.

To use the proposed QP controller, reference trajectories need to be defined for each task. In this work, these references are defined through the use of multiple quintic polynomials [45]. A quintic polynomial to define a reference trajectory for joint position q is formulated as

$$\bar{q}^{ref}(t) = c_0^{qu} + c_1^{qu}t + c_2^{qu}t^2 + c_3^{qu}t^3 + c_4^{qu}t^4 + c_5^{qu}t^5. \tag{4-29}$$

The constants c^{qu} can be analytically derived when desired initial and final positions, initial and final velocities, and initial and final accelerations are defined. The reference trajectory of the velocity $\dot{\bar{q}}^{ref}(t)$ and the acceleration $\ddot{\bar{q}}^{ref}(t)$, both used in the desired task acceleration of the QP controller (3-18), can be obtained by taking the second and third derivative of (4-29) respectively. As quintic polynomials allow for the definition of desired initial and final positions, initial and final velocities, and initial and final accelerations, a $\mathcal{C}^3$ smooth trajectory can be straightforwardly defined by multiple sequential polynomials. To do so, the final position velocity, and acceleration of the n^{qu}-th polynomial are set equal to the initial position, velocity, and acceleration of the $(n^{qu}+1)$-th polynomial. These desired points are further referred to as via points.

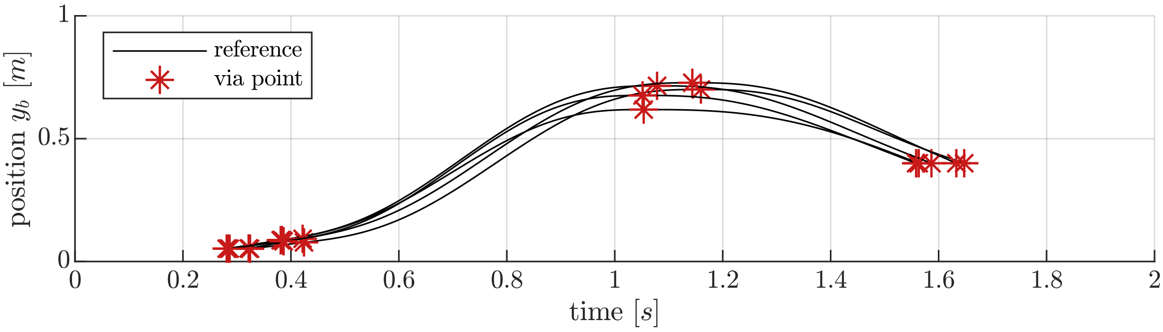


Figure 4-5: The reference trajectories $\bar{y}_b^{ref}(t,1)$ for five demonstrations resulting from four sequential quintic polynomials. Gaussian noise has been added to the third and the last via point. The reference trajectories are of the mode in which the box is lifted.

In this work, the objective for defining demonstration data is to create realistic demonstrations in which each execution of the task slightly differs from the others, with varying event times $t_i^{k|d}$ and varying sequences of these events. To do so, a Gaussian noise is added to the via points. Figure 4-5 depicts the reference trajectories for five demonstrations of the position of the box $\bar{y}_b^{ref}(t,1)$ resulting from four sequential quintic polynomials. These reference trajectories are defined for each task and tracked using the QP controller.

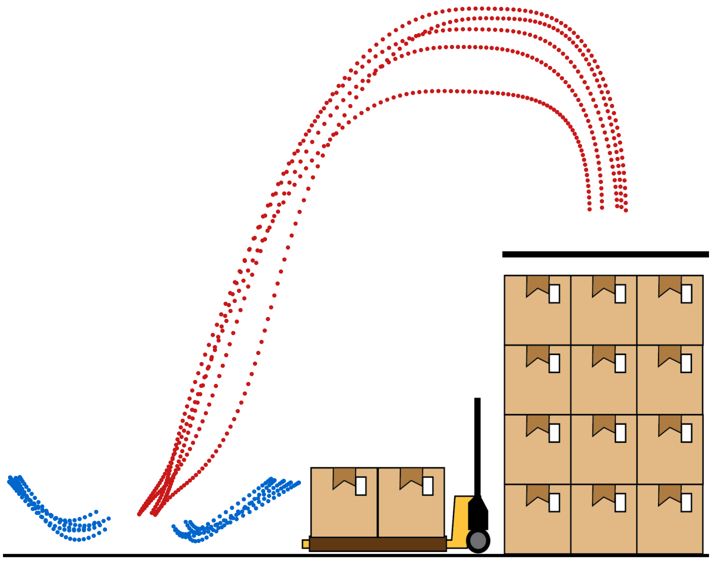


Figure 4-6: The demonstration data resulting from tracking defined reference trajectories using a QP controller. The position data of the end effectors in free motion is depicted in blue and the position data of the box when it is lifted is depicted in red.

The defined reference trajectories of all tasks and the via points are illustrated in Appendix C, which also covers generating the demonstration data in more detail. Figure 4-6 depicts demonstration data of the position of the end effectors in free motion and the position of the box when it is lifted. The demonstration data of all tasks are plotted over time in Figure D-1 and Figure D-2 of Appendix D. A video of the simulations which contains the demonstrations is available online[1].

The data from the demonstrations is extended and used to define extended trajectory distributions through the use of ProMPs. The process of going from demonstration data of the dynamic pick and place task to the trajectory distributions, is covered in the next section.

4-5 Demonstration data to mode dependent trajectory distributions

This section describes how the extended reference trajectories, used by the QP controller in combination with the reference spreading error, are obtained from the demonstration data. To do so, the data of the non-impact modes are extended to then fit a ProMP. The data is extended using the data extension methods described in Section 3-1-2. After extending the data, mode dependent movement primitives are fit as described in Section 3-2-3. The reader is referred to Appendix D, for plots of the extended demonstration data, as well as the resulting ProMP fit, for each reference trajectory and input of the non-impact modes. This section covers a few examples.

[1] https://youtu.be/FJ2i09gGAiU

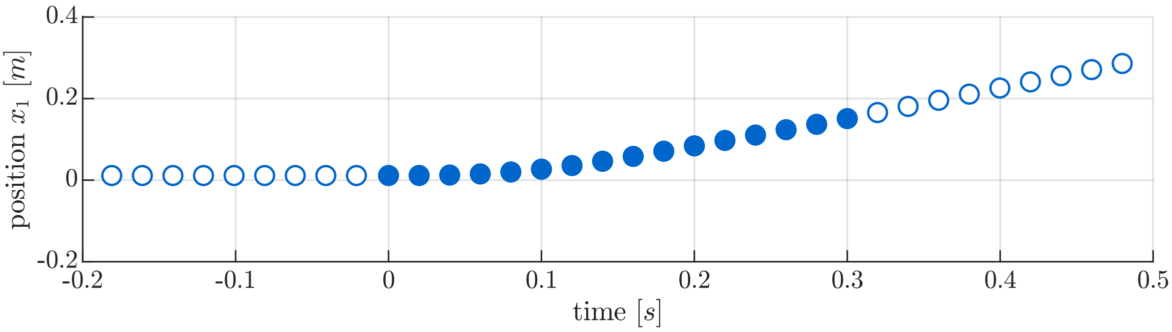


Figure 4-7: The data and the extended data of the position x_1, of one demonstration in the first mode. The data is extended forward until $t_0^{fw} = 0.5$ and backward until $t_0^{bw} = -0.2$. The original data is presented by blue disks while the extended data is presented by blue circles. For the sake of visibility, only one of every ten data points is plotted.

The extension of the position data of one of the end effectors in the first mode, using the first order hold, as is denoted in Section 3-1-2, is depicted in Figure 4-7, for one demonstration. The figure illustrates that the data is extended backward beyond $t = 0$, until $t_0^{bw} = -0.2$. This is done to assure that the boundary issue of Radial Basis Function (RBF) fitting, detailed in Section 3-2-1, occurs outside the time domain in which the system is expected to be in the corresponding mode. The data is extended forward until $t_0^{fw} = 0.5$.

The data of the third mode is extended backward until $t_1^{bw} = 0$ and forward up until $t_1^{fw} = 0.5$. The extension of the input data $u_{1,y}$ of one demonstration, using the zero order hold data extension method described in Section 3-1-2, is illustrated in Figure 4-8.

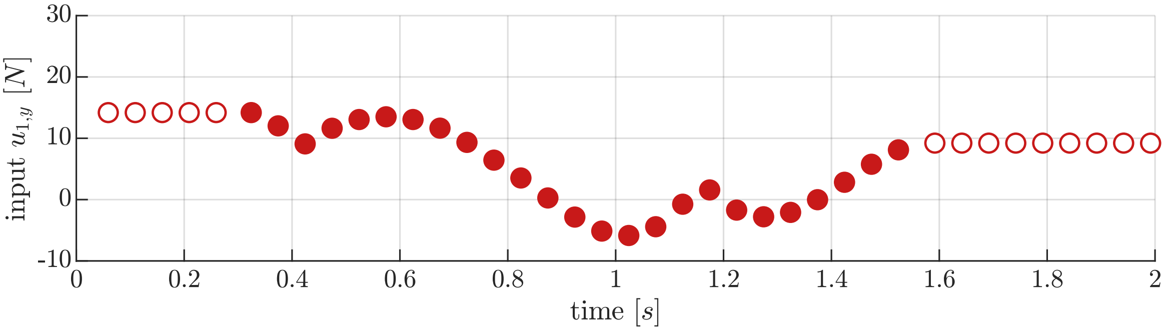


Figure 4-8: The demonstration data and the extended demonstration data of the input $u_{1,y}$, of one demonstration in the first mode. The data is extended forward until $t_0^{fw} = 2$ and backward until $t_0^{bw} = 0$. The original data is presented by red disks while the extended data is presented by red circles. For the sake of visibility, only one of every forty data points is plotted.

After extending the data of each demonstration of the non-impact modes, ProMPs are fit. This is done using the method described in Section 3-2-3, resulting in extended trajectory distributions for each mode. Appendix D shows the trajectory distributions of each task of the non-impact modes. We briefly cover the trajectory distribution of the angle of the box θ_b as an example.

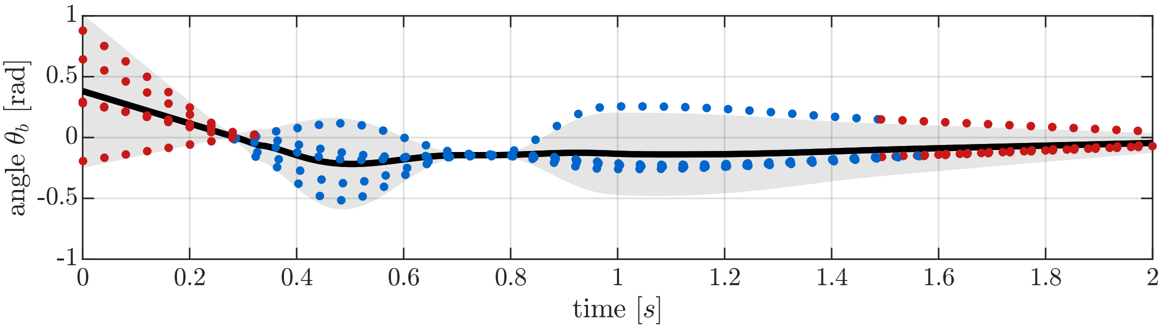


Figure 4-9: The demonstration data (blue) and the extended demonstration data (red) of the angle θ_b of five demonstrations. It also shows the resulting trajectory distribution with its mean depicted by the black line and the variance in grey. The used hyper parameters are: $Z = 200$, $h = 1e - 4$ and $\lambda_{reg} = 0$. One in every forty data points is plotted, for the sake of visibility.

The extended data and trajectory distribution of the angle of the box θ_b, resulting from fitting a ProMP on the extended data in the third mode, is illustrated in Figure 4-9. The chosen parameters, for this mode, to fit the trajectory distribution are: the number of RBFs $Z = 200$, the width of the RBFs $h = 1e - 4$, and the regression parameter $\lambda_{reg} = 0$. The mean of this trajectory distribution $\bar{\theta}_b^{ref}(t, 1)$ can be used as a reference by the QP controller.

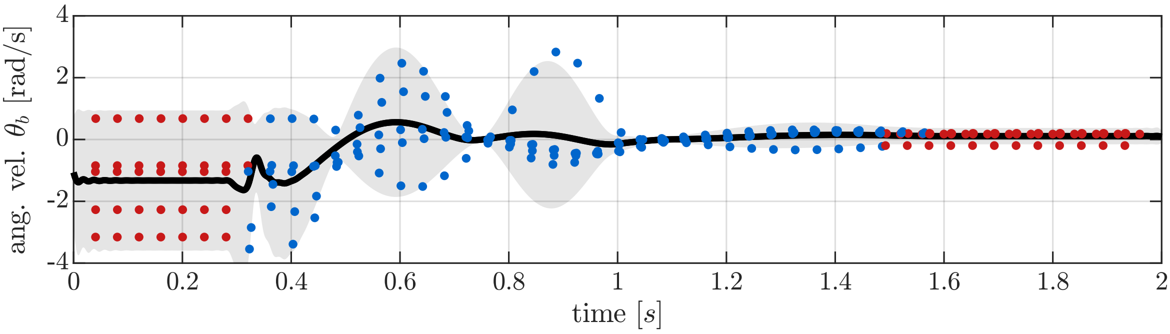


Figure 4-10: The numerical derivatives of the demonstration data (blue) and the extended demonstration data (red) of the angle θ_b of five demonstrations. It also shows the derivative of the trajectory distribution with its mean depicted by the black line and the variance in grey. The used hyper parameters are: $Z = 200$, $h = 1e - 4$ and $\lambda_{reg} = 0$. One in every forty data points is plotted, for the sake of visibility.

The derivative of the probability distribution of the angular velocity, as well as the numerical derivative of the angle data, is illustrated in Figure 4-10. The derivative of the mean of the velocities trajectory distribution $\dot{\bar{\theta}}^{ref}(t, 1)$, is used as a reference by the QP controller.

The mode dependent trajectory distributions are used to define a extended reference trajectory and feedforward, as is explained in Section 3-3. The mean of the extended trajectory distribution is used, in conjunction with the QP controller combined with the reference spreading error, as described in Section 4-3, for reference tracking. The performance of tracking the reference trajectories learned from the extended data is detailed in the next section.

4-6 Tracking results

This section covers the tracking performance resulting from tracking the reference trajectories obtained by the IA-LfD method. The tracking performance is covered separately for each mode.

The position data is extended using a first order hold and thus a jump may occur in the acceleration trajectory. This, in combination with the boundary issue (Section 3-2-1) being amplified when the second derivative is taken, is the reason that reference task acceleration $\ddot{g}_m^{ref}$ is set to zero, for each position and orientation task. The desired task acceleration, used in the QP controller of the non-impact modes, can be written as

$$\ddot{g}_m^{des}(t,i) = k_m^p e_{rs,m}(t,i) + k_m^d \dot{e}_{rs,m}(t,i). \tag{4-30}$$

End effectors in free motion. The mean of the trajectory distribution, of this mode, resulting from fitting ProMP on the extended demonstration data, is given by

$$\bar{\boldsymbol{\mu}}_\xi(t,0) = \begin{bmatrix} \bar{\mathbf{q}}_{ee}^{ref}(t,0) \\ \bar{\mathbf{u}}^{ff}(t,0) \end{bmatrix}, \tag{4-31}$$

The position component $\bar{\mathbf{q}}_{ee}^{ref}(t,0)$ and its derivative $\dot{\bar{\mathbf{q}}}_{ee}^{ref}(t,0)$, are used as references in the desired task accelerations $\ddot{g}_m^{des}(t,0)$ of the QP controller. These references, together with the tracking results are depicted in Figure 4-11. The used controller parameters are detailed thereafter.

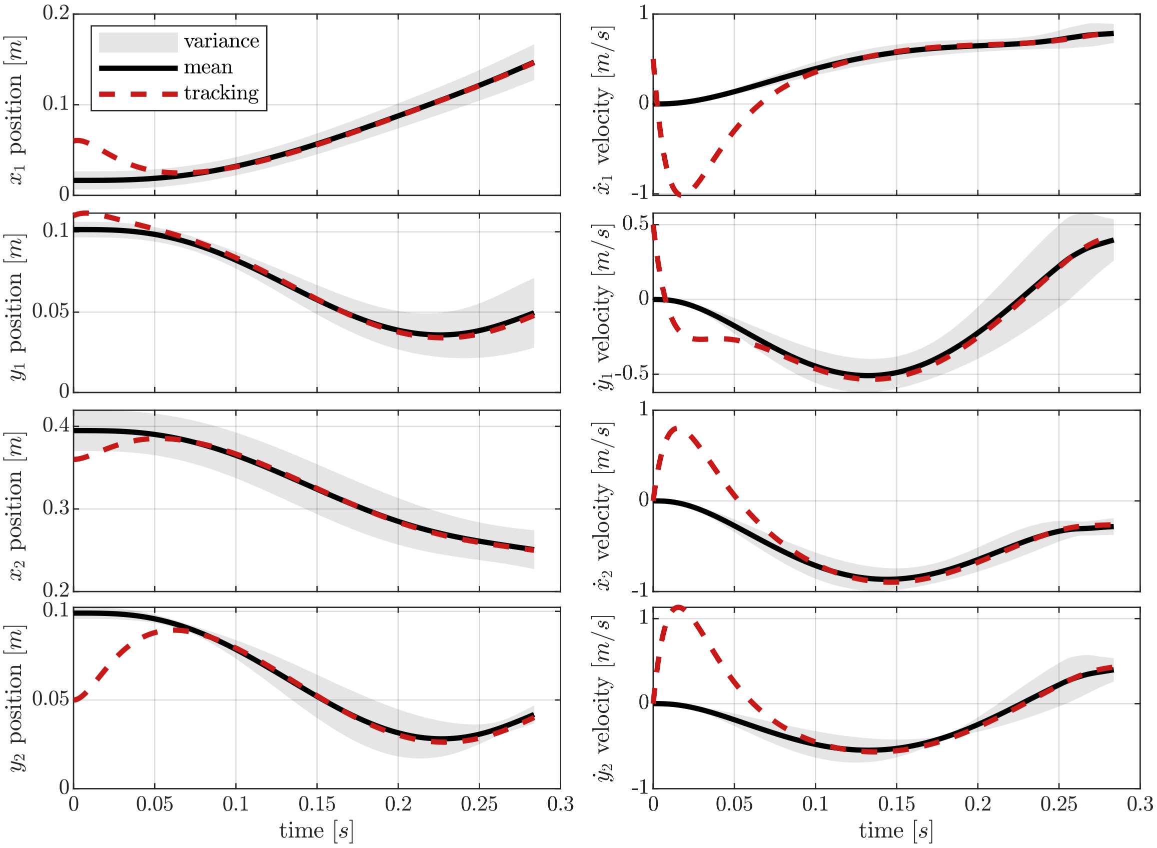


Figure 4-11: The performance of the QP controller tracking the mean of the trajectory distribution resulting from fitting a ProMP on the extended demonstration data of position of the end effectors. The tracking performance of velocity of the end effectors is also illustrated. The tracking performance is shown of the mode in which the end effectors are in free motion. The system is initiated with an offset.

The parameters of the QP controller in the first mode are the weights w_m, the derivative gain k_m^d, and the proportional gain k_m^p, for each task $m \in \{1, 2, 3, 4\}$. The weights of the tasks w_m are set equal to each other, as no position task, within this mode, is prioritized over another. The proportional gain $k_m^p = 900$ and the derivative gain $k_m^d = 60$ for each task m. These controller parameters are not optimized for performance. The system has been initiated with an offset to illustrate that the position of the end effectors converge to the mean of the trajectory distributions. Figure 4-11 illustrates that the QP controller is able to track both the learned reference trajectory $\bar{\mathbf{q}}_{ee}^{ref}(t, 0)$ and its derivative $\dot{\bar{\mathbf{q}}}_{ee}^{ref}(t, 0)$, with the parameters mentioned above.

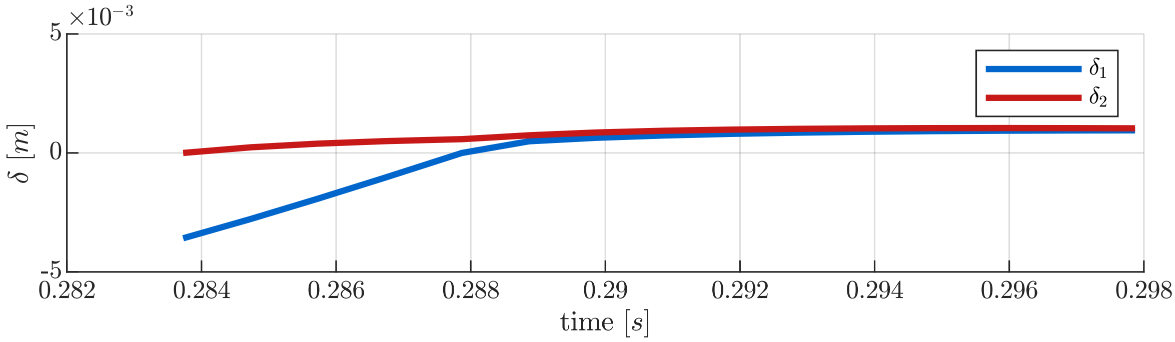


Figure 4-12: A plot of the indentation values over time in the first impact phase. A positive indentation value indicates that the corresponding end effector is in contact with the box. Both end effectors establish contact with the box.

The first impact phase. The input during the first impact phase is the feedforward component $\bar{\mathbf{u}}^{ff}(t,0)$, of the learned trajectory distribution, of the mode in which both end effectors are in free motion. The first impact phase is initiated the first time an end effector establishes contact with the box. The goal of the feedforward term in this mode is to assure that this end effector stays in contact with the box while the second end effector moves towards the box, establishes contact, and stays in contact. The plots of the indentation values δ_1 and δ_2, between the end effector and the box, are depicted in Figure 4-12 to show that the desired behaviour is achieved. The figure illustrates that the learned feedforward $\bar{\mathbf{u}}^{ff}(t,0)$ in combination with the systems dynamics results in the stabilization of contact, as is mentioned in Section 2-1. The initial state $\mathbf{q}(t_0^1,0,1)$ of this mode is the result of applying a disturbance near the end of the previous mode.

Lifting the box. The mean of the extended trajectory distributions of this mode, resulting from fitting ProMP on the extended demonstration data, is formulated as

$$\bar{\boldsymbol{\mu}}_{\xi}(t,1) = \begin{bmatrix} \bar{\mathbf{q}}_b^{ref}(t,1) \\ \bar{\mathbf{u}}^{ff}(t,1) \\ \bar{\lambda}_{N,1}^{ref}(t,1) \\ \bar{\lambda}_{N,2}^{ref}(t,1) \end{bmatrix}. \tag{4-32}$$

The position component $\bar{\mathbf{q}}_b^{ref}(t,1)$, containing references for the position and orientation of the box, together with its derivative $\dot{\bar{\mathbf{q}}}_b^{ref}(t,1)$, and the references on the normal forces $\bar{\lambda}_{N,1}^{ref}(t,1)$, $\bar{\lambda}_{N,2}^{ref}(t,1)$, are tracked using the QP controller. These references, together with the tracking performance are depicted in Figure 4-11. The used controller parameters are detailed thereafter.

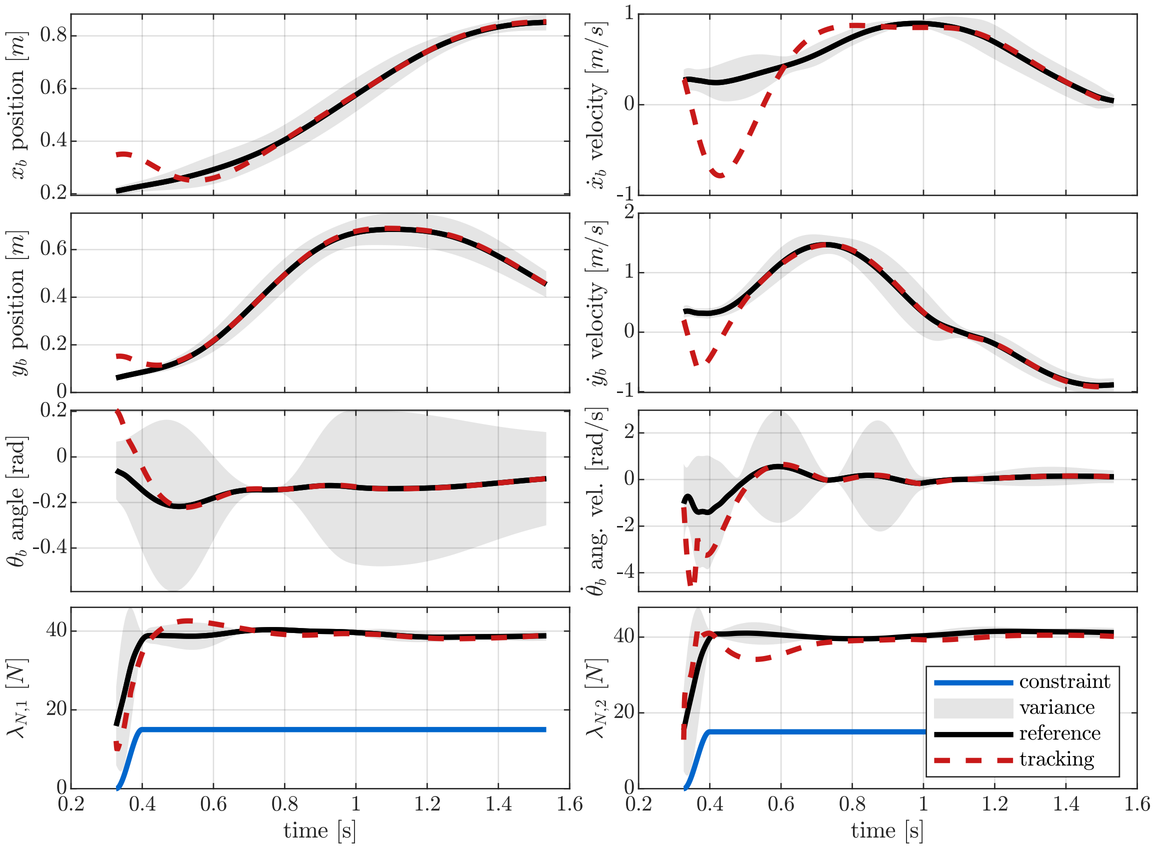


Figure 4-13: The performance of the QP controller tracking the mean of the trajectory distribution resulting from fitting a ProMP on the extended demonstration data of the position of the box, orientation of the box, and contact forces. The tracking performance of velocity and angular velocity is also illustrated. The tracking performance is shown of the mode in which both end effectors are in contact and the box is lifted. The system is initiated in this mode with an offset.

The parameters that are to be defined, for the QP controller in the third mode, are the weights of the position tasks w_m^{pos}, the weights on the force tasks w_v^{force}, the proportional gain k_m^p and the derivative gain k_m^d. No position or orientation task of the box is prioritized over the others, thus these weights are all set equal to one. In this work, the tracking performance of the position and orientation of the box is defined to be more important than the tracking performance of the force tasks and thus $w_z^{force} = w_2^{force} = 1e-3$. To maintain contact with the box, the constraint on the contact force $\lambda_N^{min} = 15$. The controller gains are equal to the first mode: $k_m^p = 900$ and $k_m^d = 60$ for each task m. Again, these controller parameters are not optimized for performance. To show the tracking performance, of the QP controller in this mode, the system is initiated in this mode with an offset. It is shown in Figure 4-13 that the QP controller is able to track both the learned reference trajectory $\bar{\mathbf{q}}_b^{ref}(t,1)$ and its derivative $\dot{\bar{\mathbf{q}}}_b^{ref}(t,1)$. It also shows that the contact forces $\lambda_{N,1}$ and $\lambda_{N,2}$ stay above the defined constraint.

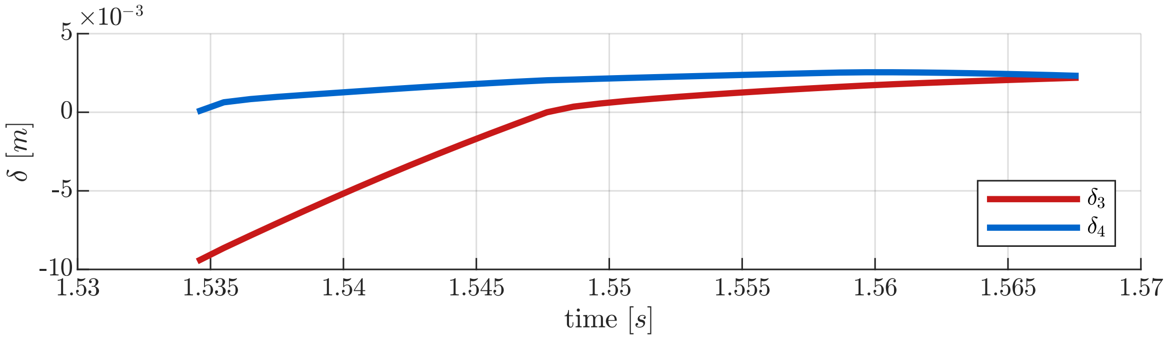


Figure 4-14: The indentation values δ_3 and δ_4 of the corners of the box and the shelf, plotted over time. A positive indentation value indicates that the corresponding corner is in contact with the shelf.

The second impact phase. The second impact phase is initiated when one of the corners establishes contact with the shelf. The input in this mode is the feedforward component $\bar{\mathbf{u}}^{ff}(t, 1)$, from the learned trajectory distribution of the previous mode, in which the box is lifted from the ground. This feedforward is able to put both corners on the shelf, as is shown by the plots of the indentation values δ_3 and δ_4 in Figure 4-14.

The initial state of this mode $\mathbf{q}(t_1^1, 1, 1)$ is equal to the final state of the previous mode using the controller denoted in the previous subsection. The plot illustrates that the first corner stays in contact with the shelf. It also depicts that the second corner establishes and stays in contact with the shelf. Again, showing that the learned feedforward $\bar{\mathbf{u}}^{ff}(t, 1)$ in combination with the systems dynamics results in the stabilization of contact.

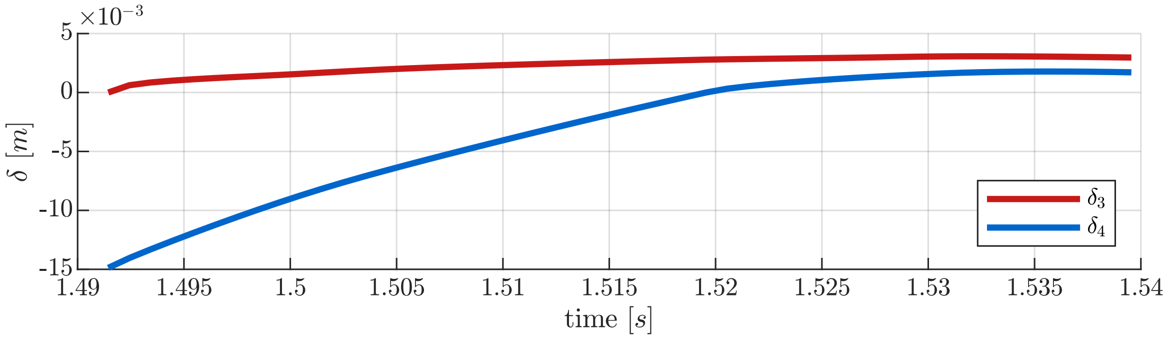


Figure 4-15: The indentation values δ_3 and δ_4 of the corners of the box and the shelf, plotted over time. A positive indentation value indicates that the corresponding corner is in contact with the shelf. The system is initialized such that the sequence in which the corners establish contact differs from the expected sequence.

The feedforward, in combination with the systems dynamics, can result in the stabilization of contact when the system is disturbed before initial contact is established. This illustrated in Figure 4-15 in which the sequence of the corners establishing contact differs from the expected sequence.

Tracking the extended reference trajectories, obtained by fitting ProMP on the extended demonstration data of each mode with a QP controller, results in the successful execution of the task to lift a box with two end effectors to subsequently put it on top of a shelf. The snapshots of the execution of the dynamic pick and place task and the trajectory distributions are depicted in Figure 4-16. A video of the tracking, together with the demonstrations, is available online[2].

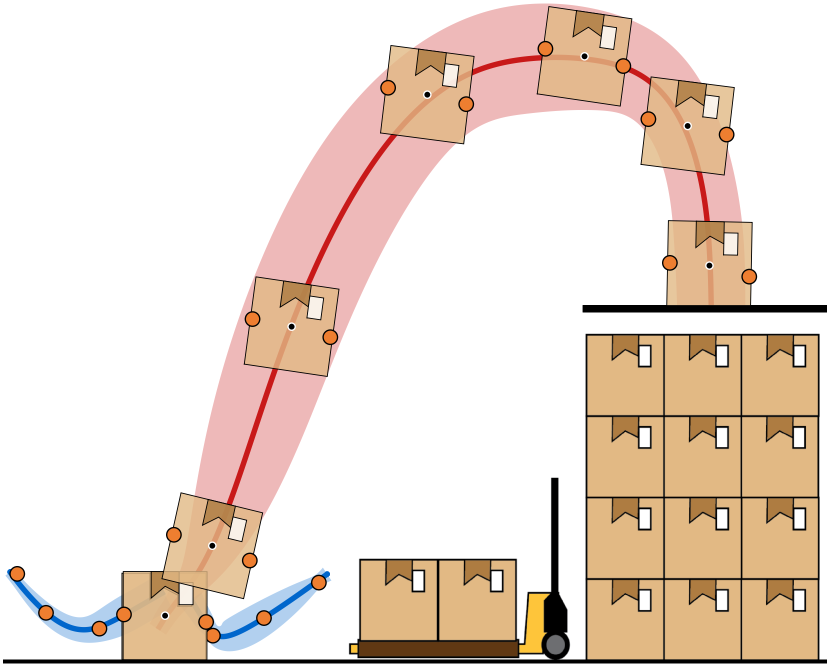


Figure 4-16: Figure illustrating the effectiveness of the IA-LfD method for two end effectors tasked to pick up a box and put it on a shelf. The trajectory distribution of the mode in which the end effectorss are in free motion ($i = 0$) is depicted in blue. The trajectory distribution of the mode in which the box is lifted ($i = 1$) is depicted in red.

4-7 Summary

This chapter covered the numerical validation of the IA-LfD method. The proposed method is validated by simulating a dynamic pick and place task. The simulation consists of two end effectors tasked to simultaneously grab a box to then put it on top of a shelf. The model of the dynamical system of two end effectors and a box has been detailed. To apply the proposed method, demonstration data is generated. This chapter explained that the demonstration data is generated by tracking manually defined reference trajectories with the QP controller. The demonstration data is extended and trajectory distributions are obtained through the use of ProMP. The mean of the extended trajectory distributions is tracked, which results in the successful completion of the dynamic pick and place task. The next chapter covers the conclusions drawn from this research and the recommendations for future work.

[2] https://youtu.be/FJ2i09gGAiU

Chapter 5

Conclusion and Recommendations

In this chapter, the conclusions that can be drawn from this work are discussed. The first section contains a reflection on the defined goal of the research. Thereafter, opportunities for future research are stated as recommendations.

5-1 Conclusion

The goal of this thesis has been stated as:

To develop a learning strategy that is able to learn and execute tasks in which contact is made at non-zero speed.

This thesis shows that it is possible to extend state of the art learning from demonstration methods, such as probabilistic movement primitives, to learn and execute tasks in which contact is made at non-zero speed by incorporating in the learning pipeline key elements of the control strategy reference spreading. A learning strategy has been presented for both single and simultaneous impacts. The thesis proposes to extend demonstration data split into different modes, segmented by impact events, to then fit probabilistic movement primitives. This results in an extended reference trajectory which can be used in the reference spreading error. To validate the new learning approach, the case is considered in which two end effectors are tasked to grab a box simultaneously, making contact at non-zero speed, to then place the box on a shelf. The task is successfully learned and executed through the use of the proposed method in conjunction with a quadratic programming controller, showing the effectiveness of the impact-aware learning strategy.

5-2 Recommendation

The conducted research has shown the effectiveness of the impact-aware learning strategy. However, based on the findings of this research, there are several topics that can be thought of for future research. The topics are impact detection, learning controllers, and physical experiments. These topics are listed below.

5-2-1 Impact detection

To apply the control strategies described in this thesis, it is necessary to have a tool to detect the current mode. The impact detection tool is necessary to define in which mode the system is, such that the controller can switch accordingly. The impact phase for simultaneous impacts is denoted as the time between the start of the first impact and the end of the last impact. The tool needs to be able to make this discretion between the start of the impact phase and the end of the impact phase. In [11] a tool is proposed which assumes a maximum bound on acceleration (or jerk) that is used to predict the allowed future joint displacement. If the displacement is beyond this range, an impact is deemed to have occurred. In the context of learning from demonstration, the demonstration data may be used to define said bound. A data-driven approach to detect in which mode the system is in can be done similarly to [46] in which the state of the system is classified through Support Vector Machines or Gaussian Mixture Models to determine which primitive is to be executed.

5-2-2 Learning controllers

This thesis covers how reference trajectories can be obtained through the use of movement primitives. In the field of learning from demonstration, research is done on learning controllers as well. In other words, how can the demonstration data be used to define a control law. Examples of research in which controller parameters are learned are [41, 42, 17]. Learning the controller gains would be a useful addition to the proposed learning approach of this work as the number of controller parameters can rapidly increase for multiple modes.

5-2-3 Physical experiments

Once an effective impact detection tool has been defined, physical experiments can be performed. If possible, this physical experiment can be similar to the considered case in this thesis: grabbing a box with two robotic arms to then place it on top of a shelf. However, the impact-aware learning from demonstration method can also be applied to other physical experiments in which impact occurs.

Appendix A

Comparison between ProMP and GMR

The the accompanying literature review resulted in two movement primitives able to learn a time driven trajectory. These movement primitives are Probabilistic Movement Primitive (ProMP) and Gaussian Mixture Regression (GMR). This appendix covers the differences between the two and why ProMP is better suited for learning time driven trajectories than GMR.

To understand the difference between the two movement primitives, each movement primitive has to be explained. The reader is referred to Section 2-2 for an explanation of ProMP. GMR is quickly explained in this appendix. For a detailed explanation on GMR, it is recommended to read [47].

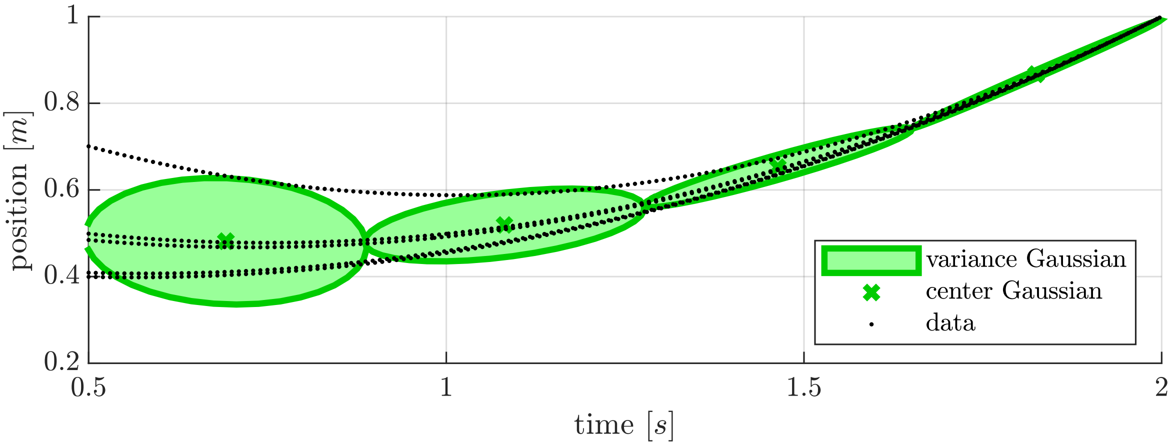


Figure A-1: Figure illustrating the Gaussian mixture model fit on position data. The number of Gaussians is four. Code from [48] was used to obtain this figure.

In the method explained in [47], Gaussian mixture models are fit on the data. Gaussian mixture models are used to model a distribution from a limited amount of observations. Fitting of the Gaussian distributions can be done through the use of expectation maximization

[49]. Figure A-1 illustrates the resulting Gaussian distributions, fit on the position data of multiple demonstrations. The figure illustrates that the temporal component (time) of the data is treated equally to the spatial component (position) of the data.

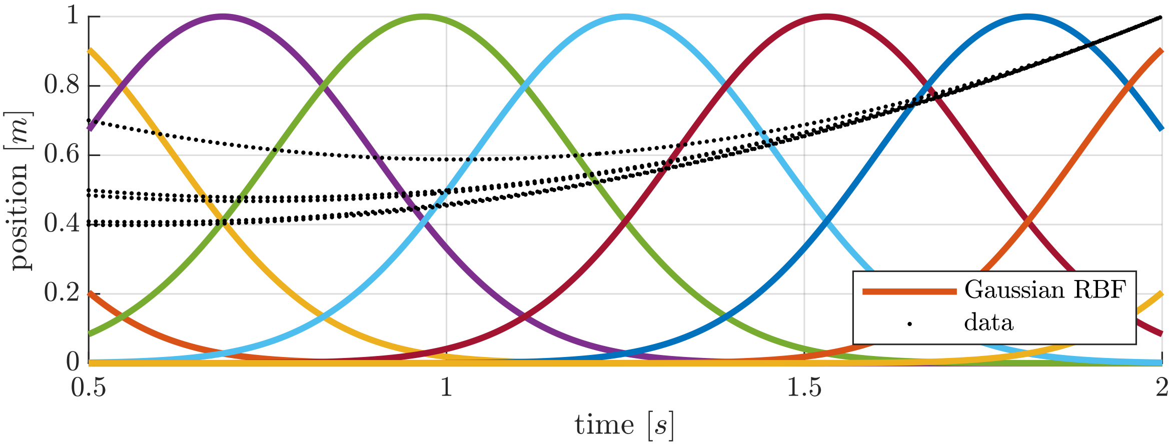


Figure A-2: Figure illustrating multiple radial basis functions and the demonstration data upon which they are to be fit.

The Gaussian radial basis functions in ProMP treat the time data as a different entity than the position data are illustrated in Figure A-2.

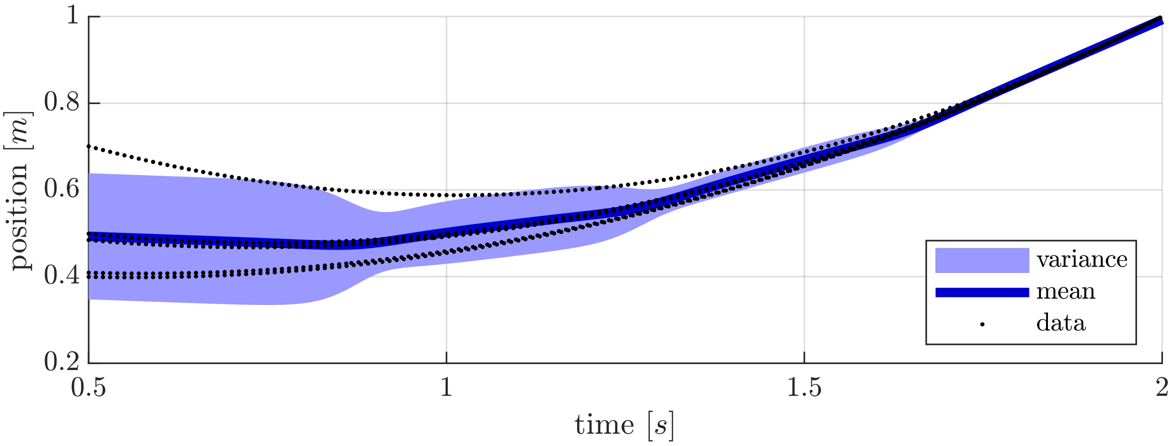


Figure A-3: Figure illustrating the trajectory distribution resulting from GMR. It also shows the demonstration data which it aims to represent. Code from [48] was used to obtain this figure.

The Gaussian mixture models illustrated in Figure A-1 are used to generate a trajectory distribution. This is done by deriving the conditional probability $p(x|t)$, in which x is the spatial component and t the temporal component. In our example of position data, this conditional probability distribution determines the probability of the position x given time t. Doing so for each time step, results in the trajectory distribution illustrated in Figure A-3.

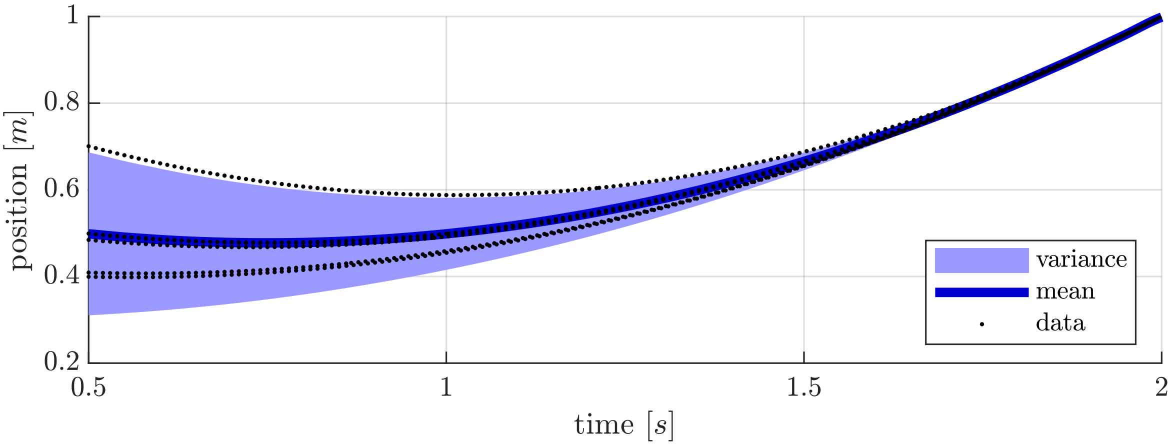


Figure A-4: Figure illustrating the trajectory distribution resulting from ProMP. It also shows the demonstration data which it aims to represent.

For ProMP, the radial basis functions shown in Figure A-2 are multiplied by weights and summed to be fit on each trajectory. The trajectory distribution resulting from fitting Gaussians on these weights is depicted in Figure A-4. It shows that it does a better job representing the demonstration data than the trajectory distribution resulting from GMR.

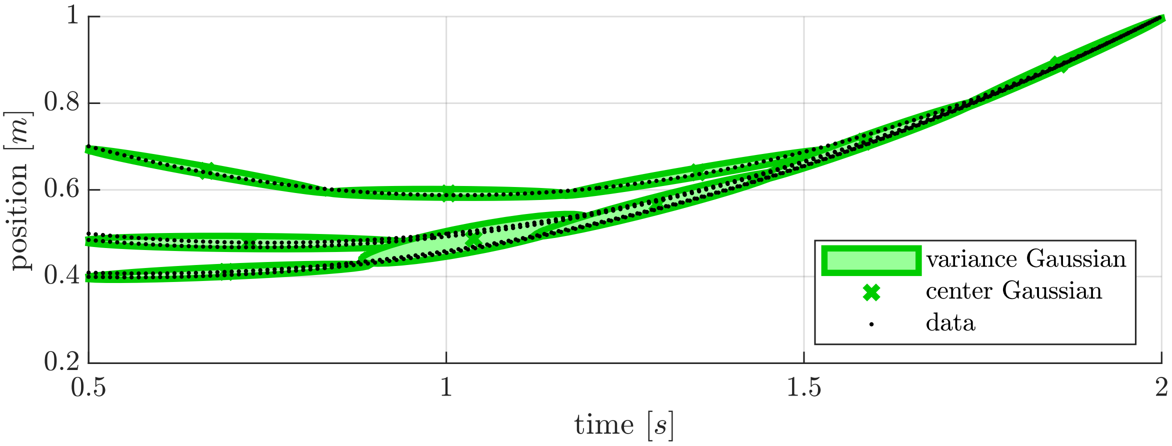


Figure A-5: Figure illustrating the Gaussian mixture model fit on position data. The number of Gaussians is nine. Code from [48] was used to obtain this figure.

The comparison above between GMR and ProMP can be seen as an unfair comparison due to the number of Gaussians used for GMR (four) being much lower than the number of Gaussians used for ProMP (nine). However, increasing the number of Gaussians for GMR to nine, results in them overlapping in time. This is illustrated in Figure A-5. The resulting trajectory distribution is depicted in Figure A-6.

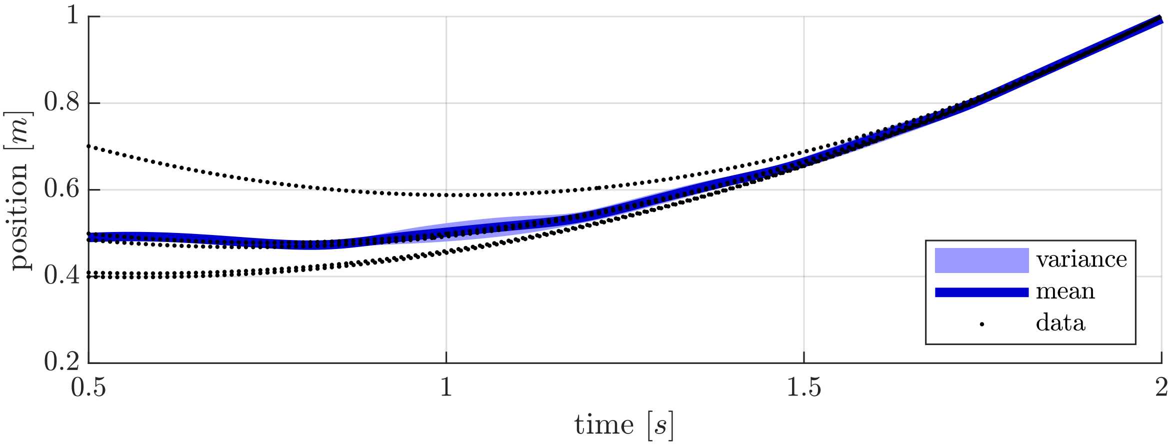


Figure A-6: Figure illustrating the trajectory distribution resulting from GMR. It also shows the demonstration data which it aims to represent. Code from [48] was used to obtain this figure.

The variance of the trajectory distribution does a poor job representing the variance of the data. The variance of the trajectory distribution is an average of the variances of the Gaussian mixture model. Increasing the number of Gaussians results in multiple Gaussian mixture models representing the data of the same time domain. As the data is represented by these Gaussian mixture models that overlap in time, the variance of each Gaussian becomes less, resulting in a lower variance of the trajectory distribution.

Whilst GMR is able to learn time driven trajectories, it is also facilitates learning the relation between high dimensional input data and high dimensional output data. However, it is shown that ProMP is better suited for learning time driven trajectories as it treats time as a separate entity.

Appendix B

Model parameters

This appendix contains an overview of the parameters used in the dynamic model of the end effectors and the box. Table B-1 includes their symbols, descriptions, and values used in the simulation of lifting a box and putting it on the shelf.

Table B-1: Overview of the model parameters and the used values during simulation.

Symbol	**Description**	**Value**
m_1	Mass of end effector 1	1
m_2	Mass of end effector 2	1
m_b	Mass of the box	1
h_b	Height of the box	0.1
w_b	Width of the box	0.1
I_b	Inertia of the box	$\frac{1}{12}(w_b^2 + h_b^2)$
k_{ccm}	Stiffness coefficient of compliant contact model	$1e5$
c_{ccm}	Damping coefficient of compliant contact model	20
μ_w	Friction coefficient	0.7
k_w	The slope of the friction model at zero velocity	100
x_{shelf}	The x value of the left point of the shelf	0.7
y_{shelf}	The height of the shelf	1

Appendix C

Generating demonstration data

As is covered in Section 4-4, demonstration data is generated by manually defining reference trajectories and tracking those with the Quadratic Programming (QP) controller detailed in Section 3-3-2. The reference trajectories are defined through the use of quintic polynomials in conjunction with via points. Generating the demonstration data is covered for each mode individually in this appendix.

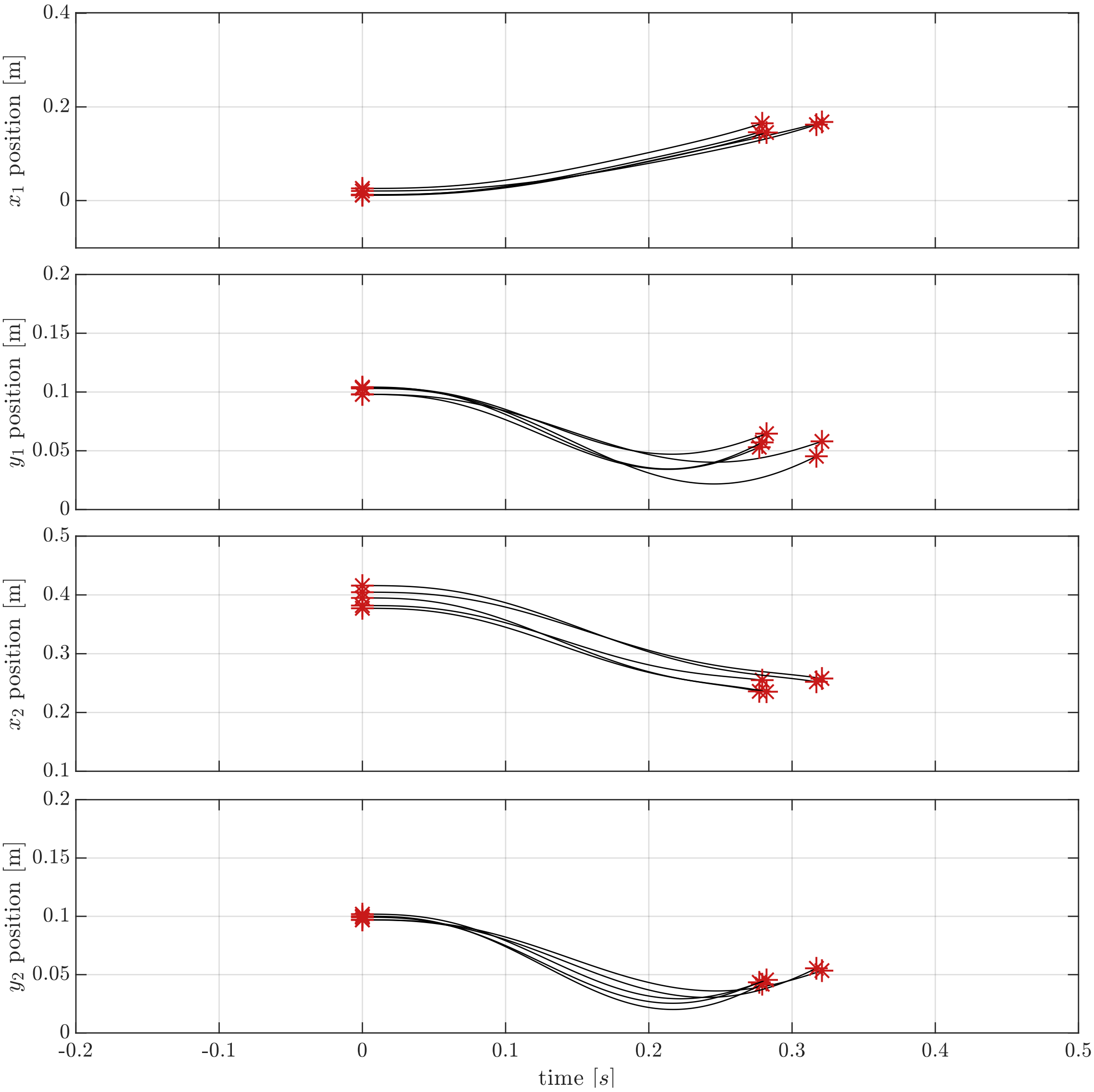


Figure C-1: The reference trajectories for the position of the end effectors resulting from four sequential quintic polynomials. It shows that a Gaussian noise has been applied to some of the via points.

End effectors in free motion. As described in Section 4-3, the tasks of the QP controller in the first mode are defined as a desired position of the end effectors. Thus, the reference trajectories $\bar{x}_1^{ref}(t,0), \bar{y}_1^{ref}(t,0), \bar{x}_2^{ref}(t,0)$, and $\bar{y}_2^{ref}(t,0)$ need to be defined.

The variance of the demonstrations in this mode results from, inter alia, the Gaussian noise added to the the initial position of the box $x_b(0)$ and the initial position of the end effectors $x_1(0), y_1(0), x_2(0), y_2(0)$. The reference trajectories in this mode consist of one quintic polynomial, and thus only two via points need to be defined for each reference trajectory. The desired final position of the end effectors is placed slightly inside the box. The manually defined reference trajectories and the via points are illustrated in Figure C-1.

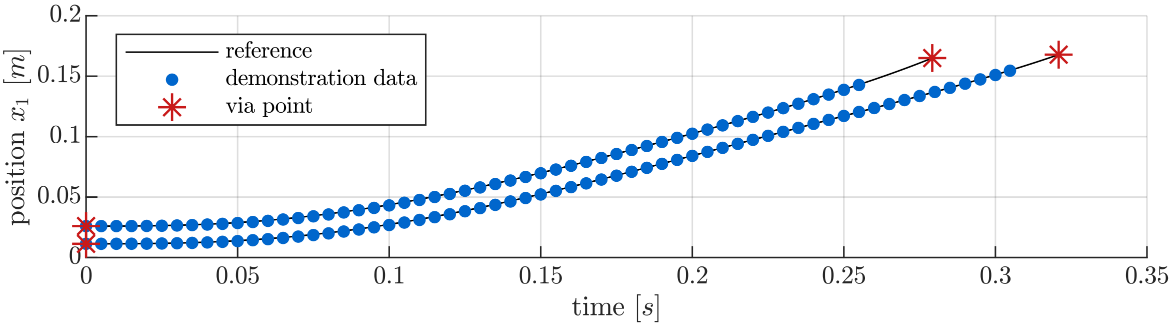


Figure C-2: The demonstration data of two of the five demonstrations resulting from references defined through the use of quintic polynomials. The demonstration data is plotted until the event times $t_0^{1|1}$ and $t_0^{1|2}$ which differ between demonstrations. For the sake of visibility only one of every ten data points is plotted.

The noise added to the initial position of the end effectors and the box together with the noise added to the desired final velocity of the end effectors results in varying event times $t_0^{1|d}$ between demonstrations when tracking these manually defined reference trajectories. Figure C-2 depicts demonstration data resulting form tracking the reference trajectories for two of the five demonstrations. The demonstration data is plotted until one of the end effectors establishes contact with the box, illustrating the difference in event times $t_0^{1|d}$ between the two demonstrations.

The resulting demonstration data, that is used to learn a trajectory distribution, is the position of the end effectors $\mathbf{q}_{ee} \in \mathbb{R}^4$ and the input data $\mathbf{u} \in \mathbb{R}^4$. Thus, one data point is written as

$$\boldsymbol{\xi}_a^{0|d} = \begin{bmatrix} \mathbf{q}_{f,a}^{0|d} \\ \mathbf{u}_a^{0|d} \end{bmatrix}. \tag{C-1}$$

This demonstration data is depicted in Figure D-1 and Figure D-2 of Appendix D.

First impact phase. The QP controller uses the extended feedforward of the previous mode during the impact modes. The input during the first impact phase is a constant value equal to the last input of the previous mode. This can be written as

$$\mathbf{u}(t, 0, 1) = \mathbf{u}(t_0^1, 0). \tag{C-2}$$

In this work, no data of the impact phase is used to learn extended trajectory distributions.

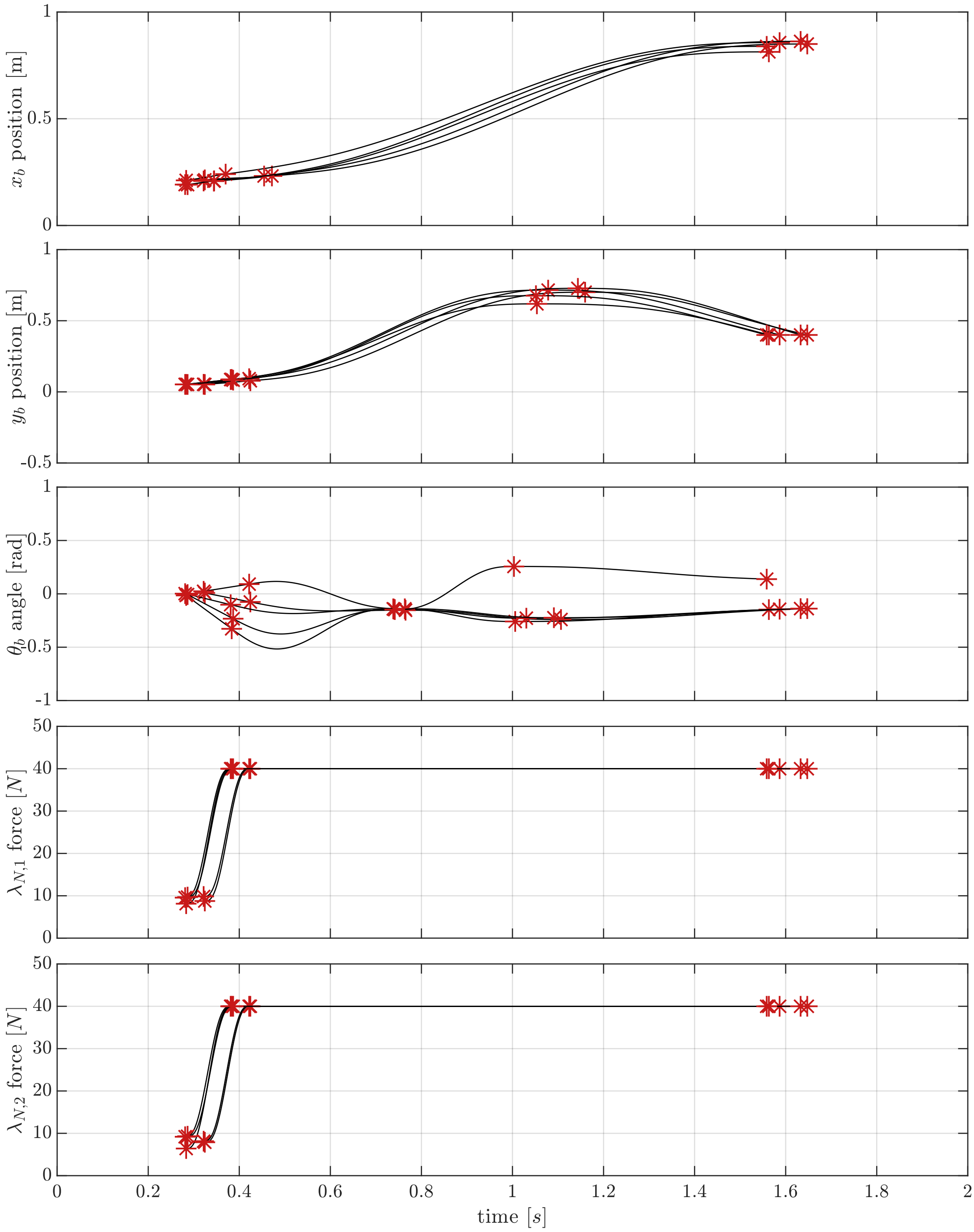


Figure C-3: The reference trajectories for the position and orientation of the box. It also shows the reference trajectories for the normal forces applied by the end effectors on the box. A Gaussian noise has been applied to some of the via points. The reference trajectories are indicated by a black line while the via points are visualised by red stars.

Lifting the box. The tasks of the QP controller in the mode in which the box is lifted are the desired position and orientation of the box together with desired contact forces. Thus the reference trajectories $\bar{x}_b^{ref}(t,1)$, $\bar{y}_b^{ref}(t,1)$, $\bar{\theta}_b^{ref}(t,1)$, $\bar{\lambda}_{N,1}^{ref}(t,1)$, and $\bar{\lambda}_{N,2}^{ref}(t,1)$ need to be defined.

Similarly to the trajectory of the end effectors, the final position of the box y_b is placed slightly below the shelf such that the reference is still defined if the time of impact is later than expected when tracking the reference. The variance within this mode results, inter alia, from the different initial state between demonstrations. The variation of the initial position, velocity, and acceleration is the result of the variation between demonstrations in the previous modes. Additional variance has been applied to the trajectories by adding a Gaussian noise to the via points. These via points, and the trajectories resulting from the quintic polynomial, are illustrated in Figure C-3. The reference trajectories defined for the angle of the box and the contact forces are explained in more detail below.

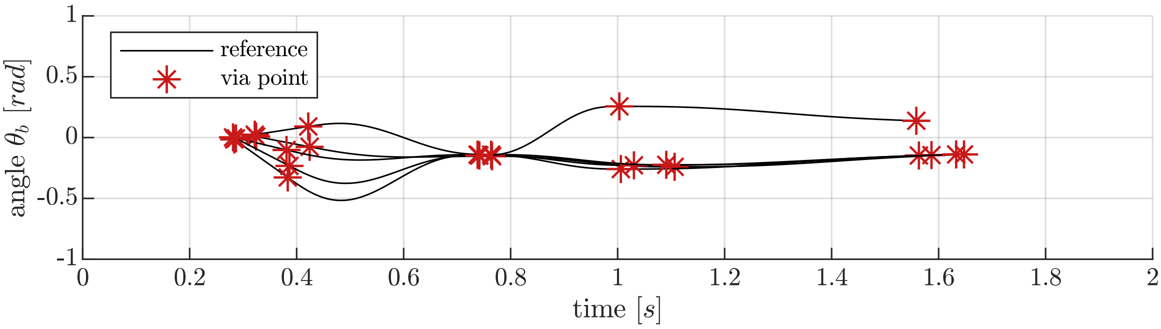


Figure C-4: The reference trajectories $\bar{\theta}_b^{ref}(t,1)$ for five demonstrations resulting from four sequential quintic polynomials. Gaussian noise has been added to the fourth and the fifth via point. The position of the fourth and fifth via point is negative for four of of the five demonstrations.

To vary the sequence in which the corners of the box make contact with the shelf, a noise has been added to some of via points used to define the reference $\bar{\theta}_b^{ref}(t,1)$ of the angle of the box. The last and second to last via point of the angle of the box θ_b have both been multiplied by the outcome of a Bernoulli distribution, essentially setting the sign of that via point to be negative with a chance of 80%. The resulting reference trajectories are illustrated Figure C-4.

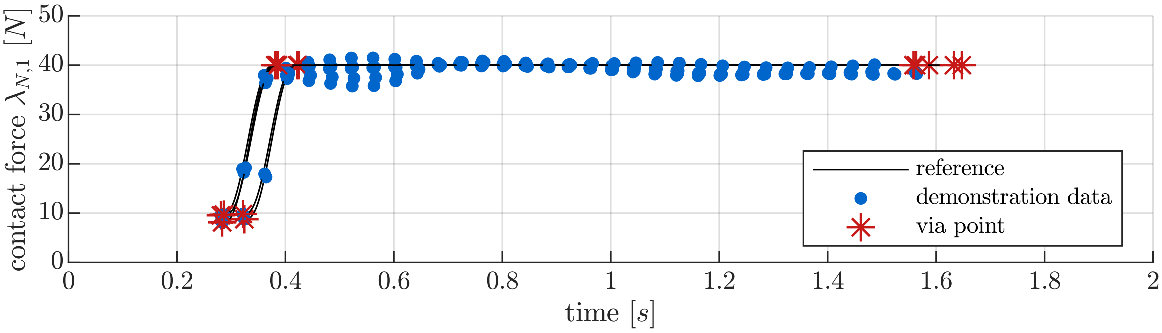


Figure C-5: The reference trajectories $\bar{\lambda}^{ref}_{N,1}(t,1)$ for five demonstrations and the data resulting from tracking these references. For the sake of visibility, only one of every forty data points is plotted.

Reference trajectories need to be defined for the normal forces applied by the end effectors on the box. For the sake of simplicity, the reference trajectory is a smooth ramp up to a constant value, equal between the demonstrations. The weights of the contact forces w^{force}_{z} are set to be relatively low compared to the weights of the position task w^{pos}_{m}. The reference trajectories defined by the quintic polynomials and their via points are depicted in Figure C-5. The figure also illustrates the resulting demonstration data.

The demonstration data of this mode, resulting from tracking the references with the defined QP controller, is the position and orientation of the box $\mathbf{q}_b \in \mathbb{R}^3$, the normal forces $\lambda_{N,1}$, $\lambda_{N,2}$, and the input data $\mathbf{u} \in \mathbb{R}^4$. Thus, one data point is formulated as

$$\boldsymbol{\xi}^{1|d}_{a} = \begin{bmatrix} \mathbf{q}^{1|d}_{b,a} \\ \mathbf{u}^{1|d}_{a} \\ \lambda^{1|d}_{N,1,a} \\ \lambda^{1|d}_{N,2,a} \end{bmatrix}. \tag{C-3}$$

This demonstration data is depicted in Figure D-3 and Figure D-4 of Appendix D.

The second impact phase. Similar to the first impact phase, a simple feedforward is used as a control law. The input during the second impact phase is a constant value equal to the last input of the previous mode. This can be written as

$$\mathbf{u}(t,1,1) = \mathbf{u}(t^1_1,1). \tag{C-4}$$

Again, in this work no data of this mode is used to learn an extended trajectory distribution.

Appendix D

ProMP fits on extended demonstration data

This appendix contains plots of the trajectory distributions resulting from fitting a Probabilistic Movement Primitive (ProMP). The demonstration data comes from tracking the reference trajectories which were defined by quintic polynomials in the non-impact modes. The extension of the data is done through a first order hold and zero order hold for position and force data respectively. The regularization parameter λ_{reg} is set to zero. The number of Gaussian distributions fit on the demonstration data is 50 and 200 respectively. The probability distributions resulting from the input data of the mode in which the end effectors are moved towards the box are illustrated in Figure D-1, the probability distribution for the position in the same mode is depicted Figure D-2. The probability distribution of the input data of the mode in which the box is lifted from the ground is illustrated in Figure D-3. For the position data of the same mode it is visualized in Figure D-4.

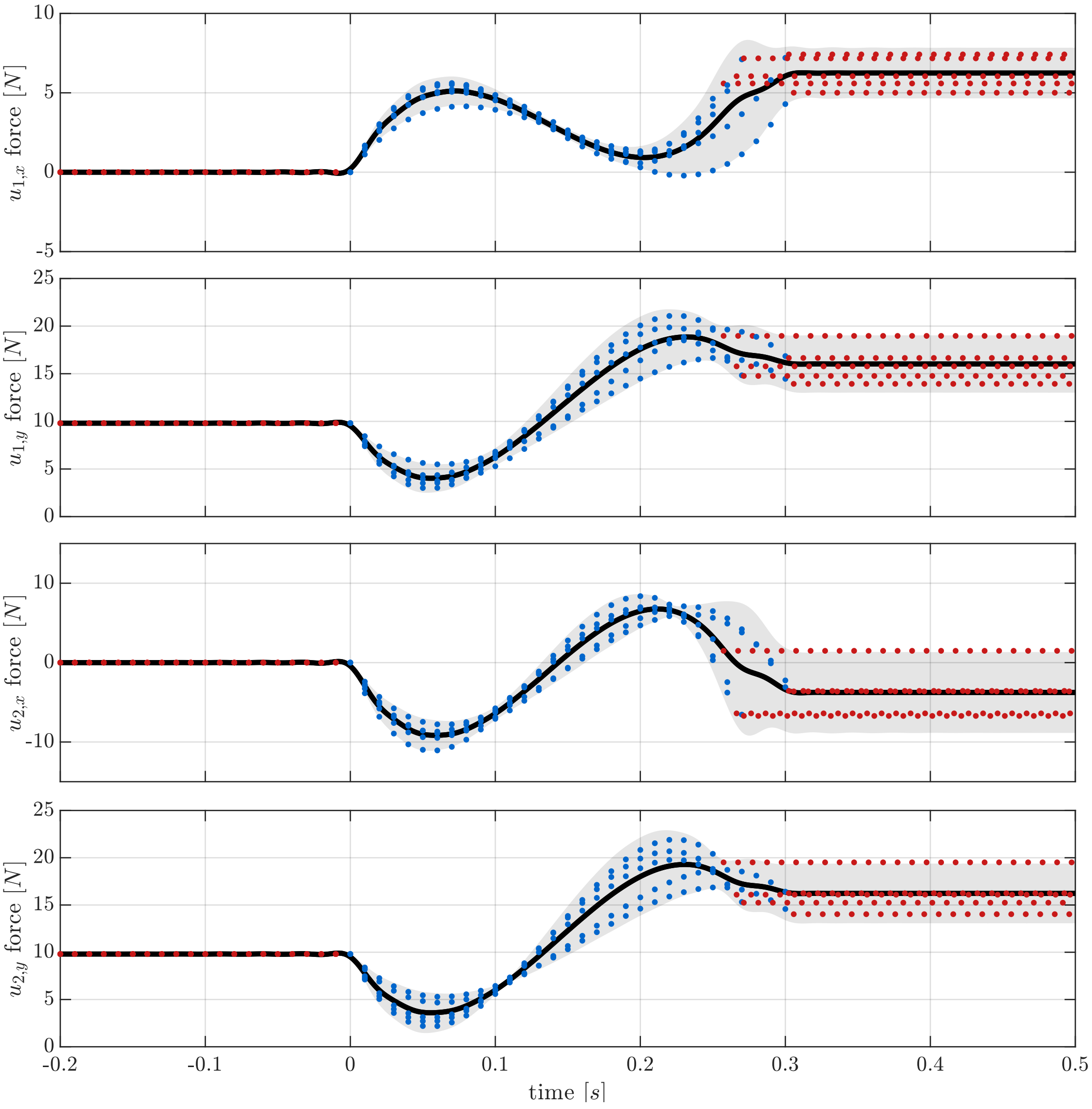


Figure D-1: The demonstration data (blue) and the extended demonstration data (red) of the input $\mathbf{u}$ of five demonstrations, in the first mode. It also shows the resulting trajectory distribution with its mean as the black line and the variance in grey. The used hyper parameters are: $Z = 50$, $h = 1e - 4$ and $\lambda_{reg} = 0$. One in every ten data points is plotted for the sake of visibility.

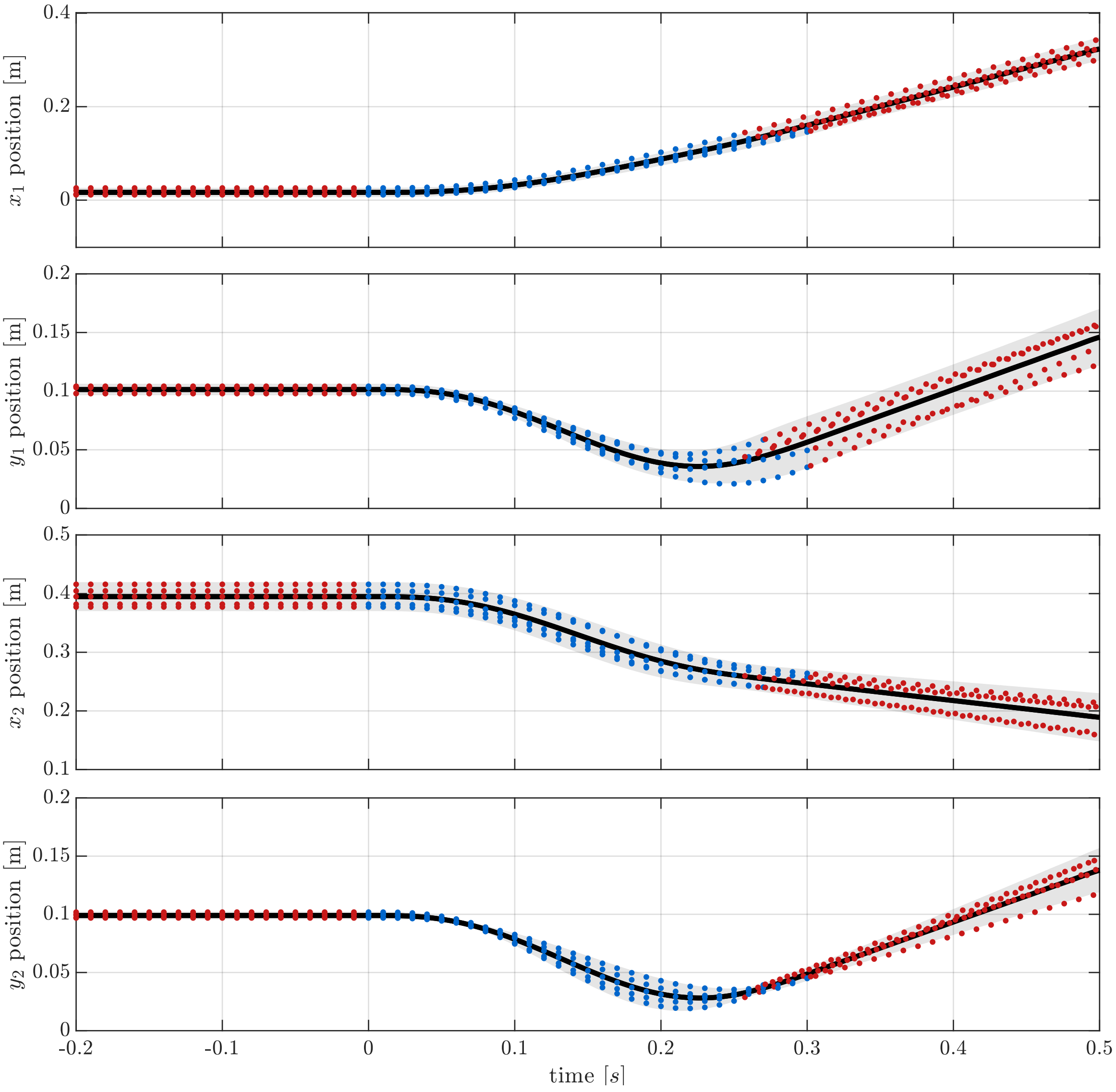


Figure D-2: The demonstration data (blue) and the extended demonstration data (red) of the position of the end effectors of five demonstrations, in the first mode. It also shows the resulting trajectory distribution with its mean as the black line and the variance in grey. The used hyper parameters are: $Z = 50$, $h = 1e - 4$ and $\lambda_{reg} = 0$. One in every ten data points is plotted for the sake of visibility.

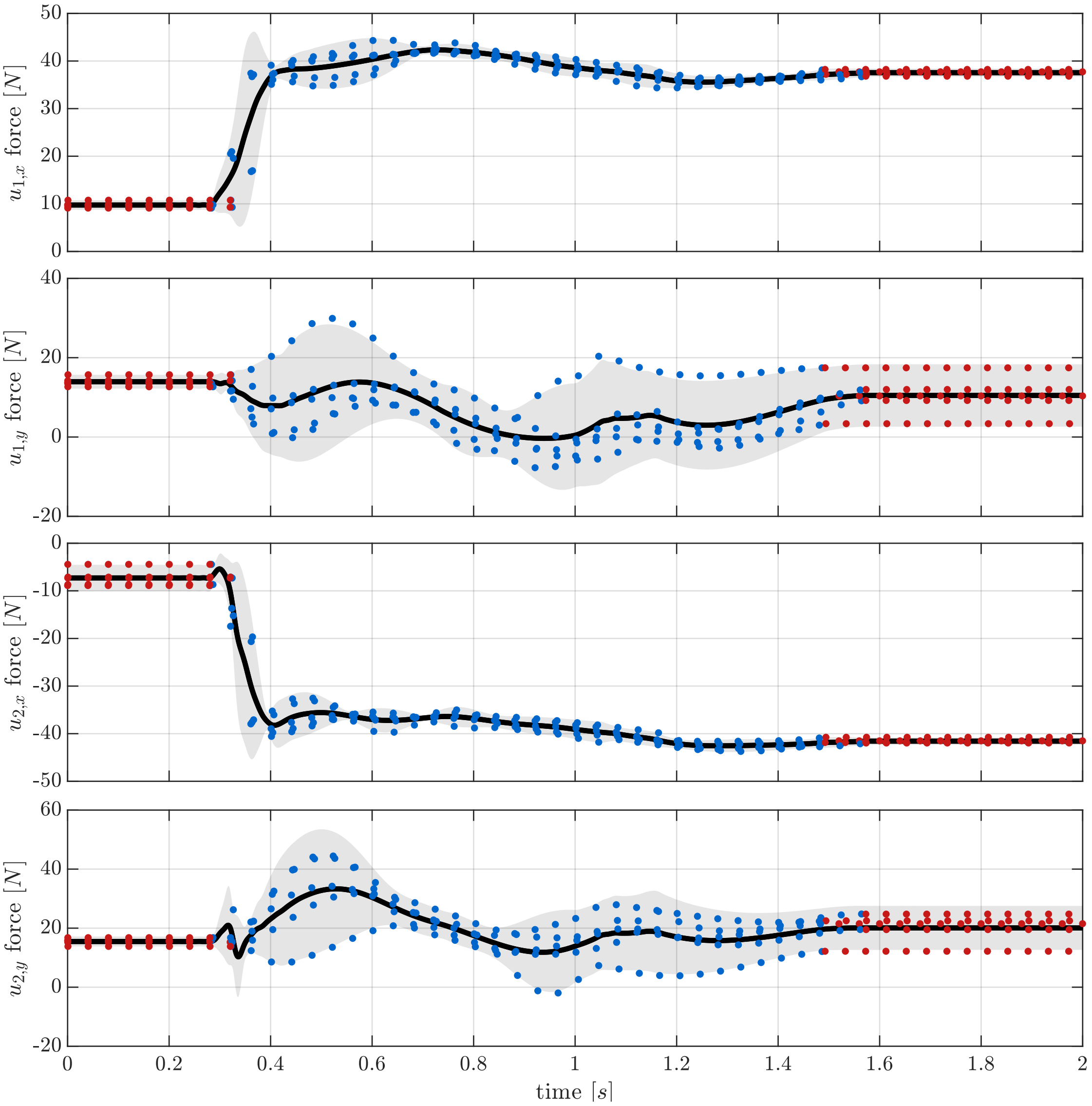


Figure D-3: The demonstration data (blue) and the extended demonstration data (red) of the input of five demonstrations, in the third mode. It also shows the resulting trajectory distribution with its mean as the black line and the variance in grey. The used hyper parameters are: $Z = 200$, $h = 1e - 4$ and $\lambda_{reg} = 0$. One in every forty data points is plotted for the sake of visibility.

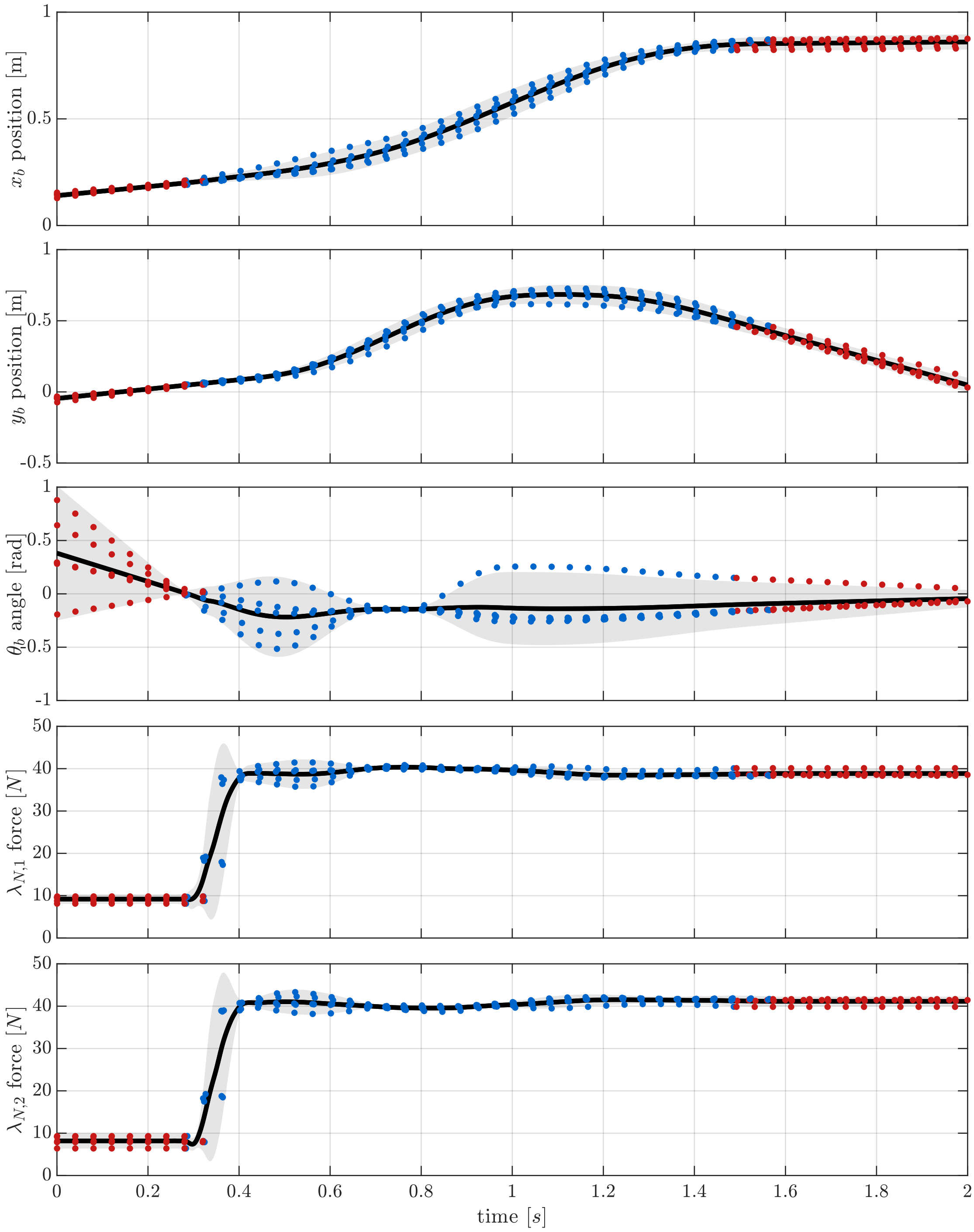


Figure D-4: The demonstration data (blue) and the extended demonstration data (red) of the position of the box and the contact forces of five demonstrations, in the third mode. It also shows the resulting trajectory distribution with its mean as the black line and the variance in grey. The used hyper parameters are: $Z = 200$, $h = 1e-4$ and $\lambda_{reg} = 0$. One in every forty data points is plotted for the sake of visibility.

Bibliography

[1] S. S. M. Salehian and A. Billard, “A dynamical-system-based approach for controlling robotic manipulators during noncontact/contact transitions,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2738–2745, 2018.

[2] S. Calinon, “Learning from demonstration (programming by demonstration),” in *Encyclopedia of Robotics*, Springer, 2019.

[3] S. S. M. Salehian, M. Khoramshahi, and A. Billard, “A dynamical system approach for softly catching a flying object: Theory and experiment,” *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 462–471, 2016.

[4] A. Robotics, “Irb 360 flexpicker.” https://new.abb.com/products/robotics/industrial-robots/irb-360, 2019. [Online; accessed 02-September-2019].

[5] S. Galeani, L. Menini, A. Potini, and A. Tornambè, “Trajectory tracking for a particle in elliptical billiards,” *International Journal of Control*, vol. 81, no. 2, pp. 189–213, 2008.

[6] J. W. Grizzle and E. R. Westervelt, “Hybrid zero dynamics of planar bipedal walking,” in *Analysis and Design of Nonlinear Control Systems*, pp. 223–237, Springer, 2008.

[7] J. B. Biemond, N. van de Wouw, W. Heemels, R. G. Sanfelice, and H. Nijmeijer, “Tracking control of mechanical systems with a unilateral position constraint inducing dissipative impacts,” in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pp. 4223–4228, IEEE, 2012.

[8] F. Forni, A. R. Teel, and L. Zaccarian, “Follow the bouncing ball: Global results on tracking and state estimation with impacts,” *IEEE Transactions on Automatic Control*, vol. 58, no. 6, pp. 1470–1485, 2013.

[9] A. Saccon, N. van de Wouw, and H. Nijmeijer, “Sensitivity analysis of hybrid systems with state jumps with application to trajectory tracking,” in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp. 3065–3070, IEEE, 2014.

[10] M. Rijnen, *Enabling motions with impacts in robotic and mechatronic systems.* PhD thesis, Eindhoven University of Technology, Dynamics and Control section, 2018.

[11] M. Rijnen, A. Saccon, and H. Nijmeijer, "Reference spreading: Tracking performance for impact trajectories of a 1dof setup," *IEEE Transactions on Control Systems Technology*, 2019.

[12] M. Riley, A. Ude, K. Wade, and C. G. Atkeson, "Enabling real-time full-body imitation: a natural way of transferring human movement to humanoids," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2, pp. 2368–2374, IEEE, 2003.

[13] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pp. 255–262, ACM, 2007.

[14] J. Kober and J. R. Peters, "Policy search for motor primitives in robotics," in *Advances in neural information processing systems*, pp. 849–856, 2009.

[15] Y. Mohammad and T. Nishida, "Tackling the correspondence problem," in *International Conference on Active Media Technology*, pp. 84–95, Springer, 2013.

[16] S. Schlott, "Teamwork between humans and robots," *ATZproduktion worldwide*, vol. 1, pp. 4–7, Jun 2008.

[17] S. Calinon and D. Lee, "Learning control," in *Humanoid Robotics: A Reference*, pp. 1–52, Springer, 2016.

[18] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, p. 21, 2017.

[19] Z. Zhu and H. Hu, "Robot learning from demonstration in robotic assembly: A survey," *Robotics*, vol. 7, no. 2, p. 17, 2018.

[20] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 2, pp. 1398–1403, IEEE, 2002.

[21] D. Reynolds, "Gaussian mixture models," in *Encyclopedia of biometrics*, pp. 827–832, Springer, 2015.

[22] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.

[23] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," *Advances in Neural Information Processing Systems*, vol. 2, no. 26, 2013.

[24] A. Del Prete, *Control of Contact Forces using Whole-Body Force and Tactile Sensors: Theory and Implementation on the iCub Humanoid Robot.* PhD thesis, Istituto Italiano di Tecnologia, 2013.

[25] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots*, vol. 42, no. 3, pp. 529–551, 2018.

[26] K. Bouyarmane and A. Kheddar, "Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations," in *2011 IEEE/RSJ international conference on intelligent robots and systems*, pp. 4414–4419, IEEE, 2011.

[27] J. Salini, V. Padois, and P. Bidaud, "Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1283–1290, IEEE, 2011.

[28] J. Vaillant, K. Bouyarmane, and A. Kheddar, "Multi-character physical and behavioral interactions controller," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 6, pp. 1650–1662, 2016.

[29] K. Bouyarmane, K. Chappellet, J. Vaillant, and A. Kheddar, "Quadratic programming for multirobot and task-space force control," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 64–77, 2018.

[30] M. Machado, P. Moreira, P. Flores, and H. M. Lankarani, "Compliant contact force models in multibody dynamics: Evolution of the hertz contact theory," *Mechanism and Machine Theory*, vol. 53, pp. 99–121, 2012.

[31] F. Pfeiffer, "The idea of complementarity in multibody dynamics," *Archive of Applied Mechanics*, vol. 72, no. 11-12, pp. 807–816, 2003.

[32] K. H. Hunt and F. R. E. Crossley, "Coefficient of restitution interpreted as damping in vibroimpact," *Journal of applied mechanics*, vol. 42, no. 2, pp. 440–445, 1975.

[33] A. S. Carvalho and J. M. Martins, "Exact restitution and generalizations for the hunt-crossley contact model," *Mechanism and Machine Theory*, vol. 139, pp. 174–194, 2019.

[34] F. Marques, P. Flores, J. P. Claro, and H. M. Lankarani, "A survey and comparison of several friction force models for dynamic analysis of multibody mechanical systems," *Nonlinear Dynamics*, vol. 86, no. 3, pp. 1407–1443, 2016.

[35] C. A. Coulomb, *Théorie des machines simples en ayant égard au frottement de leurs parties et à la roideur des cordages.* Bachelier, 1821.

[36] B. Fornberg, T. Driscoll, G. Wright, and R. Charles, "Observations on the behavior of radial basis function approximations near boundaries," *Computers and Mathematics with Applications*, vol. 43, pp. 473–490, 02 2002.

[37] J. P. Boyd, "Six strategies for defeating the runge phenomenon in gaussian radial basis functions on a finite interval," *Computers and Mathematics with Applications*, vol. 60, no. 12, pp. 3108–3122, 2010.

[38] N. Mai-Duy and T. Tran-Cong, "Approximation of function and its derivatives using radial basis function networks," *Applied Mathematical Modelling*, vol. 27, no. 3, pp. 197–220, 2003.

[39] H. Q. Dinh, G. Turk, and G. G. Slabaugh, “Reconstructing surfaces by volumetric regularization using radial basis functions,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 10, pp. 1358–1371, 2002.

[40] G. J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters, “Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks,” *Autonomous Robots*, vol. 41, no. 3, pp. 593–612, 2017.

[41] S. Calinon, I. Sardellitti, and D. G. Caldwell, “Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 249–254, IEEE, 2010.

[42] D. Bruno, S. Calinon, M. S. Malekzadeh, and D. G. Caldwell, “Learning the stiffness of a continuous soft manipulator from multiple demonstrations,” in *International Conference on Intelligent Robotics and Applications*, pp. 185–195, Springer, 2015.

[43] S. Calinon, “A tutorial on task-parameterized movement learning and retrieval,” *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1–29, 2016.

[44] Y. Wang and A. Kheddar, “Impact-friendly robust control design with task-space quadratic optimization,” in *Robotics: Science and Systems*, 2019.

[45] M. W. Spong, S. Hutchinson, M. Vidyasagar, *et al.*, *Robot modeling and control.* John Wiley and Sons, 2006.

[46] J. Kober, M. Gienger, and J. J. Steil, “Learning movement primitives for force interaction tasks,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 3192–3199, IEEE, 2015.

[47] S. Calinon, F. Guenter, and A. Billard, “On learning, representing, and generalizing a task in a humanoid robot,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, 2007.

[48] S. Calinon, “Gmm-gmr code.” http://calinon.ch/codes_prev.htm, 2009. [Online; accessed 10-December-2018].

[49] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

Glossary

List of Acronyms

DMP	Dynamic Movement Primitive
SEDS	Stable Estimator of Dynamical Systems
ProMP	Probabilistic Movement Primitive
GMR	Gaussian Mixture Regression
RBF	Radial Basis Function
FBD	Free Body Diagram
CG	Center of Gravity
DTW	Dynamic Time Warping
QP	Quadratic Programming
IA-LfD	Impact-Aware Learning from Demonstration

List of Symbols

Greek Symbols

$\boldsymbol{\alpha}$	Reference trajectory
$\boldsymbol{\epsilon}$	Inaccuracy of the ProMP approximation
γ^d	A vector containing measured data points of demonstration d
$\boldsymbol{\lambda}$	Contact forces
$\boldsymbol{\mu}_\xi$	Mean of the trajectory distribution
$\boldsymbol{\Psi}$	Matrix containing $\boldsymbol{\psi}$ for different time values
$\boldsymbol{\psi}$	Matrix containing Radial Basis Functions (RBFs)

$\boldsymbol{\Sigma}_\xi$	Covariance of the trajectory distribution
$\boldsymbol{\Sigma}_w$	Covariance matrix of the distribution of the weights
δ	Indentation variable
λ_N	Normal contact force
λ_T	Tangential contact force
λ_{reg}	A regression parameter
$\mathbf{u}^{ff}$	Feedforward
μ_w	Friction coefficient
ϕ_z	The z-th normalized RBF
τ	Expected event time
θ	The angle
v_T	Relative tangential velocity

Latin Symbols

$\boldsymbol{\xi}(t)$	A time dependent function
$\boldsymbol{e}_{rs}$	Reference spreading error
$\boldsymbol{I}$	The identity matrix
$\boldsymbol{w}$	A vector of weights
$\boldsymbol{x}$	State
${}^{A}\mathbf{R}_{B}$	Rotation matrix from B to A
$\mathbf{g}$	Gravity vector
$\mathbf{J}$	Jacobian matrix
$\mathbf{J}_c$	The contact Jacobian
$\mathbf{K}$	Feedback gain
$\mathbf{K}^d$	Derivative feedback gain
$\mathbf{K}^p$	Proportional feedback gain
$\mathbf{M}$	Mass matrix
$\mathbf{N}$	Coriolis matrix
$\mathbf{o}_A$	Origin of the world frame
$\mathbf{o}_B$	Origin of the box frame
$\mathbf{p}$	Arbitrary point
$\mathbf{q}$	Joint position
$\mathbf{S}$	Actuation matrix
$\mathbf{u}$	Input
$\mathcal{C}$	Smoothness parameter
A	The world frame
a	Discrete counter
B	The box frame
b_z	The z-th RBF
c^{qu}	Coefficient of the quintic polynomial
c_z	Center of the z-th RBF

c_{ccm}	Stiffnes coefficent of compliant contact model
D	The number of demonstrations
e	Classical error
E_{bw}	Number of data points by which data is to be backwards extended
E_{fw}	Number of data points by which data is to be forwards extended
f_w	Coulumb friction model
f_{ccm}	Compliant contact model
g_m	The m-th task
h	Width parameter of the Radial Basis Function
h_b	The height of the box
i	Macro event counter
I_j^α	Time interval between the j-th and $(j+1)$-th event
j	Event counter
k	Micro event counter
k_m^d	Derivative feedback gain of task m
k_m^p	Proportional feedback gain of task m
k_w	Variable of friction model that determines angle of slope at zero velocity
k_{ccm}	Damping coefficient of compliant contact model
L	Total number of data points
l_i	Total amount of micro events for macro event i
M	Number of tasks
N	Number of events
n^{qu}	The n^{qu}-th sequential polynomial to define a reference trajectory
Q	The number of dimensions
s_a	Time value of a-th data point
t	Regular time
t^{bw}	Time value up to which data is backwards extended
t^{fw}	Time value up to which data is forwards extended
t_i^k	Event time for micro event k of macro event i
t_j	Event time for j-th event
v	A general discrete counter variable
v_T	Relative tangential velocity
w_b	The width of the box
w_m	Weight of m-th task
x	The x coordinate
y	The y coordinate
Z	The number of RBFs
z	A general discrete counter variable
m	The mass

Subscripts

$(\cdot)_a$	Belonging to a-th data point
$(\cdot)_b$	Related to the box
$(\cdot)_{ee}$	Related to the end effectors
$(\cdot)_{max}$	Maximum inequality constraint
$(\cdot)_{min}$	Minimum inequality constraint
$(\cdot)_m$	Related to task m
$(\cdot)_x$	The x value of a vector
$(\cdot)_y$	The y value of a vector

Superscripts

$(\cdot)^{des}$	Desired quantity
$(\cdot)^d$	Belonging to demonstration d
$(\cdot)^i$	Belonging to mode with macro counter i
$(\cdot)^k$	Belonging to mode with micro counter k
$(\cdot)^{ref}$	Reference quantity
$(\cdot)^T$	The transpose of a vector or a matrix
${}^A(\cdot)$	Expressed in the world frame
${}^B(\cdot)$	Expressed in the box frame

Other

$[A]$	Orientation of the world frame
$[B]$	Orientation of the box frame
$\bar{(\cdot)}$	Extended adaptation
Δ_{im}	Small time step added such that transition of the mode takes place at the end of impact
Δ_{ode}	Time step of ode solver
Δ_{qp}	Time step of the Quadratic Programming (QP) controller
Δ_{rs}	Time extension of reference spreading
Δ_{sa}	Sampling time