Compliant control of Uni/ Multi- robotic arms with dynamical systems

THÈSE N^{O} 8131 (2018)

PRÉSENTÉE LE 11 MAI 2018

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE D'ALGORITHMES ET SYSTÈMES D'APPRENTISSAGE
PROGRAMME DOCTORAL EN ROBOTIQUE, CONTRÔLE ET SYSTÈMES INTELLIGENTS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Seyed Sina MIRRAZAVI SALEHIAN

acceptée sur proposition du jury:

Prof. J. Paik, présidente du jury
Prof. A. Billard, directrice de thèse
Prof. J. J. Steil, rapporteur
Dr M. Mistry, rapporteur
Prof. H. Bleuler, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

Abstract

Accomplishment of many interactive tasks hinges on the compliance of humans. Humans demonstrate an impressive capability of complying their behavior and more particularly their motions with the environment in everyday life. In humans, compliance emerges from different facets. For example, many daily activities involve reaching for grabbing tasks, where compliance appears in a form of coordination. Humans comply their hands' motions with each other and with that of the object not only to establish a stable contact and to control the impact force but also to overcome sensorimotor imprecisions. Even though compliance has been studied from different aspects in humans, it is primarily related to impedance control in robotics. In this thesis, we leverage the properties of autonomous dynamical systems (DS) for immediate re-planning and introduce active complaint motion generators for controlling robots in three different scenarios, where compliance does not necessarily mean impedance and hence it is not directly related to control in the force/velocity domain.

In the first part of the thesis, we propose an active compliant strategy for catching objects in flight, which is less sensitive to the timely control of the interception. The soft catching strategy consists in having the robot following the object for a short period of time. This leaves more time for the fingers to close on the object at the interception and offers more robustness than a "hard" catching method in which the hand waits for the object at the chosen interception point. We show theoretically that the resulting DS will intercept the object at the intercept point, at the right time with the desired velocity direction. Stability and convergence of the approach are assessed through Lyapunov stability theory.

In the second part, we propose a unified compliant control architecture for coordinately reaching for grabbing a moving object by a multi-arm robotic system. Due to the complexity of the task and of the system, each arm complies not only with the object's motion but also with the motion of other arms, in both task and joint spaces. At the task-space level, we propose a unified dynamical system that endows the multi-arm system with both synchronous and asynchronous behaviors and with the capability of smoothly transitioning between the two modes. At the joint space level, the compliance between the arms is achieved by introducing a centralized inverse kinematics (IK) solver under self-collision avoidance constraints; formulated as a quadratic programming problem

(QP) and solved in real-time.

In the last part, we propose a compliant dynamical system for stably transitioning from free motions to contacts. In this part, by modulating the robot's velocity in three regions, we show theoretically and empirically that the robot can (I) stably touch the contact surface (II) at a desired location, and (III) leave the surface or stop on the surface at a desired point.

Keywords: Compliant dynamical system, Coordinated motion planning, Multi-arm Control, Fast motion planning.

Résumé

L'accomplissement de nombreuses tâches interactives dépend de la compliance des humains. Les humains font preuve d'une capacité à adapter leurs comportements et plus particulièrement leurs mouvements à leur environnement. Chez l'humain, la conformité émerge de différents points de vue. Par exemple, de nombreuses activités quotidiennes impliquent la saisie d'objets, où la conformité apparaît sous forme de coordination. Les humains coordonnent les mouvements de leurs mains avec celui de l'objet, non seulement pour établir un contact stable et contrôler la force d'impact, mais aussi pour surmonter les imprécisions sensorimotrices. Même si la conformité a été étudiée sous différents aspects chez l'homme, elle est principalement liée au contrôle d'impédance en robotique. Dans cette thèse, nous utilisons les propriétés des systèmes dynamiques autonomes et introduisons des architectures de contrôle pour contrôler les robots dans trois scénarios différents, où la conformité ne signifie pas nécessairement impédance et n'est donc pas directement liée au contrôle en force ou impédance.

Dans la première partie de la thèse, nous proposons une stratégie de contrôle conforme pour la capture d'objets en vol qui est moins sensible au temps d'interception. La stratégie de capture "soft" consiste à suivre le mouvement de l'objet pendant une courte durée, laissant plus de temps aux doigts pour se fermer sur l'objet avant interception. Cela offre plus de robustesse qu'une méthode de capture "hard" dans laquelle la main attend l'objet au point d'interception choisi. Nous montrons théoriquement que le système dynamique proposé interceptera l'objet au point d'interception, au bon moment avec la direction de la vitesse désirée. La stabilité et la convergence de l'approche sont évaluées par la théorie de stabilité de Lyapunov.

Dans la deuxième partie, nous proposons une architecture de contrôle conforme unifiée permettant à un système robotisé multi-bras d'approcher et saisir de façon coordonnée un objet en mouvement. En raison de la complexité de la tâche et du système, chaque bras se conforme non seulement au mouvement de l'objet, mais aussi aux mouvements des autres bras, dans l'espace de travail et dans l'espace articulaire. Au niveau de l'espace de travail, nous proposons un système dynamique unifié qui fournit au système multi-bras à la fois des comportements synchrones et asynchrones et une capacité de transition en douceur

entre les deux modes. Au niveau de l'espace articulaire, la compliance entre les bras est obtenue en introduisant un résolveur de cinématique inverse centralisé sous contraintes d'évitement d'auto-collision; formulé comme un problème de programmation quadratique (QP) et résolu en temps réel.

Dans la dernière partie, nous proposons un système dynamique compliant permettant une transition stable de mouvements libres à des mouvements en contact avec une surface. Dans cette partie, en modulant la vitesse du robot en trois régions, nous montrons théoriquement et empiriquement que le robot peut: (I) toucher de manière stable la surface de contact, (II) au point désiré, (III) quitter la surface ou s'arrêter sur la surface à un point désiré.

Mots Clés: Système dynamique conforme, Planification de mouvement coordonnée, Contrôle multi-bras, Planification de mouvement rapide.

To my wonderful parents, lovely sister and beloved Mina.

Acknowledgments

This thesis would not have been completed without the help and support of many people to whom I am extremely grateful.

First of all, my deepest gratitude goes to my thesis director Prof. Aude Billard who gave me the great opportunity to work at the learning Algorithms and Systems lAboratory (LASA). This thesis would not have been possible without her high expectations, patience, advice and availability. Thank her for all the time she spent reading and correcting my articles and for all our long discussions.

I would like to express my very great appreciation Dr. Alireza Karimi, for his insightful comments and useful advice. Even though our collaboration was very short, his help and advice helped me a lot.

I would like to extend my gratitude to the members of my review committee: Dr. Michael Mistry, Prof. Jochen Steil, Prof. Hannes Bleuler and Prof. Jamie Paik for kindly agreeing to be a part of this committee and for the time and effort invested in their evaluation of this thesis. Their insightful comments helped me to improve the final manuscript.

Many thanks to my wonderful officemate, Nadia Figueroa, with whom I frequently collaborated during my Ph.D. research, for all our nice long discussions, her endless enthusiasm, help, smart advice and for her friendship. It was indeed a pleasure for me to share all the different steps in this Ph.D. research with her, from the beginning to the end.

I have been lucky to be surrounded by wonderful colleagues and friends in LASA who made my time in the office and outside of it more joyful. I thank Nicolas Sommer for the movie nights, coffee breaks, barbecues, Kinder surprises and for being a geek. I thank Ashwini Shukla for all his help during the first six months of my internship, his endless patience, support and for being my C++ reference. I would like to thank Klas Kronander and Iason Batzianoulis for being amazing guys. Working with them was indeed an amazing experience and having all the discussions during our lunch breaks were fantastic. I would like to thank Jose Ramon Medina Hernandez, Sahar El Khoury, Lucia Pais Ureche, Kevin Kim and Basilio Noris for all our nice discussions (partly scientific, but mostly not!) about various topics, such as different cultures, unique experiences, humor, life styles and scientific curiosity. It was indeed great to have them in the lab. I would like to thank the new generation: Murali Krishna Karnam, Leonardo Urbano, Andrew Sutcliffe, Ilaria Lauzana for all the funny conversations we have shared over lunch, coffee, and much more. I would like to thank my old

and new colleagues at LASA, Guillaume de Chambrier, Joel Rey, Ravin Luis De Souza, Ajay Kumar Tanwani, Felix Duvallet, Seungsu Kim, AJung Moon, Hang Yin and Walid Amanhoud, Jordi Bautista Ballester, Neda Taymourtash, Mahdi Khoramshahi, Sergey Voronov, Michael Bosongo Bombile, Kunpeng Yao and Anais Betsabeh Haget. I thank them for their friendship and interesting discussions. I would like to thank our secretary Joanna Erfani and the secretary of EDPR, Corinne Lebet, for taking care of the administrative aspects of my Ph.D. research.

I was fortunate to meet many wonderful people during my stay in Lausanne and to make lifelong friends. Many thanks go to the Rads (Mahdi and Saeed), Samaneh, Mehran and Rayan, Azadeh, Saleh, Esmaeil, Behnoush, the KKs (Mohammad, Soha, Mostafa, Aseye, Lego, Amir, Parastoo, Mohsen, Hadis, Shah Mehran, Zohreh and Behzad), the six-packs (Masoumeh, Mohammad, Mohammad (scientist), Elham and Mostafa), Farnaz, Marie, Amin and so many other friends that are too numerous to be mentioned here, for making my days in Lausanne memorable and unforgettably happy days of my life. Many thanks to IRSA (all the committees) for organizing many unforgettable Iranian cultural events in Lausanne. Special thanks to my friends from Volleyball, Ashkan, Amin, Vahid, Sadaf, Salman, MohammadJavad, Luara and Francesca and many others. I would also thank our trip buddies: Navid and Neda.

I would have never reached the place where I am now without the endless support, love, and encouragement of my parents. I am immensely thankful for their confidence that enabled me to independently discover many aspects of life. Many thanks go to my sister, my brother-in-law and my lovely nephew (Rodin) for all their support and the nice moments we spent together.

And last, but far from least, from the bottom of my heart, I would like to thank my lovely wife Mina, for all the moments we shared through the ups and downs of our lives. Infinite thanks go to her for being my better half and my dearest companion.

Table of Contents

Chapter 1

Introduction

"No matter what you look at, if you look at it closely enough, you are involved in the entire universe"

Micheal Faraday
1791 – 1867

1.1 Motivation

Even though robot sales have significantly increased since 2003, the sales are dominated by industries with standardized production lines, e.g., automotive and electronics industries with, respectively, 38% and 25% of the total sales in 2015 (Economist, 2017a,b; IFR, 2016). The applications in these factories mostly involve moving the manipulators with high speed on predefined paths. Hence, the factories' environments need to be accurately controlled so that even blindfolded robotic systems can reliably perform repetitive task descriptions; Figure 1.1a. These applications are consequently very sensitive to uncertainties in that any unforeseen circumstances might result in a total shutdown of the production lines.

Whereas, human workers demonstrate an impressive capability of complying with the environment's states (Figure 1.1b), hence, they are capable of manipulating objects in dynamically changing environments. By complying with the states of the environment, humans are able to adapt their motions with the changes caused by perturbations and uncertainties. In this way, unlike traditional robots, humans accomplish tasks by determining the desired behavior, on the fly, based on the observations of the current state of the environment.

Even though humans are not perfect and are slower and less powerful than robotic systems, collaborations between human operators and robotic systems improve the productivity of both sides (Economist, 2013). Thus, hybrid production lines, where the robots work together with humans, are commonly considered as the future of automation (Economist, 2017c); Figure 1.1c. In such production lines, robotic systems that exhibit the same level of compliance and human-like reactivity and adaptivity are essential.

(a) ©Telegraph **(b)** ©Independent **(c)** ©BMW Group

Figure 1.1: Examples of different types of production lines. 1.1a and 1.1b illustrate examples of fully-automated versus fully-manual tasks in a car manufacturer, respectively. In 1.1c, an example of hybrid production line, where a seamless collaboration between humans and robots is essential, is shown.

Generally, compliance relates to the circumstances whereby a person acts in accordance with a wish or a command (Billard, 2017). Hence, compliance can be interpreted in different ways. For example, many activities in industry involve reaching for objects on a conveyor belt, where compliance appears as a form of coordination. Humans need to comply with the state of the object such that their hand motions are coordinated with that of the object, to not only establish a stable contact and control the force of impact but also to grab the object at a desired location. If the object is too big or too heavy, human workers have to use both hands, where both hands should be in compliance and coordination with the object, as well as with each other.

Even though compliance has been studied from different aspects in humans[1], in robotics, it is primarily related to the behavior of systems in contact (Brock et al., 2008). Broadly, compliance in robotics can be categorized in two levels, passive and active (Wang et al., 1998; Albu-Schäffer et al., 2007). In the passive, a compliant behavior emerges from the inherent elasticity of mechanical linkages/joints. However, in the active compliance, the controller of the actuators imitates the behavior of a spring, i.e., impedance control.

Our main goal in this thesis is to devise actively compliant motion generators for controlling robots in scenarios where (i) there is no mechanical compliance or it is negligible, in comparison with the interaction forces, and (ii) the impact lasts less than a millisecond, leaving no time for the sensors to measure it or for the robots to properly react to it; hence, impedance controllers are either insufficient or practically inapplicable. In this thesis, we explain how compliance can be provided through active control of the arm motions at the kinematic level, i.e., the state level[2]. By defining compliant motion generators, the robots are able to adapt, sufficiently in advance, their motions with respect to the state of the environment. This leaves more time for re-adjustments and compensating for sensory-motor noise. To this end, we use the idea of Dynamical Systems (DSs) for immediate re-planning and introduce motion generators that are coupled

[1] i.e., Mechanical, cognitive and social (Billard, 2017) .

[2] In this thesis, state refers to the position, velocity and acceleration in either joint or task spaces.

with the state of the environment that can be the position/velocity of moving or fixed objects or other robots.

1.2 Problem Statement and Approach

Our ultimate goal in this thesis is to accomplish interactive tasks where defining compliance in the traditional force/velocity domain is, otherwise, either insufficient or inapplicable. Hence, we study compliance in the motion domain, where the motion of a robot actively complies with the state of the environment. We study compliance in three different exemplary tasks, where the accomplishment of the task hinges on the robot's compliance with (i) the state of a moving object, (ii) the state of a moving object and other robots, and (iii) the state of a fixed surface.

In the first part of the thesis, we address compliance in the scenario of catching objects in flight. The compliant-catching strategy consists of having the robot move with the object for a short period of time before grabbing it. In this scenario, compliance appears as a form of softness. This leaves more time for the fingers to close on the object at the interception and offers more robustness than the "hard" catching method, in which the hand stops at the interception point.

In the second part of the thesis, we address compliance in the scenario of reaching a moving object by a multi-arm robotic system. In this scenario, compliance needs to be addressed at two sub-levels, specifically at the arm and the object levels. At the arm level, the motion of each arm must comply with the motion of the other arms in both: the task and joint spaces, so that any collision can be avoided. At the object level, the resultant motion of the arms must comply with that of the object, such that the object is simultaneously intercepted by all the arms. In this scenario, compliance appears as a form of coordination.

In the third part of the thesis, we address compliance in non-contact/contact transition scenarios. Compliant transition consists of adjusting the velocity of the robot at the impact, so that it does not bounce on the surface. In this scenario, compliance appears as a from of modulation. The approaches devised in this thesis can be broadly categorized in the dynamical-system-based motion generators, described in detail in Section 1.2.1.

1.2.1 Dynamical systems

Close your eyes and ask your colleague to place an apple in front of you. Try grabbing it in one shot. Obviously you were not successful as you do not know its precise position. Open your eyes, locate the apple's position and close them again. Try reaching the apple while your colleague is re-locating it. Note, you cannot accomplish the task as the apple's state is dynamically changing

while your perceived information is not. However, if you see the apple, you can undoubtedly grab it.

Perceiving the states of the environments is necessary for identifying the task's goals and achieving them in dynamically changing environments. However, there is a gap between high-level task goals and low-level robot's motions. Robots, in general, are made of electrical motors that do not understand anything except voltage and ampere. Hence, we need to translate the task's goals into commands interpretable by the joint controllers.

In user-friendly robotic systems, human operators are not required to write down complicated time/space functions to exactly translate the goals into joint-level motions. Whereas, the system must be able to generate them based on the high-level goals, e.g., the states of the initial, target or even obstacles. Motion-planning and trajectory-planning approaches are the main means for converting high-level task specifications into low-level joint-level movements. During the last four decades of automation, many different architectures have been proposed for specifying the detailed motion of robotic systems; e.g., potential-field (Khatib, 1986), randomly exploring random trees (LaValle and Kuffner Jr, 2000), polynomial (Craig, 1989), optimal control (Constantinescu and Croft, 2000; Bäuml et al., 2010) and dynamical systems (Khansari-Zadeh and Billard, 2011; Schaal et al., 2000). A more comprehensive review of these methods is provided in Section 2.1.

Dynamical systems have been successful in modeling human motions (Bullock and Grossberg, 1988; Diedrichsen and Dowling, 2009; Schöner and Kelso, 1988) and in generating robot motions (Katoh and Mori, 1984; Hollerbach, 1984; Schaal et al., 2000; Brady, 1982) for decades. The main benefit of using DSs can be summarized in the immediate re-planning property, i.e., the ability of countering perturbations in real-time. Dynamical systems do not generate the entire path at once rather the next action, based on the current observations, e.g., the time, states of the system, environment or measured forces. Hence, the desired behavior of the system is calculated instantaneously, based on the current measurements. Consequently, perturbations are, indirectly, compensated by considering their effects on the calculation of the next action of the robot. Moreover, if a dynamical system is defined properly, one can ensure its convergence to an attractor or a goal, regardless of the robot's initial configurations, i.e., asymptotical stability.

For the aforementioned reasons, throughout the thesis, we exploit dynamical-system-based control laws in order to generate the robots' motions. To achieve the desired level of compliant behavior, the dynamical systems are coupled with the states of the environments, specifically the state of the object in the first scenario, the state of the object and the other robots in the second scenario, and the distance between the robot and the contact surface in the third scenario.

1.3 Main Contributions and Thesis Outline

This thesis is composed of a number of chapters that are structured in accordance with the three main scenarios outlined in the previous section. Brief overviews of each chapter, as well as the corresponding contributions, are outlined as follows:

Chapter 2- Background and Related Work

In this chapter, we comprehensively review the state-of-art approaches for single/multi-arm motion planning, with the main emphasis on the aforementioned scenarios, is presented in this chapter.

Chapter 3 - Softly Intercepting a moving object with a robotic arm

This chapter presents our work on softly catching flying-objects. In order to successfully softly catch an object , the arm must intercept the object on time, at the right place, and with a specific velocity aligned with that of the object; see Figure 1.2. The first part of this chapter presents our control law that is formulated as a Linear Parameter Varying (LPV) based dynamical system for generating the soft catching motion. Furthermore, with the purpose of maximizing the softness at the interception subject to the kinematic constraints of the robot, a closed-loop optimal-control problem is suggested. Then, we present our probabilistic approach for estimating the parameters of the LPV system that use Gaussian Mixture Models (GMM), to account for the inherent stochasticity of the training data-points. In this scenario, as catching is an extremely rapid action, the training data-set should be a representative of the fastest kinematically feasible motions of the robot. As it would be difficult to have these provided by a human expert, as kinesthetic teaching would not make it possible to move the arm at its maximal speed, we present our algorithm for generating the training data-set.

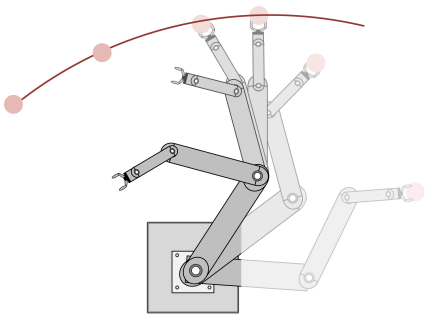


Figure 1.2: Schematic of catching softly (compliantly) a ball. The arm meets the object at the *intercept* point and moves with it for a short period of time while slowly reduces its velocity. See Chapter 3 for more information.

We theoretically prove that the generated soft catching motion, and consequently the end-effector, intercept the object's trajectory at the desired intercept point with the desired velocity aligned with that of the object. The performance of our proposed method is validated in a real-world experiment with KUKA LBR IIWA (7 degrees-of-freedom arm robot) mounted with a 16 DOF Allegro hand.

Chapter 4 - Coordinated Multi-Arm Motion Planning

This chapter presents our work on coordinated multi-arm motion planning for reaching a moving object. In order to successfully reach and intercept an object with a multi-arm system, the compliance in two sub-levels needs to be addressed. First, each arm should comply with the motion of the other arms in both the joint and task spaces, as the robots should avoid self-collisions at all times. Second, the resultant motion of the robots should comply with the object's motion so that they reach and simultaneously intercept it at the desired points. Moreover, a multi-arm system could provide not only synchronous behaviors, as the one mentioned previously, but also asynchronous behaviors, where each robot follows its own goal-oriented tasks, specifically independent point-to-point reaching motions.

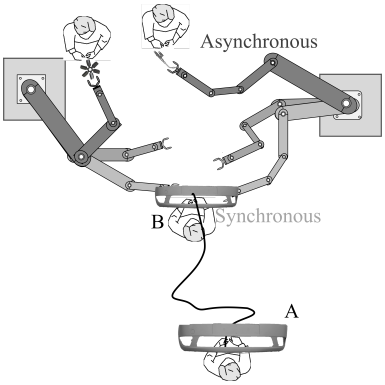


Figure 1.3: Schematic of transition between Synchronous and Asynchronous behaviors. See Chapter 4 for more information.

The first part of this chapter presents our unified control architecture that generates two types of behaviors: (i) multi-arm asynchronous task-space behaviors, where each robot has its own target and (ii) multi-arm synchronous task-space behaviors, where the robots' tasks are to reach, with all the arms, for a moving object; see Figure 1.3. To provide a smooth transition between these behaviors, a notion of synchronization allocation is introduced. Given the motion of the object and the joint workspace of the multi-arm system, each arm is continuously allocated to a desired behavior. While allocated to the synchronous behavior, the robots are taken over by a virtual object based dynamical system, expressed as a LPV system. The motion of the virtual object is coupled with the motions of the robots and the real object. While allocated to the asynchronous behavior, the robots are controlled by independent target-oriented dynamical systems. In the proposed unified architecture, both behaviors are encoded in a single dynamical system. In the second part of the chapter, compliance in the joint-space is provided by introducing a centralized inverse kinematics (IK) solver under self-collision avoidance constraints; formulated as a quadratic program (QP) problem subject to linear equality and inequality constraints which can be solved in real-time.
We show theoretically that the proposed architecture is capable of generating both behaviors, and the system remains stable during transitions. Moreover, we show that the overall closed-loop system, including the IK solver, is passive. We validate the framework on two dual-arm robotic systems. The results demonstrate that the control architecture can adapt the motion of each arm within milliseconds, even when the motion of the object is fast and not accurately predictable.

Chapter 5 - Stable Non-contact/Contact Transitions

In this chapter, we present our work on controlling robotic manipulators during non-contact/contact transitions. Unlike in Chapter 3, in this chapter, we study scenarios where the impact forces can be strong. Hence, controlling for a smooth transition from free motion to contact is crucial as incurring a strong force of impact might lead to unstable contact with the robot bouncing on the surface, i.e., chattering. In order to successfully establish a stable contact with a rigid surface, the robot should touch the surface with zero, or near to zero, velocity such that the post-contact velocity, in the normal direction to the surface, is zero, i.e., the robot remains in contact with the surface. Accordingly, in this thesis, we call a contact stable if the impact occurs only one time and the robot remains in contact with the surface after the impact. In this chapter, we present our strategy in which the robot adapts its motion before entering into the contact, i.e., its speed is modulated to align with the surface. The robot then slides on the surface while controlling the velocity against the surface. We exploit the properties of autonomous dynamical systems to enable on-line controlling of the robot, and we exploit local modulations of DS to control the smooth transitions upon contact.

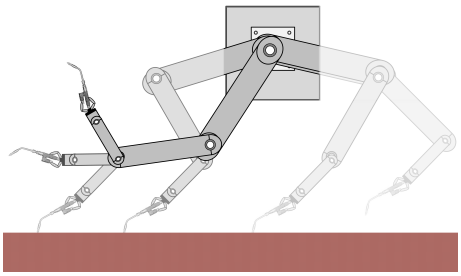


Figure 1.4: Schematic of a stable contact with a surface in an interactive task. The robot gets into the contact at the desired location such that the impact happens only one time and the robot remains in the contact with the surface after the impact. Moreover, the robot should be able to leave the surface at a desired departure point. See Chapter 5 for more information.

We show theoretically and empirically that by using the modulation framework, the robot can (i) stably touch the contact surface (ii) at a desired location, and finally (iii) leave the surface or stop on the surface at a desired point. We validate the performance of the framework with a set of simulations and in a real-world experiments with KUKA LBR IIWA.

Chapter 6 - Conclusion

In the final chapter, we summarize our technical achievements, our contributions and their limitations. We further discuss the possible research directions and future works.

Appendices

In order to improve the readability of the main text and not overload it with mathematical equations, we collect the proofs of theorems in the Ap-

pendices. In the main text of the thesis, we will refer to them when they are used.

1.4 Publications and Source Codes

Most of the material of this thesis was published in peer-reviewed conferences and journals. Large portions of the materials presented in Chapter 3 were published in (Mirrazavi Salehian et al., 2016a). The contents of Chapter 4, multi-arm motion-planning architectures, were published in (Mirrazavi Salehian et al., 2016b, 2017b,a). The contents of Chapter 5 are currently under submission. Furthermore, the videos of the robot experiments and the codes, which were recored and developed during the last four years and reported in this thesis, are available on-line in the following tables:

Table 1.1: The implementation toolboxes which are provided by the authors are available in the following links.

https://github.com/epfl-lasa
https://github.com/sinamr66

Table 1.2: The links of the videos of the robot experiments. All the videos are also available on LASA's You-tube channel at https://www.youtube.com/channel/UCqnvGUfdlr94mddDQamEBGA/videos.

Chapter 3
http://lasa.epfl.ch/files/Sina_Mirrazavi_Thesis_videos/Chapter_3/Main.mp4 http://lasa.epfl.ch/files/Sina_Mirrazavi_Thesis_videos/Chapter_3/Systematic_Assessment.mp4
Chapter 4
http://lasa.epfl.ch/files/Sina_Mirrazavi_Thesis_videos/Chapter_4/Main_1.mp4 http://lasa.epfl.ch/files/Sina_Mirrazavi_Thesis_videos/Chapter_4/Main_2.mp4 http://lasa.epfl.ch/files/Sina_Mirrazavi_Thesis_videos/Chapter_4/Systematic_Assessment.mp4
Chapter 5
http://lasa.epfl.ch/files/Sina_Mirrazavi_Thesis_videos/Chapter_5/Main.mp4 http://lasa.epfl.ch/files/Sina_Mirrazavi_Thesis_videos/Chapter_5/Systematic_Assessment.mp4

Chapter 2

Background

آن کس که نداند و بداند که نداند گنگان خرک خویش به منزل برساند

"The one who does not know, but knows that he doesn't know, will be successful."

Ibn-e Yamin
circa 1285 – 1368

A fundamental problem with robotic systems is planning their motions from the initial state till the target state while satisfying some path/trajectory constraints, e.g., avoiding obstacles. Hence, motion-planning has been an active research topic for decades and a vast variety of algorithms/approaches have been proposed for tackling different aspects of this problem, e.g., obstacle avoidance, kinematics/dynamics constraints, sensory noises or differential constraints. In this section, we first provide an overview of motion planning methods in Section 2.1. Then, we comprehensively review the existing approaches that address the challenges of (i) catching objects in flights in Section 2.2, (ii) multi-arm manipulations in Section 2.3 and (iii) non-contact/contact transitions is provided in Section 2.4.

2.1 Motion Planning: Overview and Historical Perspective

Motion planning is one of the first problems that has been addressed in robotics (Paul, 1971, 1972). Motion planning is primarily defined as a problem of finding a collision-free path for robotic systems, from an initial to a goal configuration among a collection of static obstacles; it has attracted considerable attention in the last four decades (Kavraki and LaValle, 2008)[1]. Assuming the

[1] Path planning is mainly concerned with finding a collision-free path regardless of its feasibility. Whereas, Kinodynamic planning approaches consider the kinematic/dynamic constraints of the robots as well (Elbanhawi and Simic, 2014).

obstacles are static and their shapes and locations are known, a motion-planning problem can be formulated as a geometrical problem: finding a continuous path in the free space such that the robot reaches the final target for a given initial configuration (Udupa, 1977; Lozano-Perez, 1983). However, these approaches are impractical as calculating the free/obstacle spaces is computationally hard (Reif, 1979).

An alternative paradigm is sampling-based algorithms that do not require the exact geometric representation of the spaces. Sampling-based approaches exploit the idea of randomly sampling the configuration space (C-space) (Elbanhawi and Simic, 2014). Extensive surveys on sampling-based methods are provided in (Elbanhawi and Simic, 2014; Al-Bluwi et al., 2012; Lindemann and LaValle, 2005; Tsianos et al., 2007). Early approaches include the Randomized Potential Planner (RPP) (Barraquand and Latombe, 1991; Carpin and Pillonetto, 2003) that uses random walks to escape local minimums. In the same line, the Probabilistic Roadmap Method (PRM) (Amato and Wu, 1996; Kavraki et al., 1996; Švestka and Overmars, 1997) was proposed, as a probabilistic complete algorithm[2] (Hsu et al., 2006) for generating a probabilistic roadmap. In the learning phase, the configuration space is randomly sampled and the collision-free nodes are connected via a local-motion planner. Then, in the query phase, the initial and the target configurations are defined and connected to the roadmap. The learning and query phases are unified in the Rapidly exploring Random Trees (RRT) algorithm (LaValle, 1998), where the tree is incrementally grown from the initial to the target configurations by randomly sampling the space. Other notable sampling-based algorithms are Expansive Space Trees (EST) (Hsu et al., 1997), Sampling-based Roadmap of Trees (SRT) (Plaku et al., 2005) and Adriande's Clew (Ahuactzin et al., 1998). In general, the sampling-based algorithms are not complete algorithms. In other words, finding a solution either depends on a given sufficient runtime or is not guaranteed (Elbanhawi and Simic, 2014; Kavraki and LaValle, 2008). Combinatorial algorithms address this shortcoming for a narrow class of problems. Combinatorial approaches find paths through the continuous configuration space without resorting to an approximation, which implies their completeness (LaValle, 2006).

Early approaches, which have been widely used in industrialized scenarios, addressed the problem of moving the arm between user-specified states[3] by manually moving the manipulator on the desired trajectory, and then saving the corresponding joints' angles in a memory unit. However, this requires a large number of closely spaced points and the execution speed is considerably slow (Castain and Paul, 1984). An alternative approach uses polynomials to provide interpolated points between user-specified states at either the joint space (Castain and Paul, 1984; Lin et al., 1983) or the task space (Paul, 1979; Taylor,

[2] Probabilistically completeness means that the probability of producing a solution approaches 1 as more time is spent.

[3] Namely the initial, the final and possible intermediate states.

1979; Hong and Slotine, 1995). Polynomials enable the user to specify not only the desired positions but also velocities and accelerations. As the path generated by a polynomial is continuous and smooth, there is no need to bring the manipulator to rest at each of the intermediate points. Furthermore, a smaller storage space is required as only the coefficients of the polynomial are needed to be stored off-line. The use of polynomial-based motion-planning approaches is not limited to the early works. For example, (Chu et al., 2017) proposed a polynomial-based motion generator for maneuvering a multi-arm robotic system that avoids collisions in a known environment.

In all the aforementioned methods, the obstacles are assumed to be static or their motions are predictable. However, this is a very restrictive assumption in the real-world scenarios. Hence, to accomplish these scenarios, some form of feedback is necessary. Integrating feedback in the continuous state-space requires defining a navigation function (LaValle, 2006). Given the current state of the robot and a navigation function, a motion planner determines the next desired action. As the main challenge is defining a proper navigation function, a wide range of methods have been proposed for computing these functions.

By filling the robot's workspace with an artificial, attractive/repulsive velocity-field around the goal and obstacles, respectively, (Khatib, 1986) introduces the potential-field-based motion generator. Mathematical elegance and simplicity are the main features of this method. But, this approach has inherent problems, specifically, getting stuck in local minima, no passage between narrow spaces, oscillations in the presence of obstacles or in narrow passages (Koren and Borenstein, 1991). Since then, different approaches have been proposed to address these shortcomings. (Ge and Cui, 2000) propose a repulsive potential function to avoid local minima in a scenario where the goal and the obstacles are near each other. The local minima caused by the symmetrically aligned robot-obstacle-goal is addressed in (Lee et al., 2012). By defining virtual obstacles, (Chengqing et al., 2000) reshapes non-convex obstacles to convex ones. Potential fields with respect to the robot's kinematic constraints are discussed in (Lau et al., 2015). The other types of local minimums are avoided by employing harmonic potential-fields (Rimon and Koditschek, 1992; Koditschek and Rimon, 1990; Barraquand et al., 1991). In order to avoid oscillations, (Ren et al., 2006) added the Hessian matrix to the force field. However, due to the computation cost, the utility of the potential functions remains limited to low-dimension state spaces.

Navigation function can be constructed by concatenating local potential functions by sequencing a set of overlapping funnels (Choi and Latombe, 1991; Burridge et al., 1999; Conner et al., 2006). Based on the state, a corresponding potential function guides the robot to the next funnel. However, as mentioned in the previous paragraph, this method has not been applied to high-dimensional state spaces, e.g., manipulation task (Caccavale and Uchiyama, 2008). By assuming that the uncertainties in the dynamics of the system are bounded, (Ma-

jumdar and Tedrake, 2017) recently proposed a two-phase (off-line and on-line) method for maneuvering a small airplane while avoiding obstacles. The off-line phase consists of constructing a library of different funnels within which the state is guaranteed to remain. Then, in the on-line phase, to find a collision-free funnel, this library is augmented with the real-time feedback from the location of the obstacles .

By taking into account the task objectives and the kinematic/dynamic limitations of the robot, Model Predictive Control (MPC) can address the shortcoming of the mentioned methods (Fox et al., 1997; Brock and Khatib, 1999; Ögren and Leonard, 2005; Svenstrup et al., 2010; Schultz and Mombaur, 2010). MPC, also called Receding Horizon Control (RHC), exploits the benefits of both feed-forward and feedback controls by computing the best action with respect to the known events within a time horizon. If the prediction horizon lasts till the end of the motion, MPC can be formulated as a optimal control problem; which can be used for optimizing the motion duration (Smeets and Brenner, 1995; Shin and McKay, 1985) or actuator efforts (Chettibi et al., 2004). Compared to the Control Lyapunov Function (CLF) framework, a stability-orientated control approach, MPC is primarily a performance-oriented local-control approach (Primbs et al., 1999; Ögren and Leonard, 2005)[4]. Moreover, the performance of MPC based architectures depends on the availability of the robot's and environment's dynamics. Another main challenge is its computational complexity. MPC requires that a constrained optimization problem be solved at each time step.

2.1.1 Dynamical systems: overview

Unlike MPC, dynamical systems compute the next action of the robot by considering only its current state in a closed-form solution. Hence, they are computationally more efficient and can be used in fast and adaptive applications. As DSs are formulated as a set of autonomous or non-autonomous differential equitations, they can technically approximate a large set of movement primitives that can be very simple motions, such as reaching to a fixed target (Khansari-Zadeh and Billard, 2011), or more complex tasks such as flipping a pancake (Kormushev et al., 2010), peeling a zucchini (Figueroa and Billard, 2017) or playing table tennis (Paraschos et al., 2013).

DSs are widely used in modeling, not only for the motion of robotic systems but also human motions, e.g., for reaching fixed (Bullock and Grossberg, 1988; Berret et al., 2011) or moving (Lee et al., 1983; Dessing et al., 2002; Peper et al., 1994) targets. As an example, in the catching scenario, the prospective architecture suggests that the hand motions are continuously updated till the

[4]It is worth mentioning that, stability and convergence of MPC-based algorithms have been addressed in the recent years (Faulwasser and Findeisen, 2016; Alessandretti et al., 2013; Mayne et al., 2000).

interception, based on the continuous sensory inputs, specifically, the distance to the object (Peper et al., 1994) and the object's velocity (Carnahan and McFadyen, 1996; Smeets and Brenner, 1995).

A wide variety of architectures have been proposed for dynamical systems. Neural networks (NN) and its variant are proposed for modeling discrete, rhythmic, and reaching behaviors (Pearlmutter, 1988; Sudareshan and Condarcure, 1998; Lukoševičius and Jaeger, 2009; Pearlmutter, 2008). (Reinhart and Steil, 2011) proposed a recurrent network that integrates the forward and inverse model of a robot. Despite the capability of neural networks to approximate very complex motions, the stability of the neural-network-based dynamical systems are studied for only one (Lemme et al., 2014; Neumann et al., 2013) or two (Garcia-Lopez et al., 2017) hidden-layer networks. Moreover, training a neural network requires solving a non-convex problem, i.e., it is typically computationally expensive and the performance is sensitive to the initialization.

The Dynamic Movement Primitives (DMP) approach (Ijspeert et al., 2002; Schaal, 2006) provides an accurate and globally stable non-linear approximation of a set of demonstrations. DMP consists of two parts, a linear globally stable DS and a non-linear regression part. Two parts are coupled through a phase variable which is strictly decreasing to zero and controls the influence of the non-linear part over the linear part. Hence, even though DMP is not non-autonomous, it is inherently time-dependent, as the phase variable distorts the temporal pattern of the dynamic.

In contrast to non-autonomous DSs, autonomous dynamical systems are defined and act on the state of the system and the environment. Formulating autonomous DSs as a non-linear combination of linear models enables the use of many tools from the linear-systems theory for analysis and control. Two well-known formulations of these types DSs are Gaussian Mixture Regression (Gribovskaya et al., 2010; Calinon et al., 2007; Lee and Nakamura, 2007) and the Hidden Semi-Markov model (Calinon et al., 2011).[5] By adding constraints on each of these linear systems, (Khansari-Zadeh and Billard, 2011) propose a globally stable autonomous DS. This is further extended/improved to accomplish hitting tasks (Khansari-Zadeh et al., 2012; Kronander et al., 2011) and catching tasks (Kim et al., 2010, 2014). (Shukla and Billard, 2012b) propose a multi-attractor dynamical system that is based on the formulation of Support Vector Regression (SVR). Given the initial state of the system, the authors show that the generated motion is locally asymptotically stable to the attractors within a finite region of attraction. (Hersch et al., 2008; Hersch and Billard, 2008) integrate two DSs, which act simultaneously on joint and task spaces, to control the motion of humanoids. By modulating the generated motion of DSs, (Khansari-Zadeh and Billard, 2012) propose an obstacle-avoidance controller for

[5] It is worth noting that encoding demonstrated motions by using these tools were proposed before the introduction of DSs (Yang et al., 1997; Tso and Liu, 1996). Extensive surveys on learning from demonstrations are available in (Hussein et al., 2017; Billard et al., 2016).

avoiding convex shaped objects. This is further extended in (Kronander et al., 2015) and (Sommer et al., 2017) for locally reshaping an existing stable DS with internal and external signals, respectively.

In recent years, various new formulations for constructing/approximating autonomous DSs have been proposed. (Blocher et al., 2017) exploits the contraction theory and propose GMR-based dynamical system that contracts to a reference trajectory. The framework proposed by (Lemme et al., 2014, 2013) improves the re-reproducibility of the data-driven models by relaxing the quadratic condition of the Lyapunov function. (Neumann and Steil, 2015; Perrin and Schlehuber-Caissier, 2016) propose a diffeomorphic transformation for approximating global asymptotic stability DSs. Given a demonstration and a reference globally asymptotically stable DS, the authors compute a diffeomorphic transformation that maps the reference DS onto the demonstration. (Umlauft et al., 2017) propose a Gaussian-process state-space model and use a data-driven Lyapunov function to stabilize it.

In this section, we have provided an overview of motion planning approaches. However, given the variety of the approaches, comprehensively reviewing all of these methods is beyond the scope of this thesis. In the following sections, we will exclusively review the exiting approaches for accomplishing the scenarios mentioned in Section 1.2.

2.2 Soft Catching a Moving Object

The problem of catching fast-flight objects has attracted a lot of attention during the last three decades, starting with the early works of (Lin et al., 1989) and (Hove and Slotine, 1991). To successfully catch an object, the arm must be at a specific place in time. The complexity of the problem increases significantly if the object must be caught softly. In this case, the arm must intercept the object on time, at the right place, and with a specific velocity. Planning a fast motion with equality point constraints, such as in catching or hitting a flying object, has been extensively studied in the literature. Fitting a polynomial trajectory with different orders[6] to predetermined points –initial, intercept and stop– is proposed and used in (Hong and Slotine, 1995; Lippiello et al., 2013; Lippiello and Ruggiero, 2012b,a; Senoo et al., 2006; Namiki and Ishikawa, 2003; Zhang and Buehler, 1994; Buttazzo et al., 1994; Frese et al., 2001; Riley and Atkeson, 2002; Kober et al., 2012; Nishiwaki et al., 1997; Bätz et al., 2010; Uchiyama et al., 2012; Cigliano et al., 2015; Murakami et al., 2015). Similarly, (Frank et al., 2008) and (Lin and Chiu, 2008; Smith and Christensen, 2007) define a sigmoid and linear bang-bang function between the initial and the final points, respectively. Even though these methods are computationally efficient and can meet the terminal constraints, they utilize time as an explicit variable. As a

[6] Including zeroth and first.

result, they are highly sensitive to an imprecise estimate of the catching position and time, and would require a complete re-planning as a new desired intercept point is provided. Besides, there is no guarantee that the generated trajectory can be tracked by the robot.

Optimal control or calculus of variation can address some of these issues, see (Bäuml et al., 2010; Bäuml et al., 2011; Hujic et al., 1998; Croft et al., 1995; Lampariello et al., 2011), by taking into account the terminal constraints and the kinematic limitations of the robot. However, these are iterative procedures with no guarantee to converge in a small enough number of time steps to ensure extremely fast reactivity. Also, the convergence is highly dependent on the initial guess.

Imitation learning is also applicable to this problem, see (Park et al., 2009; Kim et al., 2014; Schaal et al., 1996; Hamon, 2011). By using machine learning techniques, one can construct an expert's set of demonstrations and use this to model the requested dynamics of motion. If human demonstrations are provided directly, through kinesthetic teaching, this ensures that the motions are kinematically feasible (Kim et al., 2014). However, the demonstrations can never be exhaustive and extrapolation is prone to error.

Recently, (Senoo et al., 2016; Koike et al., 2016; Senoo et al., 2017) proposed an impedance controller for achieving softness in the catching scenario. For reducing the impact at the interception, the proposed controller drives back the robot with respect to the measured/estimated impact force. However, due to the real-world constraints (e.g., the bandwidth of force sensors and actuators) this approach has been verified by a set of simulations.

In Chapter 3, we leverage the properties of autonomous DSs for immediate re-planning and tailored a dynamical system which can intercept the object at the desired point with the desired velocity aligned with that of the object. The DS is expressed as a Linear Parameter Varying (LPV) system subject to stability constraints. We show theoretically and empirically that by using the proposed dynamical system, an object can be softly caught at the desired intercept point.

2.3 Multi Arm Coordination at Task and Joint Spaces

The use of multi-arm system allows for manipulation of heavy or large objects. As introduced in Section 1.2, we select the task of reaching a moving object by a multi-arm robotic system. To accomplish this task, each arm should move in coordination with the rest of the other robotic arms such that any collision is avoided in both joint and end-effector levels. Moreover, the resultant motion of the arms must be coordinated with that of the object so that the object is intercepted at the desired location by all the arms.

Although coordinating multiple robotic arms for object manipulation has been extensively studied in robotics literature, most effort has primarily focused on devising strategies for coordinated manipulation of static objects that are partially or fully grasped by the multi-arm system. (Vahrenkamp et al., 2012, 2010) proposed a RRT-based algorithm to generate collision free motions to grasp a not-moving object with a humanoid robot. Given its search-based strategy, this approach can guarantee feasible grasps by both arms while satisfying the self-collision avoidance constraints. However, due to its computational complexity and the fact that it cannot guarantee simultaneous interception of the object by all the arms, it becomes inadequate when trying to reach for a *moving* object. (Chung and Slotine, 2009) proposed a contraction based control algorithm for synchronizing multi robotic arms with an external agent. However, the kinematic feasibility of the intercept point has not been addressed. Extensive surveys on coordination strategies for multi robotic arm systems are presented in (Wimböck et al., 2012; Wimböck and Ott, 2012a; Smith et al., 2012; Caccavale and Uchiyama, 2016).

In de-centralized control architectures, the robots are controlled separately by their own local controllers (Liu and Arimoto, 1998; Sun and Mills, 2002). In early approaches, the coordination between a dual-arm system is achieved by categorizing them into two categories; namely a master and a slave. The motion of the master robot is assumed known whereas the slave robot must follow the master's motion while satisfying the closed-chain geometrical constraints (Luh and Zheng, 1987). Similarly, (Gams et al., 2015) proposed a control architecture to perform a task of lifting an unknown object with a dual-arm system, where the slave arm is synchronized with the master arm through a coupling guided by position and velocity feedback errors. Although computationally efficient, this strategy assumes a fixed master-slave relationship, which, when dealing with moving objects, may adversely affect performance if the arms need to switch responsibility to perform the task on-line. In (Bai and Wen, 2010), by using a velocity feedback and force feed-forward strategy, a de-centralized controller is proposed for transporting a flexible payload at a constants speed with multiple arms. In this approach master/slave roles are not assigned. This approach is limited when trying to reach for a moving object with an unpredictable motion.

Centralized controller strategies can address some of the issues that arise from de-centralized control (Aghili, 2013; Suda et al., 2003; Wang et al., 2015). These strategies consider the robots and the manipulated object as a closed kinematic chain. In this line, (Wimböck and Ott, 2012b) proposed an impedance control architecture for dual-arm manipulation, where the two end-effectors and a virtual frame, which is a function of the end-effectors' poses, are coupled via spatial springs. (Zhu, 2005) proposed a motion synchronization controller to coordinate the end-effectors of two robots when they are rigidly or flexibly holding an object without payload. (Likar et al., 2013) proposed a velocity level motion synchronization algorithm for controlling a cooperative dual-arm system.

They introduced an augmented kinematic chain which is a representation of two arms and the object. The corresponding Jacobian is calculated to control the augmented kinematic chain by solving the inverse kinematics problem at the velocity level.

By exploiting advantages of centralized and de-centralized impedance control strategies, (Caccavale et al., 2008) proposed a control architecture to achieve a desired impedance at both the object and the end-effector levels. Similarly, (Chiacchio and Chiaverini, 1998) proposed a two-level control architecture. Initially, the desired task variables are transformed into the corresponding joint-space motions by solving a centralized inverse kinematic problem. Then, the desired joint motions are fed to a decentralized joint-space controller. The main advantage of the decentralized control architectures is their computation cost. This is, in particular, important in a scenario where a lot of robots are engaged. In this case, as the computations are done locally, there is no need for a centralized large computer cluster.

All previously mentioned works assume that the object is firmly attached to the robots and modeled via a virtual object frame or by closing the kinematic chain. In chapter 4.4, we exploit the idea of the *virtual object* to coordinate the motion of the robots with each other and further with the real object. The motion of each arm and the virtual object are generated by using the proposed dynamical systems. The control architecture can generate two types of behaviors: (i) multi-arm asynchronous task-space behaviors, where each robot has its own target and (ii) multi-arm synchronous task-space behaviors, where the robots' task is to simultaneously reach-for a moving object and smooth transition between these two behavior. By using Lyapunov theorem, we show that by using the proposed centralized dynamical system, the goals of both behaviors can be achieved. Then, we report the successful implementation of the proposed architecture on two different robotic platforms.

2.3.1 Multi-Arm Self-Collision Avoidance

Self-collision avoidance is one of the main challenges in multi-arm manipulations. In general, the approaches for solving collision avoidance for manipulation or locomotion in humanoids can be categorized into two types: (i) *planning* methods which generate feasible collision-free trajectories in a known environment (Gharbi et al., 2009; Vahrenkamp et al., 2012; Escande et al., 2007; Vahrenkamp et al., 2010; Escande et al., 2014; Chrètien et al., 2016) and (ii) *reactive* approaches which solve collision-avoidance through the IK problem online (Ge and Cui, 2000; Santis et al., 2007; Sugiura et al., 2007; Fang et al., 2015; Wrede et al., 2013; Steil et al., 2014).

As reviewed in Section 2.1, planning approaches are, in general, computationally expensive. In the reaching a moving object scenario, the robots must

be able to react quickly to external perturbations. Hence, solving for collision avoidance must take less than $1 - 2ms$ which is far less than the typical computation times for the state of the art planning methods; e.g., $\approx 700ms \rightarrow \approx 1s$ (Kanehiro et al., 2012; Orthey and Stasse, 2013) or $2715ms$ on a single thread CPU[7] (Chrètien et al., 2016). It is worth to note that by designing a field-programmable gate array (FPGA), (Murray et al., 2016) could achieve dynamic adaptation at $< 1ms$. However, this approach requires a prior knowledge of the robot/environment configurations.

On the other hand, reactive approaches are computationally efficient. In (Santis et al., 2007; Sugiura et al., 2007), by determining the minimum distance between the robots' segments, repulsion forces are computed and used to generate self-collision avoidance motions. (Fang et al., 2015) proposed a hierarchical-based algorithm to determined the segments in-danger and solve the IK problem such that the distance between these segments increases. However, as mentioned in Section 2.1, the main shortcoming of these approaches is their sensitivity to the local minimums. Moreover, potential-field approaches suffer from providing no passages between closely spaced obstacles (Ge and Cui, 2000).

All the mentioned methods depend on computing minimum distances between the robotic arms to detect/avoid collisions. However, as shown in (Escande et al., 2014), the minimum distance between robotic arms can introduce non-linear and non-convex constraints to the optimization problem (Ratliff et al., 2015). This, in fact, is the main reason that the planning algorithms are computationally expensive and the reactive methods might not compute the global optimum. The approaches based on signed distance fields can address these shortcomings where the distances to the obstacles are encoded as continuous costs in local trajectory optimization frameworks; by either providing explicit cost gradients (Ratliff et al., 2009; Zucker et al., 2013) or through derivative-free stochastic optimization methods (Kalakrishnan et al., 2011). However, this approach is sensitive to the shape of the segments and fails to successfully compute the feasible motion when there are many local minimums. On the same track, by representing the robots' workspace via Riemannian metrics and their gradient, (Ratliff et al., 2015) provided a motion optimization framework which can be solved in $\approx 500ms$.

In chapter 4.5, we focus on providing coordination/compliance at the joint level by solving the self-collision avoidance problem efficiently (<2 ms). To this end, a centralized inverse kinematic solver, which is formulated as a convex Quadratic Programming (QP) problem with respect to the linear inequity and equality constraints, is proposed. We show that the proposed IK solver can, not only, be solved in less than $2ms$, but also, it is passive; i.e., the entire control architecture is accordingly passive and stable.

[7] However, the computation time can be improved significantly to $54ms$ by implementing it on GPU.

2.4 Non-contact/Contact Motion Control

As establishing a stable contact with an environment is the first step toward accomplishing interactive tasks, in the robotic/control literatures, different control architectures have been proposed for handling a physical contact. One approach is to regulate the contact force by decomposing the force and motion controllers; i.e., Hybrid Force/position controller (Raibert and Craig, 1981; Khatib, 1987; Mistry and Righetti, 2012; Lin et al., 2017; Liu and Arimoto, 1998; Whitcomb et al., 1997; Jinno et al., 1995). In this approach, the position and force controllers simultaneously govern the robot along unconstrained and constrained directions, respectively. While the robot is in the *free motion* phase, the position controller leads the robot in all the directions. Once, the robot touches a surface, the force controller takes the lead on the constrained directions. Although a precise contact force can be achieved, stabilizing the forces at contact is not possible as the impact lasts less than a millisecond, leaving no time for the robot to react to the impact force. Moreover, if the contact is unstable, where the robot bounces on the surface, a large impact force and switching between position/forces controllers leads to chattering.

Indirect force control architectures address the problem of switching between controllers by ensuring the desired contact force through a compliant behavior of the end-effector (Bonitz and Hsia, 1996; Hogan, 1985; Seraji and Colbaugh, 1997; Roveda et al., 2016; Leidner et al., 2016; Ficuciello et al., 2015). Compliance can emerge from two different facets; i.e., passive or active. While the former is restricted to a special set of actuators or surface materials, the latter is, instead, achieved through the motion control. Although, these architectures have implicitly addressed the issue of stability at impact by eliminating the need of switching, there is no guarantee that the robot remains in the contact after the impact.

The difficulties linked to achieving a stable contact has attracted attention in the last two decades. Early approaches addressed the stable contact problem with position/force hybrid control architectures. (Mills and Lokhorst, 1993; Mills, 1990; Tarn et al., 1996; Heck et al., 2015) proposed a hybrid control architecture in which a stable contact can be ultimately established after a finite number of iterations (bouncing). On the same track, (Pagilla and Yu, 2000, 2001a; Tomizuka, 1997) proposed three control laws for the three motion phases. Once the first impact is occurred, the controller in the *transition* phase is activated which, asymptotically, reduces the normal velocity to zero. In (Youcef-Toumi and Gutz, 1989), an integral force compensation with a velocity feedback controller is proposed for force tracking and rejecting the effect of impacts, where the force regulation is activated as soon as the force sensor detects the impact. (Roveda et al., 2016) proposed a two-layer controller which consists of an impedance and an admittance controllers. The parameters

of the latter are calculated by solving a Linear Quadratic Regulator problem to minimize the force overshooting. In (Lee et al., 2003; Jin et al., 2005), a hybrid impedance(admittance)/time-delayed controller is proposed to absorb the impact force where the control input becomes zero if the contact force is not sensed. By artificially saturating the feedback sensors and modeling the contact surface via a passive mass-spring system, a controller for a 2-DOFs planar robotic arm is proposed to limit the impact force in (Liang et al., 2007). (Tornambe, 1999) shows that the classical PD feedback control law can be effectively used for mechanical systems subject to inequality constraints. By assuming the contact surface is a passive mass-spring system, (Dupree et al., 2008) developed an adaptive control architecture to pushes the system to a desired set while the dynamics of neither the robot nor the environment is precisely known. The proposed controller in (Brogliato et al., 1997) guarantees stabilization of the manipulator on the contact surface after a finite times of bouncing on it. Even though in the mentioned works, it can be proved that the robot's motion and the contact is asymptotically/ultimately stable, there is no guarantee that the robot remains in the contact after the first impact.

By approximating the contact surface with a passive spring system and dividing the state space into five regions, (Martino and Broucke, 2014) uses the feedback force to propose piecewise affine controllers for each region such that a stable impact is achieved for linear one dimensional systems. However, in (Martino and Broucke, 2014) the stable impact is achieved if the environment and the tool can be precisely modeled via a spring system and the bandwidth of the position and force sensors and the communication delays are infinite and zero, respectively.

In Chapter 5, we leverage the properties of Dynamical Systems (DS) for immediate re-planning and their inherent robustness to real-time perturbations and propose an active compliant control strategy for stably contacting a surface with a robotic arm. In Chapter 3 and Chapter 4, we proposed dynamical systems to intercept a moving object with zero relative velocity by single or multi robotic arms, respectively. Those proposed DSs are particularly tailored for the reaching and compliantly intercepting moving objects. In Chapter 5, as the *transition* is a local behavior, we propose a strategy consisting of locally modulating the motion of the robot once it is close to the surface such that its velocity aligns with the contact surface so that the stable contact can be established. The proposed architecture can be integrated into existing DS-based motion control approaches, where they represent the nominal arm behavior.

We show theoretically and empirically that by using the proposed modulation framework, the impact happens once and the robot remains in contact after the impact.

Chapter 3

Softly Intercepting a moving object with a robotic arm

"It is not enough to be in the right place at the right time. You should also have an open mind at the right time."

Paul Erdos
1913 – 1996

3.1 Introduction

In this chapter, we tackle the problem of coordinating a robotic arm with a moving object such that it can compliantly intercept the object at a desired position. As stated before, to highlight the challenges of the problem, we are, in particular, interested in catching flying objects as the flight and more importantly the impact durations are extremely short, leaving less than a few milliseconds for the hands to close on the object to such that it is tightly secured in the grip.

By coordinating the motion of the arm with that of the object, we introduce a soft (compliant) catching strategy which is less sensitive to timely control of the interception. The *soft* catching strategy consists of having the robot move with the object for a short period of time (Figure 3.1). This leaves not only more time for the fingers to close on the object but also reduces the chance of failure due to imprecise control of the time and position at which the hand intercepts the object. The *soft* catching strategy we propose here assumes that the arm-hand-object system has no mechanical compliance or that the inherent mechanical compliance of the system is negligible in comparison to the strength of the impact forces. Compliance is then provided through active control of the arm motion.

To successfully catch an object *softly*, the arm must intercept the object *on time*, *at the right place*, and with a *specific velocity*. Planning a trajectory that satisfies the above three constraints in time, position and velocity can be done using standard optimal control approaches. This is however time consuming and cannot be performed within the few milliseconds at our disposal. We leverage the properties of autonomous dynamical systems for immediate re-planning of

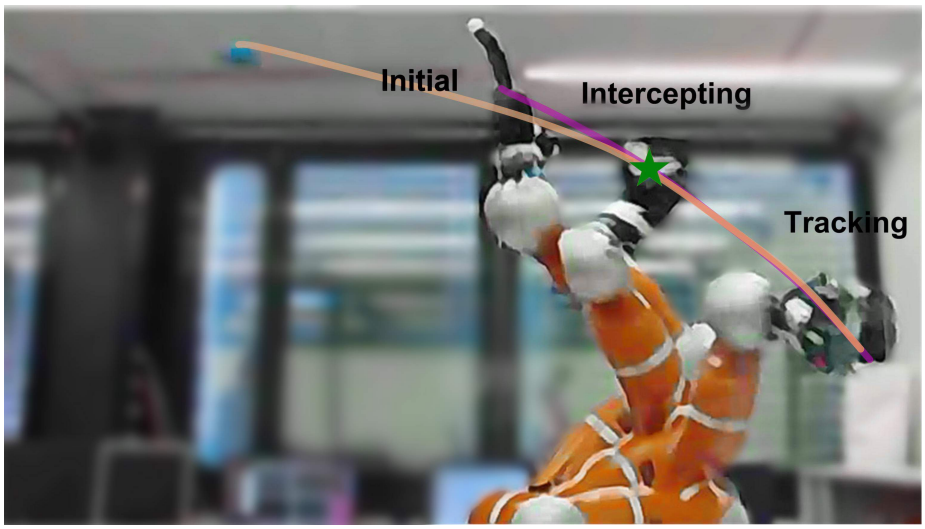


Figure 3.1: Schematic of the soft catching strategy. The arm starts from an *initial* point and moves to the *intercept* point, where it meets the object. It then continues its motion aligning its trajectory with that of the object and slowly reducing its velocity, while the fingers close on the object.

motion and devise a control law which provides the robot with a kinematically feasible motion which intercepts the object at the desired position and with a velocity aligned with that of the object.

This chapter starts by introducing our control law formulated as a (Linear Parameter Varying) LPV based dynamical system to generate soft catching motion in Section 3.3. There, we theoretically show how coupling the motion of the robot with that of the object results in a dynamical system which can intercept the object at the desired position with a desired velocity aligned with the object's velocity. Moreover, with the purpose of maximizing the softness at the interception subject to the kinematics constraints of the robot, a closed loop optimal control problem is suggested.

In Section 3.4, we introduce our approach for approximating the parameters of LPV systems by using Gaussian Mixture Models (GMM), which inherently results in the normalized scheduling parameters. Hand-tuning GMM's parameters is tedious and time consuming. Hence. we use the Learning from Demonstrations (LfD) framework as it has emerged as an tool to efficiently estimate the open parameters of controller laws by transferring skills to robots. Apart from the model, the LfD framework is base on two other technical components; i.e., (i) the optimization algorithm and (ii) the demonstrations (training data-set).

To address the former, in Section 3.4.1, we propose an optimization algorithm to estimate the parameters of the model from the demonstrations. Here, we compare our method with Stable Estimator of Dynamical systems (SEDS) (Khansari-Zadeh and Billard, 2011) and show that the performances and the computation cost of the proposed method is significantly better.

Since catching is an extremely rapid action, the training data-set should be

a representative of the fastest feasible motion of the robot. It would be difficult to have this provided by a human expert as kinesthetic teaching would not make it possible to move the arm at its maximal speed. Hence, in Section 3.4.2 we propose an algorithm which constructs the training data-set by solving a closed loop optimal control problem which maximizes the robot's velocity at each step.

The performance of the proposed method is systematically evaluated in a real-world experiment with KUKA LBR IIWA (7 degree of freedom arm robot) in Section 3.5 where, we choose three objects with different stiffness. The objects are almost impossible to catch with the hard catching approach (Kim et al., 2014) as they bounce out of the hand instantaneously, see accompanying video here: https://youtu.be/FxvVJzb61js.

Large portions of this chapter correspond to the following publication, where the stability and the convergence proofs were developed collaboratively with Mahdi Khoramshahi, currently a PhD student at LASA.

- Mirrazavi Salehian, S. S., Khoramshahi, M. and Billard, A. (2016) A Dynamical System Approach for Catching Softly a Flying Object: Theory and Experiment. in IEEE Transactions on Robotics, Vol. 32, No. 2, pp. 462-471, April 2016.

3.2 The Control Framework

In order to achieve stable and soft interception of a moving object with a robotic arm, two main problems need to be solved simultaneously: (i) computing the feasible intercept point and (ii) planning coordinated motion of the arm such that it intercepts the object at the desired position and with desired velocity. An overview of the proposed framework is illustrated in Figure 3.2.

As seen on the illustration, sub-component (A) predicts the future object's trajectory and then determines the feasible intercept point from this prediction. It uses an estimate of the reachable workspace of the robotic arm (learned off-line prior to experiment) and an on-line step in which it continuously measures the object's position from a visual tracking system. Sub-component (B) uses the intercept points predicted from (A) and the current robot's state to generate the desired end-effector position. The desired end-effector motion is converted into the desired joint motion in sub-component (C) and then it is sent to the robotic arm. A detailed description of each of these sub-components are presented in Section 3.2.1, Section 3.3 and Section 3.5, respectively.

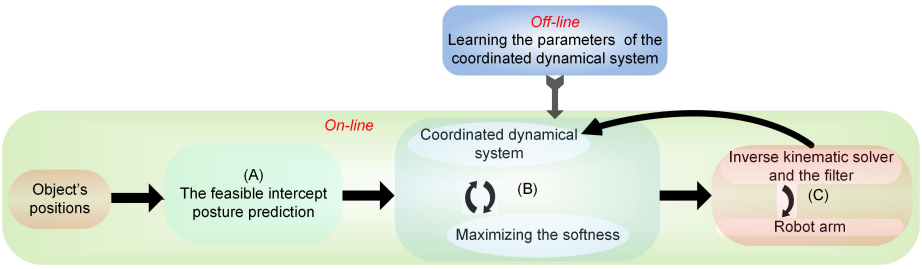


Figure 3.2: Block diagram for robotic intercepting an object in flight.

3.2.1 Object Trajectory and Intercept Point Prediction

The feasible intercept point ($\xi^O(T^*)$) is estimated by a simplified version of catching point prediction algorithm proposed in (Kim et al., 2014). Given the predicted trajectory of the object, this algorithm assesses the kinematics feasibility of the predicted trajectory to reach. The reachable workspace of each robot is modeled via a probabilistic classification model $p(\xi^O; \theta^W) \quad \forall j \in \{1, \ldots, K_r\}$, namely a Gaussian Mixture Model as follows:

$$p(\xi^O; \theta^W) = \sum_{l=1}^{K^w} \pi_l \mathcal{N}(\xi^O | \mu_l, \Sigma_l) \tag{3.2.1}$$

where π_l, μ_l, Σ_l correspond to the prior, mean and covariance matrix of the $l = \{1 \ldots K^w\}$ Gaussian functions, respectively, estimated by using the Expectation-Maximization algorithm (Bishop, 2007). In order to generate the training dataset, all possible postures of the robot are simulated by systematically testing all possible displacements of its joints. δ is the *minimum likelihood* threshold and it is determined such that the likelihood of 99% of the training points is higher than the threshold δ. If $p(\xi^O; \theta^W)$ exceeds it, ξ^O is classified as a feasible configuration.

3.3 The Compliant Dynamical System for Softly Intercepting Moving Object

Dynamical systems are popular and powerful methods for autonomously generating stable and robust motions according to training data-points (Khansari-Zadeh and Billard, 2011). DSs are applicable for various robotic manipulations which need an accurate and fast re-computing of motions. Formulating DSs as LPV systems allows modeling a wide class of non-linear systems and the use of many tools from the linear systems theory for analysis and control (Emedi

and Karimi, 2015). LPV systems can be thought of as a weighted combination of linear models, each valid at a specific operating point. We consider a class of continuous-time LPV systems given by the following model:

$$\begin{aligned}\ddot{\xi}(t) &= \mathbf{A}_1(\theta(t))\xi(t) + \mathbf{A}_2(\theta(t))\dot{\xi}(t) + u(t)\\ y(t) &= C_g \begin{bmatrix}\xi(t) & \dot{\xi}(t)\end{bmatrix}^T\end{aligned} \tag{3.3.1}$$

Where $\xi(t) \in \mathbb{R}^D$ is the state of the dynamical system. $u(t)$ is the control input vector. $y(t)$ is the plant output vector. $\theta \in \mathbb{R}^{K\times 1}$ are the vector of scheduling parameters[1];

$$\theta = \begin{bmatrix}\theta^1 & \dots & \theta^k\end{bmatrix}^T \tag{3.3.2}$$

. $\mathbf{A}^i(.) : \mathbb{R}^K \to \mathbb{R}^{D\times D}\ \forall i \in \{1, 2\}$ are the affine dependences of the state-space matrices on the scheduling parameter and the state vectors:

$$\begin{aligned}\mathbf{A}_1(\theta) &= \sum_{k=1}^{K} \theta^k A_1^k \quad A_1^k \in \mathbb{R}^{D\times D}\\ \mathbf{A}_2(\theta) &= \sum_{k=1}^{K} \theta^k A_2^k \quad A_2^k \in \mathbb{R}^{D\times D}\end{aligned} \qquad \theta^k \in \mathbb{R}^{1\times 1} \tag{3.3.3}$$

As we discussed in Section 3.1, soft catching can be achieved as a combination of *tracking* and *reaching* motions. To achieve this, the following control input vector $u(t)$ is proposed.

$$\begin{aligned}u(t) =& \gamma(t)\ddot{\xi}^O - \mathbf{A}_1(\theta)\gamma(t)\xi^O - \mathbf{A}_2(\theta)(\gamma(t)\dot{\xi}^O + \dot{\gamma}(t)\xi^O)\\ &+ 2\dot{\gamma}(t)\dot{\xi}^O + \ddot{\gamma}(t)\xi^O\end{aligned} \tag{3.3.4}$$

Where the state of the object is denoted by ξ^O. $0 < \gamma(t) < 1$ is called softness and it is a continuous and continuously differentiable parameter. The origin is located on the desired intercept point and must be reached at time T^*; i.e., $\xi^O(T^*) = \begin{bmatrix}0 & \dots & 0\end{bmatrix}^T$. To ensure smooth tracking of the object's motion, the combination of the position, velocity and acceleration of the object with the affine dependences is chosen as the control law. By substituting (3.3.4) and

[1] The scheduling parameters can be a function of time (t), states of the system ($\xi(t)$) or external signals $d(t)$, i.e., $\theta(t, \xi(t), d(t))$. In the rest of this chapter, the arguments of θ are dropped for simplicity.

(3.3.3) into (3.3.1), we have:

$$\begin{aligned}\ddot{\xi}(t) &= \gamma(t)\ddot{\xi}^O(t) + 2\dot{\gamma}(t)\dot{\xi}^O(t) + \ddot{\gamma}(t)\xi^O(t) \\ &\quad + \mathbf{A}_1(\theta)(\xi(t) - \gamma(t)\xi^O(t)) + \mathbf{A}_2(\theta)(\dot{\xi}(t) - (\gamma(t)\dot{\xi}^O(t) + \dot{\gamma}(t)\xi^O(t))) \\ y(t) &= C_g \begin{bmatrix}\xi(t) & \dot{\xi}(t)\end{bmatrix}^T\end{aligned} \tag{3.3.5}$$

Theorem 1. *The dynamical system given by* (3.3.5) *asymptotically converges to* $\begin{bmatrix}\gamma(t)\xi^O & \gamma(t)\dot{\xi}^O + \dot{\gamma}(t)\xi^O\end{bmatrix}^T$ *; i.e.*

$$\lim_{t\to\infty} \|\xi(t) - \gamma(t)\xi^O(t)\| = 0 \tag{3.3.6}$$

$$\lim_{t\to\infty} \|\dot{\xi}(t) - (\gamma(t)\dot{\xi}^O(t) + \dot{\gamma}(t)\xi^O(t))\| = 0 \tag{3.3.7}$$

if (3.3.5) *meets the following constraints:*

$$\begin{cases} \begin{bmatrix}0 & I\\ A_2^k & A_1^k\end{bmatrix}^T P + P\begin{bmatrix}0 & I\\ A_2^k & A_1^k\end{bmatrix} \prec 0 & \\ 0 \prec P, \qquad P^T = P & \forall k \in \{1,\dots,K\} \\ 0 \le \theta^k \le 1, \qquad \sum_{k=1}^{K}\theta^k = 1 & \end{cases} \tag{3.3.8}$$

where $\prec 0$ refers to negative definiteness of a matrix.

Proof: see Appendix A.1.

If we assume that $\gamma(t)$ is constant, (3.3.5) is a combination of two motions; i.e., *reaching* and *tracking*. In this equation, setting $\gamma = 0$ yields a *reaching* dynamical system; i.e., $\lim_{t\to\infty}\begin{bmatrix}\xi(t) & \dot{\xi}(t)\end{bmatrix} = \begin{bmatrix}\xi^O(T^*) & 0_{1\times D}\end{bmatrix}^T$. Hence, the *position* constraint is satisfied. However, the *time* and *velocity* constraints are not satisfied. Setting $\gamma = 1$ results in a tracking motion with an error decreasing asymptotically according to:

$$\ddot{\xi}(t) - \ddot{\xi}^O(t) = \mathbf{A}_1(\theta)(\xi(t) - \xi^O(t)) + \mathbf{A}_2(\theta)(\dot{\xi}(t) - \dot{\xi}^O(t)) \tag{3.3.9}$$

By varying the value of the gamma parameter, one can ensure that the robot reaches the object not only with the right velocity, but also at the right time and location. This is summarized in the following Proposition:

Proposition 1. *The dynamical system given by* (3.3.5) *reaches the desired in-*

Table 3.1: Pseudo-code for the optimal control formulation for maximizing the softness of catching.

Do for each step i

$$\ddot{\gamma}[i+1] = \underset{\ddot{\gamma}}{\text{argmax}}(\gamma[i+1])$$

subject to:

$0 < \gamma[i+1] < 1$ (Alg-1-1)

$-\Upsilon \leq \ddot{\gamma}[i+1] \leq \Upsilon$ (Alg-1-2)

$\dot{\gamma}[i+1] = \dot{\gamma}[i] + \ddot{\gamma}[i+1]\Delta t$ (Alg-1-3)

$\gamma[i+1] = \gamma[i] + \dot{\gamma}[i+1]\Delta t$ (Alg-1-4)

$\ddot{\xi}[i+1]$ is calculated from (3.3.5) (Alg-1-5)

$-J(q[i])\dot{q}_{max} \leq \dot{\xi}[i+1] \leq J(q[i])\dot{q}_{max}$ (Alg-1-6)

$-\dot{J}(q[i])\dot{q}_{max} - J(q[i])\ddot{q}_{max} \leq \ddot{\xi}[i+1] \leq \dot{J}(q[i])\dot{q}_{max} + J(q[i])\ddot{q}_{max}$ (Alg-1-7)

$\delta \leq p(\xi[i+1]; \theta^W)$ (Alg-1-8)

tercept point ($\xi^O(T^*)$) *asymptotically with a velocity aligned with that of the object,* $\dot{\xi}(T^*) \approx \gamma\dot{\xi}^O(T^*)$.[2]

Proof: The intercept point is located on the object's trajectory and, as mentioned before, it is the origin. Hence, $\gamma\xi^O(T^*) = \begin{bmatrix}0 & \dots & 0\end{bmatrix}^T$ and $\gamma\xi^O$ crosses ξ^O at the desired intercept point. Since the arm is ensured to reach asymptotically $\gamma\xi^O$, it is ensured to intercept the object at the desired intercept point (Theorem 1). Moreover, the arm's velocity will be a velocity vector proportional to that of the object, i.e., $\dot{\xi}(T^*) \approx \gamma\dot{\xi}^O(T^*)$. ■

On one hand, by increasing the value of γ, the end-effector's velocity at the intercept point gets closer to the object's velocity; i.e., more softness in catching. On the other hand, the generated trajectory may get kinematically infeasible for the robot to track. In order to generate a kinematically feasible trajectory with maximum softness in catching, we propose a closed loop optimal control which is formulated in Table 3.1. The goal is to maximize the softness ($\gamma(t)$) while ensuring that the trajectory is kinematically feasible for a robot to track. This introduces a constrained optimization problem at each step which can be formulated as a non-linear Programming (NLP) problem.

In Table 3.1, $\leq$ corresponds to the component-wise inequality. Υ is a large positive number. Δt is the time step. $q[i]$ is the joint configuration which is corresponding to the end-effector position and orientation $\xi[i]$; i.e., $\xi[i] = \mathbb{F}(q[i])$ where $\mathbb{F}$ is the robot forward kinematic function. (Alg-1-1) satisfies the velocity constraint in a soft catching motion. (Alg-1-6) and (Alg-1-7) guarantee that the velocity and acceleration of the generated motion are kinematically feasible for the robot. The feasibility of the generated motion at the position level is guaranteed by (Alg-1-8), where the workspace of the robot is modeled through a probabilistic representation of the feasible postures (3.2.1).

[2] We assume that the dynamical system (3.3.5) is fast enough to converge to the acceptable neighbourhood of the desired trajectory $\gamma\begin{bmatrix}\xi^O & \dot{\xi}^O\end{bmatrix}^T$ before the catching time; i.e., $\|\xi(T^*) - \gamma\xi^O(T^*)\| \leq \varepsilon$ and $\|\dot{\xi}(T^*) - \gamma\dot{\xi}^O(T^*) - \dot{\gamma}\xi^O(T^*)\| \leq \varepsilon$, where ε is a small positive number.

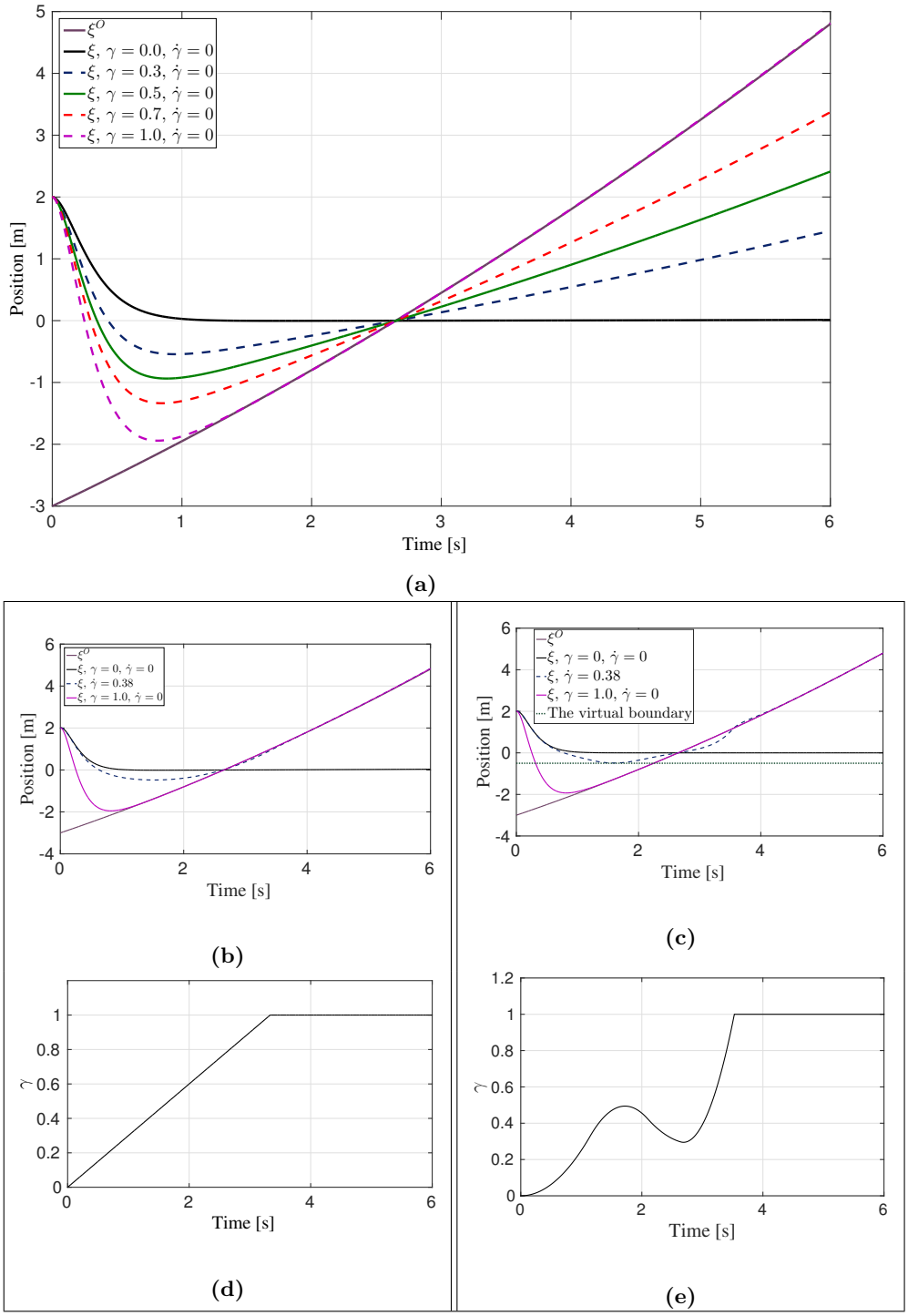


Figure 3.3: The behavior of 1-D forced dynamical system subject to the value of γ and $\dot{\gamma}$. $\dot{\gamma} = 0$ in (a). (b) and (c) show the behavior of the 1-D system when $\dot{\gamma}$ is constant and time-varying, respectively. (d) and (e) show the corresponding value of γ in (b) and (c), respectively. In (c), the virtual workspace constraint is satisfied as Table 3.1 is used to optimize the value of γ.

Example: Consider (3.3.5) as the following 1-D dynamical system with one scheduling parameter.

$$\begin{cases} A_2 = -12 \\ A_1 = -36 \end{cases} \Rightarrow \ddot{\xi} = -12\dot{\xi} - 36\xi + u(t) \tag{3.3.10}$$

This is a critically damped system which asymptotically converges to zero from an arbitrary initial condition ($\xi(0) = 2,\ \dot{\xi}(0) = 0$). Let's assume the following dynamic model for the object.

$$\ddot{\xi}^O(0) = 0.1 \qquad \dot{\xi}^O(0) = 1 \qquad \xi^O(0) = -3 \tag{3.3.11}$$

The solutions of (3.3.5) for this example are illustrated in Figure 3.3 for different values of γ and $\dot{\gamma}$. It shows that for all the values of γ, $\dot{\gamma}$ and $\ddot{\gamma}$, the generated trajectories join the object's trajectory on time at the origin. Moreover, by increasing the value of γ, the end-effector's velocity is getting closer to the object velocity: compare the black line with the purple line in Figure 3.3a. Figure 3.3c shows an example of use of Table 3.1 to generate a feasible trajectory where there is a geometrical constraints on the motion of the dynamical system.

In order to use the LPV dynamical system (3.3.1) to successfully catch a flying object *softly*, the scheduling parameters must be accurately and precisely modeled. Next, we propose a GMR based second order dynamical system and reformulate it to model the unforced[3] LPV dynamical system (3.3.1).

3.4 Approximating The Parameters of LPV-Based Dynamical Systems

One typical choice is to approximate LPV systems by using linear regression models. The most popular approaches to perform this approximation use polynomial or periodic functions (Bamieh and Giarre, 2002). In this section, we introduce a probabilistic approach for approximating the parameters of the LPV based dynamical systems from a training data-set using Gaussian Mixture Regression (GMR). Approximating LPV systems via a GMR inherently results in a normalized scheduling parameters; i.e., $0 < \theta^k \leq 1,\ \sum_{k=1}^{K} \theta^k = 1$. Having normalized parameters is advantageous for two reasons. First, it bounds the search space. Second, the parameters can be approximated independent of the stability constraints.

For the sake of completeness, in the following, we present our approach for

[3] In this thesis, we call the LPV dynamical system (3.3.1) "unforced" if $u(t) = 0$.

approximating both first and second-order dynamical systems. To estimate an unforced first-order LPV dynamical system with Gaussian Mixture Model, the mean (μ^k) and the covariance matrix (Σ^k) of a Gaussian k$^{\text{th}}$ are defined by:

$$\mu^k = \begin{bmatrix} \mu^k_{\xi} \\ \mu^k_{\dot{\xi}} \end{bmatrix} \qquad \Sigma^k = \begin{bmatrix} \Sigma^k_{\xi} & \Sigma^k_{\xi\dot{\xi}} \\ \Sigma^k_{\dot{\xi}\xi} & \Sigma^k_{\dot{\xi}} \end{bmatrix} \tag{3.4.1}$$

A representation of an unforced first order LPV system with GMM is formulated as:

$$\dot{\xi} = \sum_{k=1}^{K} h^k(\xi) A^k \xi + b^k \tag{3.4.2}$$

where

$$\begin{cases} A^k = \Sigma^k_{\dot{\xi}\xi} (\Sigma^k_{\xi})^{-1} \\ b^k = \mu^k_{\dot{\xi}} - A^k \mu^k_{\xi} \\ h^k(\xi) = \dfrac{P^k(k) P^k(\xi|k)}{\sum\limits_{i=1}^{K} P^k(i) P^i(\xi|i)} \end{cases} \tag{3.4.3}$$

In this formulation, $P^k(k) = \pi^k$ is the prior probability of each Gaussian component and $P(\xi|k) = \mathcal{N}(\xi|\mu^k, \Sigma^k)$ denotes the conditional probability distribution function (pdf) corresponding to the k$^{\text{th}}$ Gaussian function. Hence, $0 < h^k(\xi) \leq 1$ is a continuous and continuously differentiable function.

To estimate an unforced second-order LPV dynamical system with Gaussian mixture model, the parameters of the DS become the priors (π^k), the means (μ^k) and the covariance matrices (Σ^k) of the $k \in \{1, ..., K\}$ Gaussian function. The mean and the covariance matrix of a Gaussian k$^{\text{th}}$ are defined by:

$$\mu^k = \begin{bmatrix} \mu^k_{\xi} \\ \mu^k_{\dot{\xi}} \\ \mu^k_{\dot{\xi}} \\ \mu^k_{\ddot{\xi}} \end{bmatrix} \qquad \Sigma^k = \begin{bmatrix} \Sigma^k_{\xi} & \Sigma^k_{\xi\dot{\xi}} & \Sigma^k_{\xi\dot{\xi}} & \Sigma^k_{\xi\ddot{\xi}} \\ \Sigma^k_{\dot{\xi}\xi} & \Sigma^k_{\dot{\xi}} & \Sigma^k_{\dot{\xi}} & \Sigma^k_{\dot{\xi}\ddot{\xi}} \\ \Sigma^k_{\dot{\xi}\xi} & \Sigma^k_{\dot{\xi}} & \Sigma^k_{\dot{\xi}} & \Sigma^k_{\dot{\xi}\ddot{\xi}} \\ \Sigma^k_{\ddot{\xi}\xi} & \Sigma^k_{\ddot{\xi}\dot{\xi}} & \Sigma^k_{\ddot{\xi}\dot{\xi}} & \Sigma^k_{\ddot{\xi}} \end{bmatrix} \tag{3.4.4}$$

A representation of an unforced second order LPV system (3.3.1) with GMM is formulated as:

$$\begin{bmatrix} \dot{\xi} \\ \ddot{\xi} \end{bmatrix} = \sum_{k=1}^{K} h^k(\xi, \dot{\xi}) \left(\begin{bmatrix} 0 & I \\ A^k_1 & A^k_2 \end{bmatrix} \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix} + \begin{bmatrix} 0 \\ b^k \end{bmatrix} \right) \tag{3.4.5}$$

where

$$\begin{cases} \begin{bmatrix} 0 & I \\ A_1^k & A_2^k \end{bmatrix} = \begin{bmatrix} \Sigma_{\dot{\xi}\xi}^k & \Sigma_{\dot{\xi}}^k \\ \Sigma_{\ddot{\xi}\xi}^k & \Sigma_{\ddot{\xi}\dot{\xi}}^k \end{bmatrix} \begin{bmatrix} \Sigma_{\xi}^k & \Sigma_{\xi\dot{\xi}}^k \\ \Sigma_{\dot{\xi}\xi}^k & \Sigma_{\dot{\xi}}^k \end{bmatrix}^{-1} \\ b^k = \mu_{\ddot{\xi}}^k - A_2^k \mu_{\dot{\xi}}^k - A_1^k \mu_{\xi}^k \\ h^k(\xi, \dot{\xi}) = \dfrac{P^k(k) P^k([\xi \ \dot{\xi}]^T | k)}{\sum_{i=1}^{K} P^i(i) P^i([\xi \ \dot{\xi}]^T | i)} \end{cases} \tag{3.4.6}$$

In this formulation, $P^k(k) = \pi^k$ is the prior probability of each Gaussian component and $P^k([\xi \ \dot{\xi}]^T|k) = \mathcal{N}([\xi \ \dot{\xi}]^T | [\mu_{\xi}^k \ \mu_{\dot{\xi}}^k]^T, \begin{bmatrix} \Sigma_{\xi}^k & \Sigma_{\xi\dot{\xi}}^k \\ \Sigma_{\dot{\xi}\xi}^k & \Sigma_{\dot{\xi}}^k \end{bmatrix})$ denotes the conditional probability distribution function (pdf) corresponding to the k$^{\text{th}}$ Gaussian function. By expanding (3.4.5), (3.4.5) is equivalent to the unforced LPV dynamical system (3.3.1) if $b^k = 0 \ \forall k \in \{1, \dots, K\}$; i.e.

$$\ddot{\xi} = \sum_{k=1}^{K} h^k(\xi(t), \dot{\xi}(t)) A_1^k \xi(t) + \sum_{k=1}^{K} h^k(\xi(t), \dot{\xi}(t)) A_2^k \dot{\xi}(t) \tag{3.4.7}$$

3.4.1 Learning First and Second Order Asymptotically Stable Models

Consider a data set of training data-points $\{\xi_m, \dot{\xi}_m, \ddot{\xi}_m\}_{m=1}^M$ that denotes examples of kinematically and dynamically feasible trajectories of the end-effector where M is the number of data-points. Based on this data-set, one can learn either a first order or a second order dynamical system by approximating the mapping from $\{\xi_m\}$ to $\{\dot{\xi}_m\}$ or from $\{\xi_m, \dot{\xi}_m\}$ to $\{\ddot{\xi}_m\}$, respectively. In order to estimate the parameters of a stable first order dynamical system given in the form by (3.4.2), we propose the following learning algorithms which uses Mean Square Error as a means to quantify the accuracy of the estimation:

$$\min_{\theta} C(\theta) = \sum_{m=1}^{M} \|\dot{\xi}_m - \dot{\xi}\|^2 \tag{3.4.8}$$

subject to :

$$\begin{cases} {A^k}^T P + P A^k \prec 0 \\ 0 \prec P, \quad P^T = P \\ b^k = 0, \end{cases} \quad \begin{cases} 0 \prec \Sigma^k \\ 0 < \pi^k \leq 1 \\ \sum_{k=1}^{K} \pi^k = 1 \end{cases} \tag{3.4.9}$$

for $\forall k \in \{1, \dots, K\}$. $C(\Theta)$ is the cost function and Θ is the GMM parameters. $\dot{\xi}$ is computed directly from (3.4.2).

Similarly, to estimate the parameters of a stable second order DS given in the form by (3.4.5), the following learning algorithm is proposed:

$$\min_{\theta} C(\theta) = \sum_{m=1}^{M} \|\ddot{\xi}_m - \ddot{\xi}\|^2 \tag{3.4.10}$$

subject to :

$$\begin{cases} \begin{bmatrix} 0 & I \\ A_1^k & A_2^k \end{bmatrix}^T P + P \underbrace{\begin{bmatrix} 0 & I \\ A_1^k & A_2^k \end{bmatrix}}_{A^k} \prec 0 \\ 0 \prec P, \qquad P^T = P \\ b^k = 0, \end{cases} \quad \begin{cases} 0 \prec \Sigma^k \\ 0 < \pi^k \leq 1 \\ \sum_{k=1}^{K} \pi^k = 1 \end{cases} \tag{3.4.11}$$

for $\forall k \in \{1, \dots, K\}$. $C(\Theta)$ is the cost function and Θ is the GMM parameters. M is the number of the training data-points. $\ddot{\xi}$ is computed directly from (3.4.5).

The group of constrains on the left-hand-side of (3.4.11) or (3.4.9) ensures asymptotic stability, while the group of constraints on the right-hand-side follows from the definition of the Gaussian Mixture Models.

Our algorithm, introduced in Table 3.2, solves the NLP problems (3.4.8) and (3.4.10) in two steps. In the first step, we use the Expectation-Maximization (EM) algorithm to only find the parameters of the scheduling functions; i.e., $\{\pi^k,\ \mu_\xi^k,\ \Sigma_\xi^k\}\ \forall k \in \{1, \dots, K\}$ for a first order and $\{\pi^k,\ \mu_\xi^k,\ \mu_{\dot{\xi}}^k,\ \Sigma_\xi^k,\ \Sigma_{\dot{\xi}}^k,\ \Sigma_{\dot{\xi}\xi}^k\}\ \forall k \in \{1, \dots, K\}$ for a second order DS. In the second step, $A^k,\ P,\ \forall k \in \{1, \dots, K\}$ are simultaneously estimated. Due to the product between A^k and P, the problem (3.4.9) or (3.4.11) is not convex. There are several approaches to solve this problem. The easiest way is to assume that P is a known matrix by guessing it in advance. As a result, (3.4.9) or (3.4.11) are converted to a convex optimization. Even though this approach is computationally fast, it is very conservative as the solution depends very much on this guess. The other approach simultaneously estimates both A^k and P. Despite the fact that this approach is non-convex, based on our experience, it usually results in a better local minimum than the first approach. Hence, all the simulations are done by using this approach.

Algorithm 3.2 is implemented in Matlab. The initialization of the EM is done by K-means algorithm. We use Yalmip (Löfberg, 2004) interface and Penlab (Fiala et al., 2013) solver to solve the optimization step of the algorithm. The source code of the learning algorithm is available on-line and can downloaded from

https://github.com/sinamr66/SESODS_lib.

Table 3.2: Learning GMM based LPV dynamical systems. For sake of clarity, we separately explain the learning algorithms for first and second order systems in 3.2a and 3.2b, respectively. The learning algorithm can be easily extended to higher orders. The source code of the learning algorithm is available on-line at here: https://github.com/sinamr66/SESODS_lib.

(a) Learning algorithm for LPV based first order DSs

Inputs: $K,\ \{\xi_m, \dot{\xi}_m\}_{m=1}^{M}$

Output: $\Theta = \{\pi^k, \mu^k, \Sigma^k\}\ \ \forall k = \{1, \dots, K\}$

1- Initialization step:

1-1- Run EM over $\{\xi_m\}_{m=1}^{M}$ to find an estimate of $\{\pi^k,\ \mu_\xi^k,\ \Sigma_\xi^k\}\ \ \forall k = \{1, \dots, K\}$.

1-2- Initialize the scheduling functions

$$h^k(\xi) = \frac{P^k(k)P^k(\xi|k)}{\sum_{i=1}^{K} P^i(i)P^i(\xi|i)}$$

2- Optimization step:

Solve:

$$\min_{A^k, P} C(\theta) = \sum_{m=1}^{M} \|\dot{\xi}_m - \dot{\xi}\|^2\ \forall k \in \{1, \dots, K\}$$

s.t.:

$$(A^k)^T P + P(A^k) \prec 0$$
$$0 \prec P, \qquad P^T = P$$

where :

$$\dot{\xi} = \sum_{k=1}^{K} h^k(\xi) A^k \xi$$

(b) Learning algorithm for LPV based second order DSs

Inputs: $K,\ \{\xi_m, \dot{\xi}_m, \ddot{\xi}_m\}_{m=1}^{M}$

Output: $\Theta = \{\pi^k, \mu^k, \Sigma^k\}\ \ \forall k = \{1, \dots, K\}$

1- Initialization step:

1-1- Run EM over $\{\xi_m, \dot{\xi}_m\}_{m=1}^{M}$ to find an estimate of $\{\pi^k,\ \mu_\xi^k,\ \mu_{\dot{\xi}}^k,\ \Sigma_\xi^k,\ \Sigma_{\dot{\xi}}^k,\ \Sigma_{\xi\dot{\xi}}^k\}\ \ \forall k = \{1, \dots, K\}$.

1-2- Initialize the scheduling functions

$$h^k(\xi, \dot{\xi}) = \frac{P^k(k)P^k([\xi\ \dot{\xi}]^T|k)}{\sum_{i=1}^{K} P^i(i)P^i([\xi\ \dot{\xi}]^T|i)}$$

2- Optimization step:

Solve:

$$\min_{A^k, P} C(\theta) = \sum_{m=1}^{M} \|\ddot{\xi}_m - \ddot{\xi}\|^2\ \forall k \in \{1, \dots, K\}$$

s.t.:

$$\begin{bmatrix} 0 & I \\ A_1^k & A_2^k \end{bmatrix}^T P + P \underbrace{\begin{bmatrix} 0 & I \\ A_1^k & A_2^k \end{bmatrix}}_{A^k} \prec 0$$

$$0 \prec P, \qquad P^T = P$$

where :

$$\begin{bmatrix} \dot{\xi} \\ \ddot{\xi} \end{bmatrix} = \sum_{k=1}^{K} h^k(\xi, \dot{\xi}) \begin{bmatrix} 0 & I \\ A_1^k & A_2^k \end{bmatrix} \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix}$$

3.4.1.1 Performance evaluation

The performance of Algorithm 3.2 is assessed by two sets of simulations. The first set of simulations is designed to qualitative and qualitatively assess the performance of Algorithm 3.2a in learning a stable first order dynamical system on a data set illustrated in Figure 3.4. Here, we compare the performance of Algorithm 3.2a with SEDS (Khansari-Zadeh and Billard, 2011). In the second set of simulations, the performance of Algorithm 3.2b is evaluated against a library of human handwriting motions (Khansari-Zadeh and Billard, 2011) for approximating a second order dynamical system.

The first set of simulations

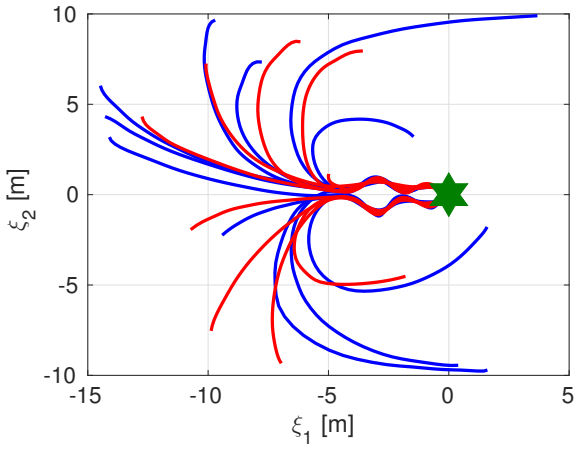
The data set illustrated in Figure 3.4 consists of three different motions and each motion is demonstrated 20 times. The first set of motions (Figure 3.4a) contains non-linearities near the target position. The second set of motions (Figure 3.4b) encapsulate steep changes. The third set of motions (Figure 3.4c) is a combination of cyclic and straight line motions. Each data-set is randomly separated into a training and a testing data sets; i.e., the blue lines are the training data set and the red lines are the testing data set.

The results of SEDS and Algorithm 3.2a are preliminary qualitatively compared for different numbers of the Gaussian components; see Figure 3.5. In general, as it is illustrated, both algorithms are capable of encoding the general shape of demonstrated motions. However, each algorithm has it own advantages and disadvantages. Ours is advantageous in encoding more local non-linearity in a DS. It is clearly visible in Figure 3.5a and Figure 3.5c where the motion of the DS at the neighborhood of the target point is adapted to the non-linearities of the demonstrations. Even though, this feature is useful in approximating very non-linear dynamics, it can also result in jerky motions. For instant, as can be seen in Figure 3.5a, encoding the non-linearities of the demonstrations comes with the cost of jerky mentions.

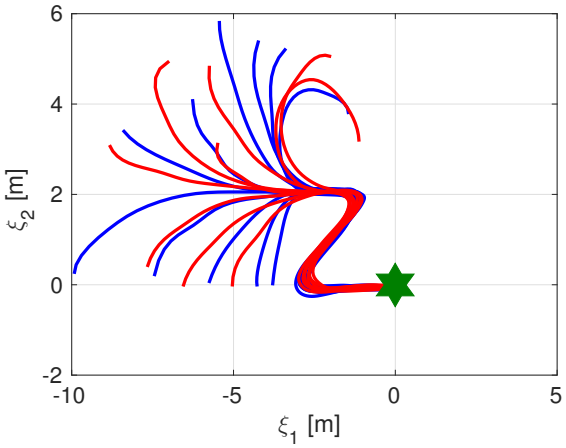
In the next step, Algorithm 3.2a is systematically compared with SEDS in terms of computation cost and the optimality of the parameters. The training and testing data sets are illustrated in Figure 3.4a. The performance metric for the optimality of the parameters is

$$\dot{e} = \frac{1}{M}\sum_{m=1}^{M}\left\|1 - \frac{\dot{\xi}^T\dot{\xi}_m}{\|\dot{\xi}\|\|\dot{\xi}_m\|}\right\| \tag{3.4.12}$$

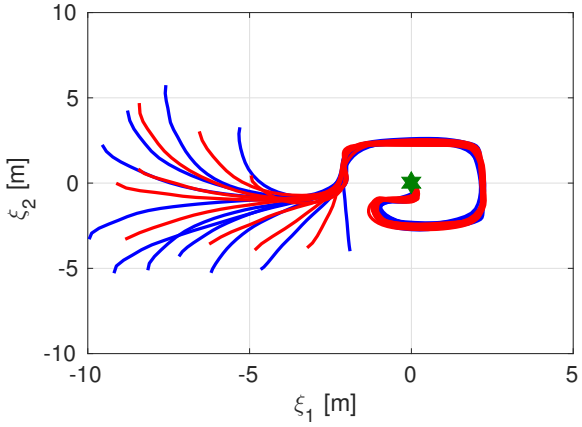
, which calculates the differences between the directions of $\dot{\xi}$ and $\dot{\xi}_m$. Figure 3.6a compares the optimality of the parameters of the Algorithm 3.2a and SEDS, where the number of the Gaussian components is 10 and both algorithms are run 25 times. Clearly the performance of Algorithm 3.2a is significantly better than SEDS. Another advantage of the proposed algorithm is its computational cost. The computation costs of both algorithms are systematically compared. The results are illustrated in Figure 3.6b and Figure 3.6c.



(a) The first motion

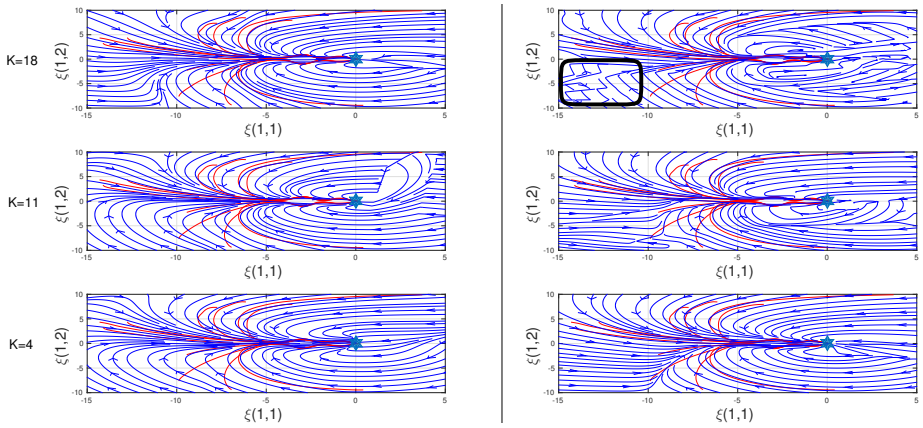


(b) The second motion

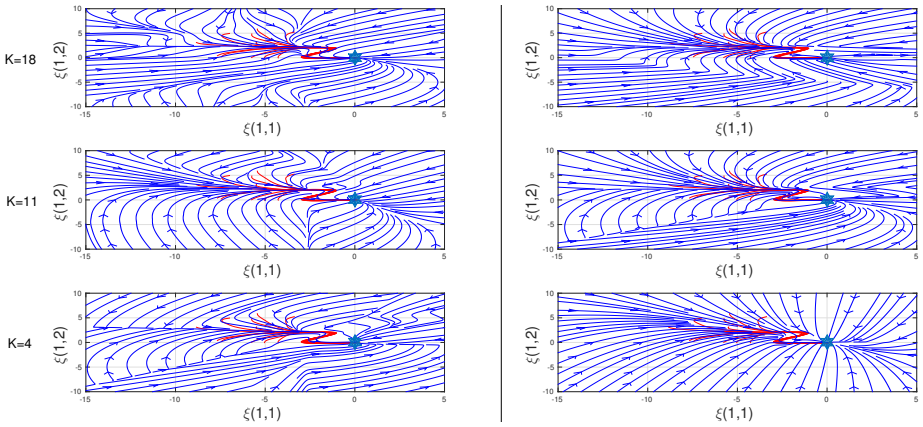


(c) The third motion

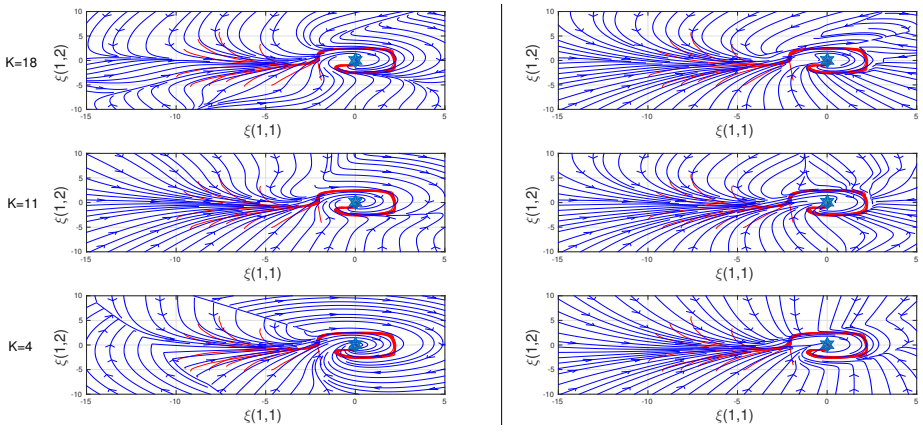
Figure 3.4: Illustration of the data-set. There are 20 demonstration pairs in each class and approximately half of them are the test data and the other half are the training data points. The training and testing data-sets are indicated by blue and red lines, respectively.



(a) The first set of motions. For $K = 4$ and $K = 18$, SEDS fails to encode the non-linear motions near the target and approximate them with a straight line. As it is highlighted by a box, Algorithm 3.2a results in a jerky motions for $K = 18$.



(b) The second set of motions. Both algorithms could encode the steep changes in the demonstrations.



(c) The third set of motions. For $K = 4$ and $K = 11$, SEDS fails to encode the straight line motions near the target as the cyclic motions are dominating.

Figure 3.5: A qualitative comparison between the results of the SEDS algorithm and Algorithm 3.2a to estimate a first order stable dynamical system.

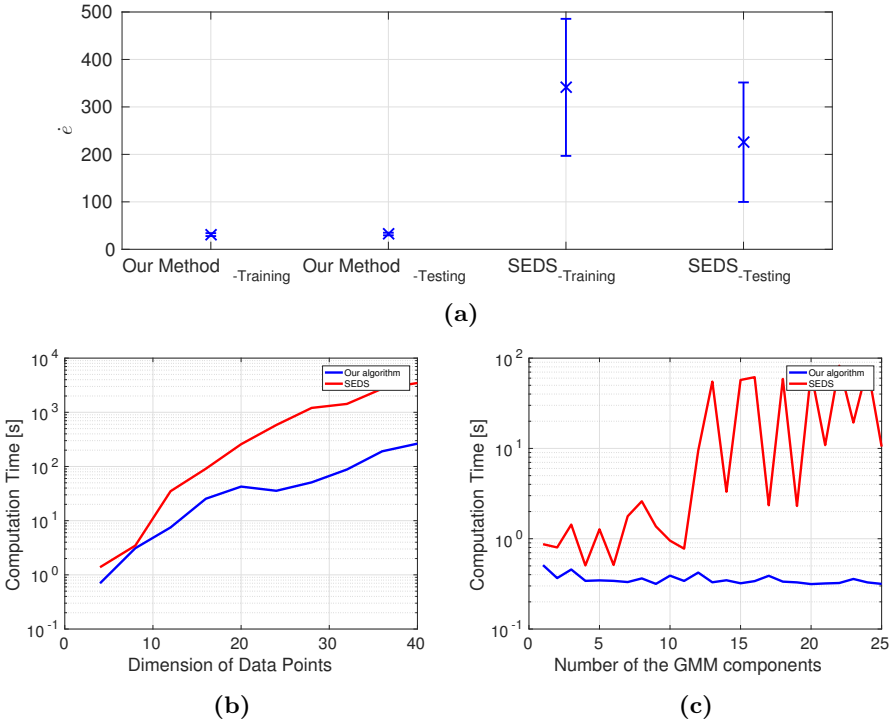


Figure 3.6: In Figure 3.6a, the results of the cross validation are illustrated. Figure 3.6c illustrates the training time where the number of the components varies from 1 to 25. In Figure 3.6b, the number of the Gaussian components is 10.

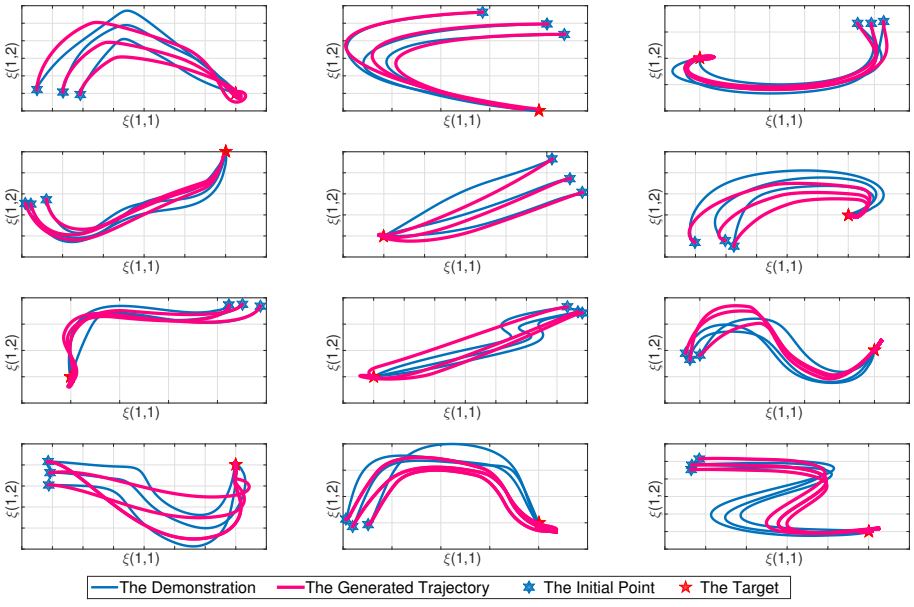


Figure 3.7: The qualitative performance evaluation of our algorithm. The red lines are the learned DS and the blue lines are the training data-points.

Table 3.3: Performance of Algorithm 3.2b in learning 11 human handwriting motions (Khansari-Zadeh and Billard, 2011).

Average/Range of $\dot{e}$	Average/range of Number of the Gaussian components	Average/range of Training Time (Sec)
15.94 $[6.85 - 22.35]$	2.8182 $[1 - 6]$	78.91 $[7.46 - 244.11]$

The second set of simulations

In the second set, the performance of Algorithm 3.2b is evaluated against a library of human handwriting motions (Khansari-Zadeh and Billard, 2011) for approximating a second order dynamical system. Figure 3.7 shows the qualitative results of the estimated motions. In all the experiments, we ran the initialization several times, and we illustrate the result from the best trial. The quantitative results of the proposed method is represented in Table 3.3.

3.4.2 Constructing the training data-set for the soft catching scenario

In the previous section, we present an algorithm to approximate the parameters of the LPV based DSs using a set of demonstrations. In this section, we seek to construct the training data-set $\{\xi_m, \dot{\xi}_m, \ddot{\xi}_m\}$ that encompass examples of kinematically and dynamically feasible trajectories of the end-effector to the intercept point. Moreover, as catching is an extremely rapid action, the training data-set should be a representative of the fastest feasible motions of the robot.

As it would be difficult to have these provided by a human expert, as kinesthetic teaching would not make it possible to move the arm at its maximal speed, we opt for generating the desired demonstrations through an off-line optimal control problem; i.e., Algorithm 3.4. The algorithm consists of two main steps. In step 1, the initial r_I and the final r_D positions of the end-effector are chosen inside the workspace of the robot. In step 2, the end effector is moved with maximum acceptable velocity and acceleration along a straight line from the initial position to a set of intercept points r_D located in its workspace. Since the maximum feasible velocity and acceleration of the end-effector depend on the joint configuration, these need to be calculated at every step. To relax the constraints of the optimization problem, (Alg-2-4) is defined as an inequality constraint. (Alg-2-1) and (Alg-2-2) guarantee the feasibility of the motion in the velocity and acceleration levels, respectively. Note that, this algorithm does not minimize the motion duration, but the generated motion is the fastest at each step time. In Algorithm 3.4, q, $\dot{q}_{max}$, $\ddot{q}_{max} \in \mathbb{R}^m$ are the joint configuration, the maximum acceptable joint velocity and acceleration, respectively. ξ_m is the end-effector's state. $\mathbb{F} : \mathbb{R}^m \to \mathbb{R}^D$ is a known forward kinematic function for the robot. $J(q) \in \mathbb{R}^{D \times m}$ is the Jacobian matrix. ε is a small positive number. The constructed data-set is illustrated in Figure 3.8.

Table 3.4: Pseudo-code for generating the fastest kinematically feasible demonstrations.

Step 1: *Initialization*
Set $i = 1$ and define a fixed initial end-effector position r_I.
$\xi_m[i] = r_I, \quad \dot{\xi}_m[i] = [0], \quad \ddot{\xi}_m[i] = [0]$
$q[i] = \mathbb{F}^{-1}(r_I), \quad \dot{q}[i] = [0], \quad \ddot{q}[i] = [0]$
Randomly define an attractor (r_D) inside the workspace of the robot.
Step 2: *Trajectory planning*
While $\|r_D - \xi_m[i]\| \geq \varepsilon$

$$\ddot{q}[i+1] = \underset{\ddot{\mathbf{q}}}{\text{argmax}}(\|\dot{\mathbf{q}}[i+1]\|)$$

subject to:

$$-\ddot{q}_{max} \leq \ddot{\mathbf{q}}[i+1] \leq \ddot{q}_{max} \qquad \text{(Alg-2-1)}$$
$$-\dot{q}_{max} \leq \dot{\mathbf{q}}[i+1] \leq \dot{q}_{max} \qquad \text{(Alg-2-2)}$$
$$\dot{q}[i+1] = \dot{q}[i] + \Delta t \ddot{q}[i+1] \qquad \text{(Alg-2-3)}$$
$$\| \frac{r_D - \xi_m[i]}{\|r_D - \xi_m[i]\|} - \frac{J(q[i])\dot{q}}{\|J(q[i])\dot{q}\|} \| \leq \epsilon \qquad \text{(Alg-2-4)}$$

Calculate the next joint configuration:
$\dot{q}[i+1] = \dot{q}[i] + \Delta t \ddot{q}[i+1]$.
$q[i+1] = q[i] + \Delta t \dot{q}[i+1]$.
$\xi_m[i+1] = \mathbb{F}(q[i+1])$.
$\dot{\xi}_m[i+1] = J(q[i+1])\dot{q}[i+1]$.
$\ddot{\xi}_m[i+1] = \dot{J}(q[i+1])\dot{q}[i+1] + J(q[i+1])\ddot{q}[i+1]$.
$i = i + 1$.
End

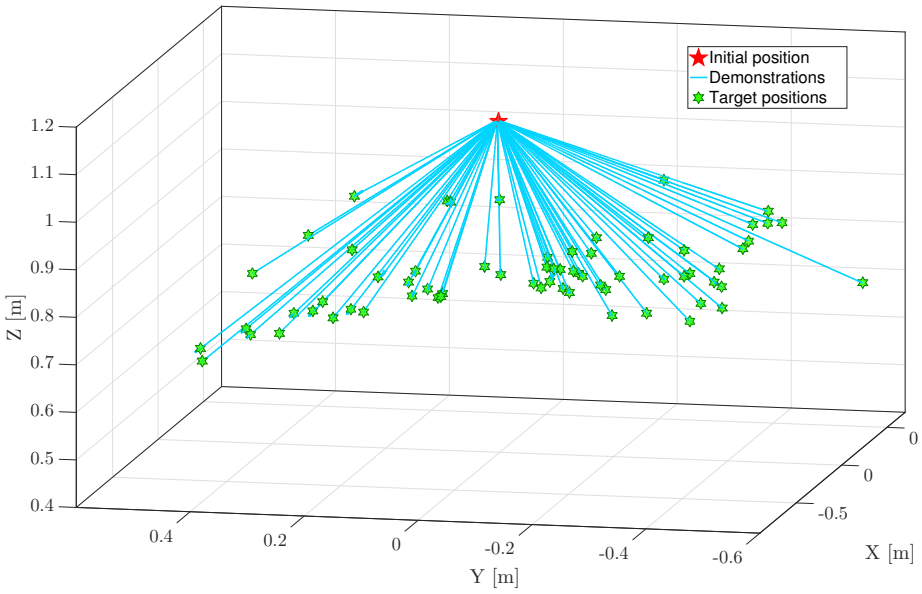


Figure 3.8: The data-set contains 67 demonstrations. The initial point is fixed to the robot's candle position.

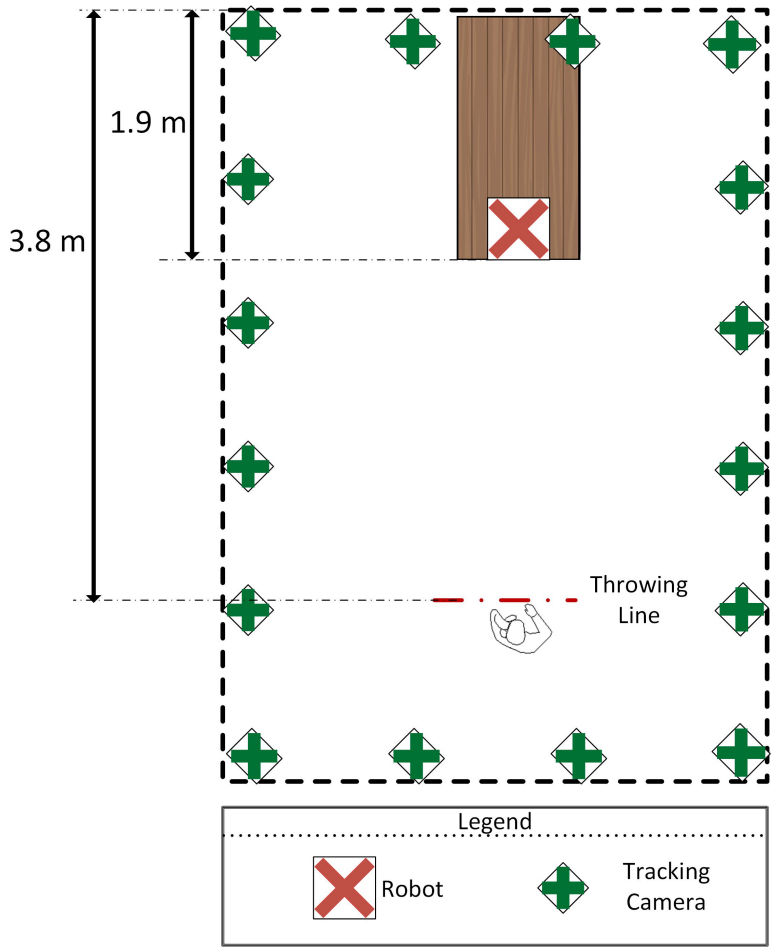


Figure 3.9: Top view illustration of the set-up. The operator stands in-front of the robot and throws the object. The motion of the flying object is captured by 18 cameras.

3.5 Soft Catching In Flight Objects: Experimental Results

The performance of the proposed framework is evaluated on a real platform, 7 DOF robot arm, KUKA LBR IIWA mounted with a 16 DOF Allegro hand. The output of the dynamical system (3.3.5) is converted into the 7-DOF joints state using the velocity based control without joint velocity integration (Nakanishi et al., 2005b). In order to avoid high torques, the resultant joint angels are filtered by a critically damped filter. The robot is controlled in the joint position level at a rate of 500 Hz. As the joint position controller of the robot is a high gain perfect tracking controller and to avoid unexpected noises and delays in measuring the joint position of the robot, (3.3.5) runs in closed-loop via computing the current end-effector position by using the filtered resultant joint angels; see Figure 3.2. A top-view sketch of the set-up is illustrated in Figure 3.9.

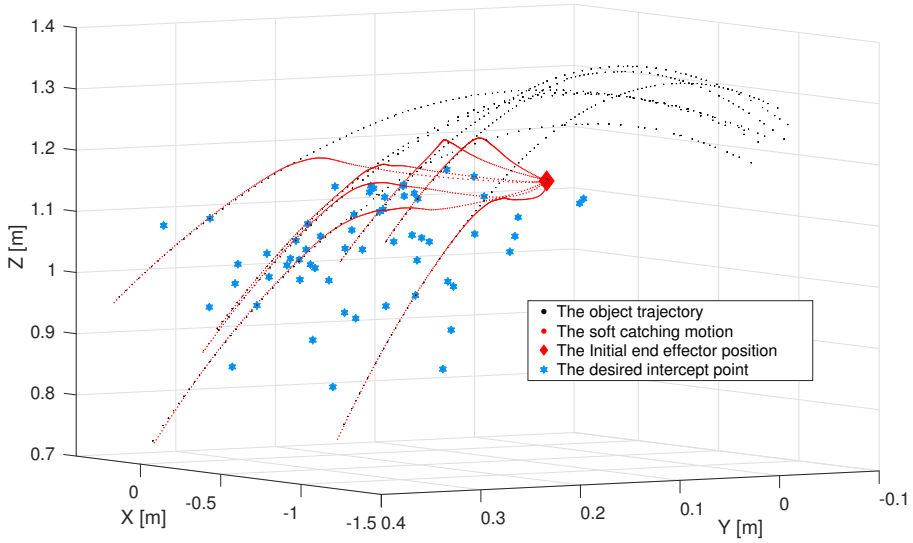


Figure 3.10: The final intercept points in the soft catching experiments. The initial position of the palm is $\begin{bmatrix}-0.05 & 0.00 & 1.134\end{bmatrix}$m. For clarity of the illustration, only seven examples of the soft catching motions and the object trajectories are shown. The object trajectories are plotted from the first points till the stop points. The first point is the first object position which is used for predicting the feasible intercept position. The experimental results verifies that the catching motion intercepts the object at the desired point with the desired velocity.

In order to coordinate the motions of all joints –the arm and the fingers joints–, the coupled dynamical system (CDS) model (Shukla and Billard, 2012a) is utilized to generate the fingers motion. This approach consists of coupling two different dynamical systems; i.e., the end-effector motion and the fingers motion. The motion of the end-effector is generated independently from the fingers states, while the fingers motion is a function of the state of the end-effector and the object. The metric of the coupling is the distance between the end-effector and the object ($\|\xi - \xi^0\|$). As a result, the fingers close when the object gets inside the hand and they reopen when the object moves away.

We choose three objects with different stiffness; a very stiff small plastic ball, a stiff fabric brick and a semi-stiff toy. The objects are almost impossible to catch with the hard catching approach (Kim et al., 2014) as they bounce out of the hand instantaneously, see the video https://youtu.be/FxvVJzb61js. The position of the objects are captured by the *Optitrack* motion capture system from *Natural point* at 240 Hz. Since the control loop is faster than the capturing system, the predicted position of the object is used as the object position in (3.3.5).

To validate the algorithm, the experiment was repeated 20 times for each object. The objects were thrown by a human operator. Data of the experimental results are summarized in Table 3.5. The initial position of the object is randomly changed. As the plastic ball and the brick are stiffer than the toy, a softer interception is required for accomplishing the catching. Hence, the experimenter

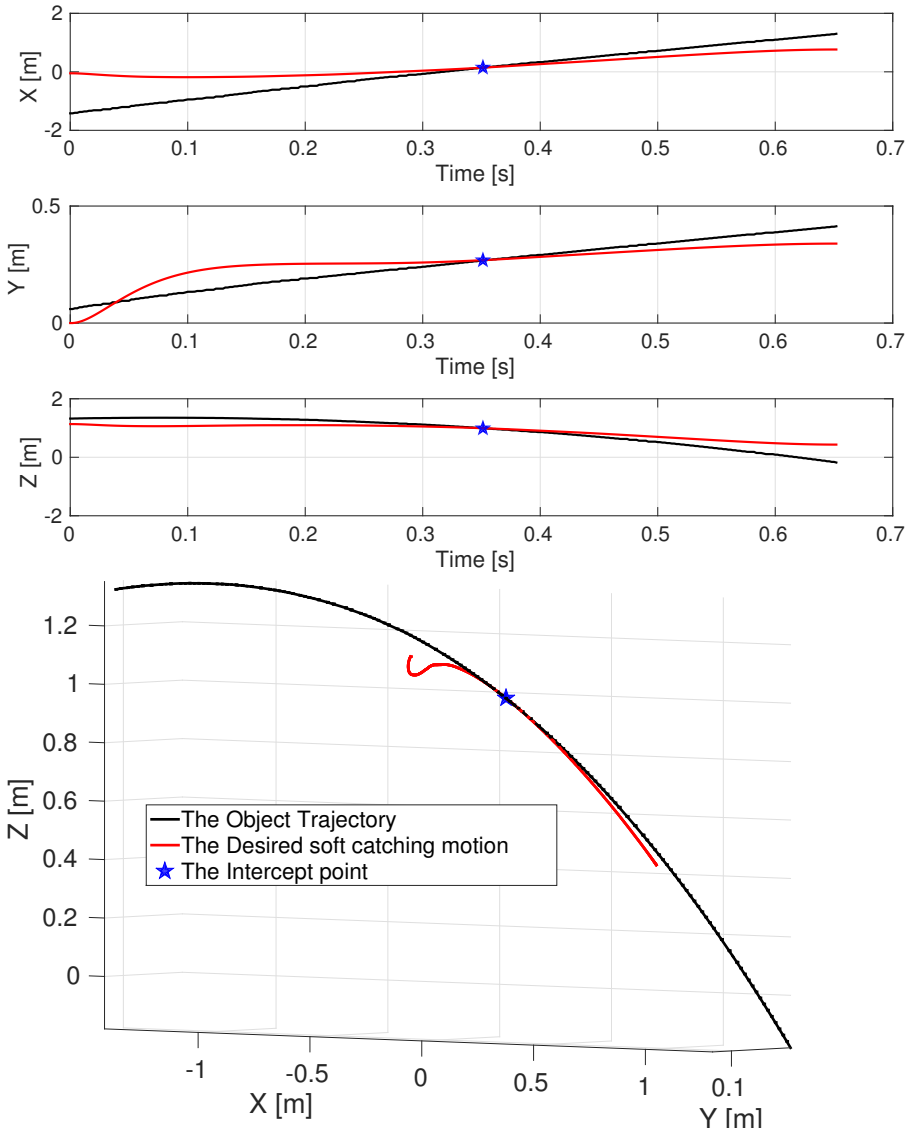


Figure 3.11: The position of the end-effector generated by the dynamical system (3.3.5). The illustrated object trajectory is the predicted trajectory of the uncaught object. This trajectory is illustrated from the first point till the stop point. The initial value of γ is 0.2 and Algorithm 1 maximizes it with respect to the kinematic constraints of the robot. As expected, the output of (3.3.5) softly intercepts the objects trajectory at the desired intercept position. In order to stop the robot, the velocity of the robot is linearly reduced during the post-interception period ($0.3s$).

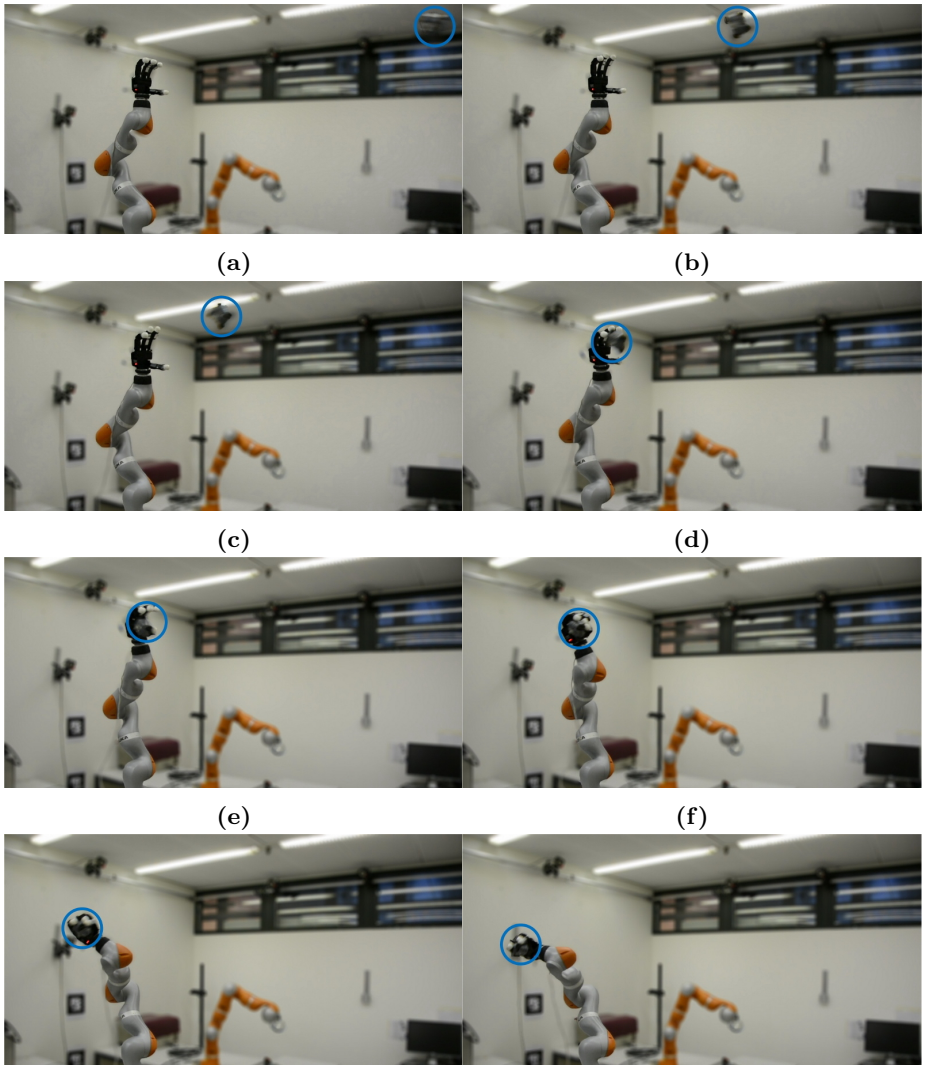
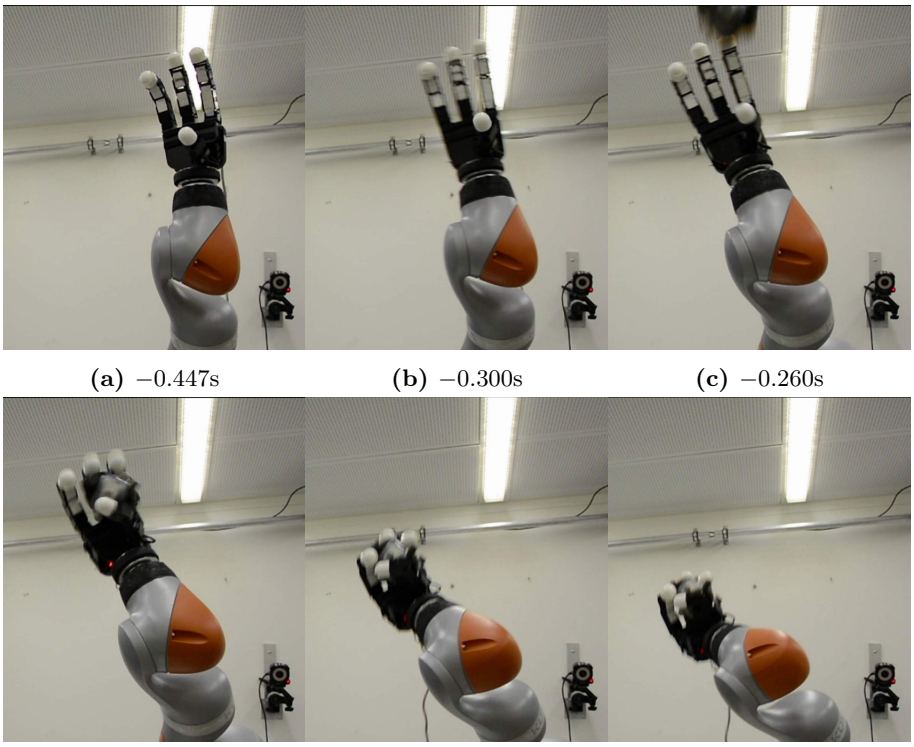


Figure 3.12: The brick is thrown. In (a), the object trajectory prediction algorithm is being initialized. (b) is approximately the first point. (e) is the interception. One can stop the robot the robot in (g) as the fingers are closed. But it might damage the robot. A corresponding video is available at https://youtu.be/l9UFsRAM_XM

Table 3.5: The details of the soft catching experiments. All the positions are in respect to the base of the robot. The throwing positions are randomly chosen. The robot does not move till the first intercept point is calculated. As the first $0.4m$ of the throwing in x direction is used to initialize the object prediction trajectory (Kim and Billard, 2012). Time of the flight is the duration of the object flight from the first point till the intercept position. Softness is the softness of the object interception. All the object are thrown 20 times and are caught by the hard (Kim et al., 2014) and our soft catching algorithms.

	The small ball
Stiffness	Too stiff
Throwing position (m)	$[-1.68 \pm 0.14 \quad , -0.07 \pm 0.02 \quad , 1.18 \pm 0.06]$
First point (m)	$[-1.29 \pm 0.14 \quad , -0.03 \pm 0.03 \quad , 1.30 \pm 0.05]$
Time of flight (s)	0.32 ± 0.03
Softness (γ)	0.67 ± 0.06
Soft catching success rate	%70
Hard catching success rate	%0.0
	The brick
Stiffness	stiff
Throwing position (m)	$[-1.84 \pm 0.07 \quad -0.05 \pm 0.02 \quad 1.11 \pm 0.04]$
First point (m)	$[-1.46 \pm 0.07 \quad -0.01 \pm 0.03 \quad 1.25 \pm 0.030]$
Time of flight (s)	0.33 ± 0.02
Softness (γ)	0.66 ± 0.04
Soft catching success rate	%70
Hard catching success rate	%0.0
	The toy
Stiffness	semi-stiff
Throwing position (m)	$[-1.52 \pm 0.62 \quad -0.02 \pm 0.07 \quad 1.14 \pm 0.27]$
First point (m)	$[-1.14 \pm 0.62 \quad 0.03 \pm 0.07 \quad 1.22 \pm 0.39]$
Time of flight (s)	0.32 ± 0.00
Softness (γ)	0.50 ± 0.02
Soft catching success rate	%75
Hard catching success rate	%5.0



(a) −0.447s **(b)** −0.300s **(c)** −0.260s

(d) −0.193s **(e)** −0.077s **(f)** 0.0 s

Figure 3.13: Snapshots of the finger motions. The object is intercepted in (d) and caught in (f). It is important to note that the closure time for fingers varies with the incoming object speed.

placed himself farther away from the robot than when throwing the toy. First, the prediction of the object's trajectory requires some time and uses almost all of the first 0.4 meter of the object flight in x. In addition, to ensure that the object travels at a reasonable speed, leaving enough time for the robot to travel to the desired position, a distance between the robot and the initial position of the object should be no more than $1.5m$ which approximatively results in $0.33s$ flight time. Due to imperfect prediction of the object's trajectory, the feasible intercept point needs to be updated and redefined during the catching. The new feasible intercept point is chosen in the vicinity of the previous one to minimize the convergence time and improve the success rate of catching. The feasible intercept point is updated approximately 29 times during the flight time. The intercept points are illustrated in Figure 3.10. In this figure, the origin is the position of the base of the robot. As the right hand of the Allegro hand is used, the experimenter threw the objects mostly to the right side of the robot.

The initial values of γ and $\dot{\gamma}$ in (3.3.5) are set to 0.2 and 0, respectively. COBYLA algorithm (Powell, 1994) of Nlopt library (Johnson, 2015) is used for solving the closed loop optimal control, where the maximum optimization time is set 0.001s . An example of the desired robot trajectory and the unperturbed object predicted trajectory are shown in Figure 3.11. As expected, the end-

effector converges to the object at the intercept point and continues to track the object's predicted trajectory. The snapshots of the real robot experiments are shown in Figure 3.12 and Figure 3.13. As the closure time of the hand is approximatively 0.1s, one can immediately stop the robot when the hand is closed on the object. However, this may damage the robot. Hence, we reduce the end-effector velocity in $0.3s$ to avoid high torques.

The overall success rate of the soft catching reached %71.6, see Table 3.5. To compare and to assess the improvements, the experiments were repeated with the similar initial conditions for the *hard* catching scenario. The overall success rate was very low and did not exceed %1.6. Visual inspection of the data and video confirmed that this poor result for the hard catching scenario was essentially due to the fact that the objects bounce out of the hand. The causes of failure for the soft catching strategy can be categorized into three different categories.

i) The main cause of the failure is due to the inability of generating an accurate joint-level motion corresponding to the desired end-effector trajectory; the toy (3 out of 5), the brick (3 out of 6) and for the plastic ball (2 out of 6). As the motion is too fast, the end-effector does not accurately track the desired motion. The tracking error between the desired and the actual end-effector positions is approximately $\begin{bmatrix} 0.03 \pm 0.02 & 0.01 \pm 0.01 & 0.02 \pm 0.01 \end{bmatrix}^T m$. This error results in situations where the object is hit by the thumb or undesired parts of the hand and bounces away.

ii) As the Allegro hand has only four fingers, there is a space between the fingers and the palm at the grasp configuration. The small plastic ball can escape the grasp using this space, (2 out of 6). For the other objects, this issue is negligible as there are big enough to be caught with four fingers.

iii) To track the object, all the markers must be visible to the cameras. In four cases, the brick (3 out of 6) and the plastic ball (1 out of 6), the tracking started very late, for lack of visible markers. As a result, the robot was not able to reach the desired intercept point on time and interception occurred at an undesired point. In these cases, the side of the hand hit the object or the interception was not soft enough. As the intercept point is approximately updated 29 times during flight, the first prediction of the object trajectory plays a main role in defining the intercept point. If the initial prediction of the object's trajectory is very inaccurate, the updated intercept points will be far from each other. As a result, (3.3.5) does not converge to the latest desired trajectory on time. This was the case for trials using the toy (2 out of 5). Finally, in one trial with the plastic ball (1 out of 6), the first point is too close to the robot and the trajectory prediction does not work. In this case, the object hits and bounces away. However, the robots tried to reach the bounced object.

3.6 Conclusion and Discussion

In this chapter, we proposed a framework to compliantly intercept a moving object with a robotic arm where the mass of the object is very small in comparison to that of the arm and the force at the impact was negligible in comparison to the robot's natural inertia. Two important constraints need to be satisfied in the compliant interception; namely reaching the object's trajectory at the right place with a velocity aligned with that of the object. The motion should be fast enough to intercept the object on *time*. This, of course, depends on having appropriate hardware. If provided with a robot that can travel fast enough within the required time, then, our algorithm ensures that the robot will intercept the object *on time*, at the *desired point* with the *desired velocity direction*.

Proof of asymptotic stability was done by using Lyapunov stability theorem. Specifically, we showed that our LPV based dynamical system asymptotically converges to the object trajectory and intercept it exactly at the desired point. In order to improve the softness of catching, we proposed a closed loop optimal control problem to maximize the value of the softness subject to the kinematic constraints of the robot. Furthermore, a new GMM based method is proposed for accurately approximating and modeling the parameters of LPV systems. Approximating the parameters of the LPV systems via a GMM based model has its own advantages and disadvantage. Using GMM is advantageous in that it can accurately model the training data points. Moreover, the scheduling parameters are of class C^∞ and the transitions between the scheduling parameters are smooth. However, as the proposed learning algorithm is not convex; i.e., the performance of the learned dynamical system depends on the initialization.

Proposition 1 shows that the system governed by (3.3.5) converges asymptotically to the object trajectory and intercepts it at the desired point. As there is no constraint on the magnitude of the eigenvalues ($|\lambda_{A_j^i}|$) of A_j^i $\forall(j,i) \in \{(1,1),\dots,(1,K),(2,1),\dots,(2,K)\}$, there is no guarantee that (3.3.5) is fast enough to converge to $\gamma\xi^O$ on time. To successfully intercepts an object, the arm should arrive in a neighborhood of the desired trajectory $\gamma \begin{bmatrix}\xi^O & \dot{\xi}^O\end{bmatrix}^T$ before the catching time; i.e., $\|\xi(T^*)-\gamma\xi^O(T^*)\| \leq \varepsilon$ and $\|\dot{\xi}(T^*)-\gamma\dot{\xi}^O(T^*)-\dot{\gamma}\xi^O(T^*)\| \leq \varepsilon$, where ε is a small positive number. As a simple example, consider a case in which the workspace of the robot is a sphere of a diameter of 100cm and the minimum flight time for the object is 0.3s with ϵ =1cm and assume (3.3.5) is a critically damped system, then $|\lambda_{A_j^i}|$ must be approximately greater than 22. In practice, $22 < |\lambda_{A_j^i}|$ is not only very conservative constraint but also may result in a dynamical system which generates kinematically infeasible motions. To address this challenge, a potential direction would be extracting the desired intercept posture subject to the dynamic constraints of the robot and the success rate of the intercept posture.

Throughout the proofs, we assume that the desired intercept point is a fixed

attractor. In practice, due to the imperfect prediction of the object trajectory, the feasible intercept postures need to be updated all the time. However, it usually does not affect the convergence of the system as the new feasible intercept point is chosen in the vicinity of the previous ones; i.e., the convergence duration is too small. Besides, thanks to the second order LPV dynamical system, the updating does not cause discontinuity at the velocity profile.

In this chapter, the desired intercept point is defined as a point on the predicted trajectory of the object which is kinematically reachable for the robot. Nevertheless, one is not restricted to use this criterion to select the intercept point. For example, this point can be selected such that the time of the flight[4] is maximized. The other criterion can be the manipulability of the intercept point.

As the sole knowledge of the object is its location, any inaccuracies in measuring the object's position would deteriorate the performance of the controller. As discussed in Section 3.5, to track the object, all the markers must be visible to the cameras. However, the object's tracking is obscured when the object is very close to the end-effector and the fingers are about to close on it. At this time, the accurate estimation of the interception is not possible. One way to address this problem is through integrating force or tactile feedback into the motion generators.

In the next chapter, we extend this method for a multi-arm scenario where each robot must comply with the motion of the object and with the motion of the other arms, so that all arms intercept the object simultaneously.

[4] As mentioned before, the time of the flight is defined as the duration of the object flight from the first point till the intercept point.

Chapter 4

Coordinated Multi-Arm Motion Planning

"Of all the things I've done, the most vital is coordinating those who work with me and aiming their efforts at a certain goal."

Walt Disney
1901 – 1966

4.1 Introduction

Humans have a remarkable way of controlling their hands' movements in everyday life. The use of both hands allows for highly complex manipulation of heavy or bulky objects. Accomplishing these tasks with one arm is mostly impossible, mainly because a single arm has a limited workspace. Moreover, the dexterity and flexibility required for accomplishing such tasks are beyond single arm's capabilities.

This is also true for robotic systems. A dual, or by extension multi, arms robotic system extends the workspace of a single robotic arm such that complex manipulations of heavy or large objects would be feasible. One can envision a plethora of applications in smart-factories or smart-buildings, that would benefit from such strategies. Examples include, grabbing, catching, carrying, lifting and manipulating a object traveling on a cart or a running conveyor belt, carried by humans or even lying towards the multi-arm robot system; Figure 4.1b. To accomplish these applications, unlike the soft catching scenario presented in Chapter 3, we breakdown compliance into two sub-levels. In the first level, each arm must comply with the motion of the other arms in both task and joint spaces. In the second level, the resultant motion of the arms must comply with the object's motion.

The first level of compliance imposes that the robots move in coordination with each other. This is necessary not only to ensure that the arms simultaneously intercept the object, but also to avoid collisions between their end-effectors and the rest of the arms while they adapt to the moving object's motion. The second level of compliance imposes position and velocity constraints at the object's interception. Position constraints ensure that the planned motion of each

robot's end-effector is coordinated with the feasible reaching positions of the object. Whereas the velocity constraints allow readjustments of the hand, palm and fingers posture while there are uncertainties in the hand orientation and position.

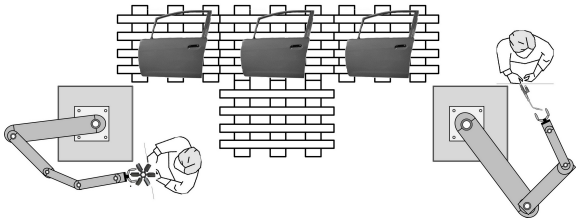
Handling multiple constraints simultaneously is a problem that could be addressed by using standard optimal control approaches. These, however, are time consuming and will not converge within a few milliseconds, which is the expected reaction time necessary for all arms to rapidly and simultaneously reach and adapt to the moving object.

Interestingly, a multi-arm system could provide not only *synchronous* behaviors, as the ones mentioned above, but also *asynchronous* behaviors, where each robot follows its own goal-oriented task (Figure 4.1a). These tasks can be considered as typical examples of co-worker scenarios for the industry 4.0 (Hermann et al., 2016), where, based on the tasks' requirements, each robot should be able to behave *synchronously* and *asynchronously*. More importantly, it must be able to smoothly transition between them.

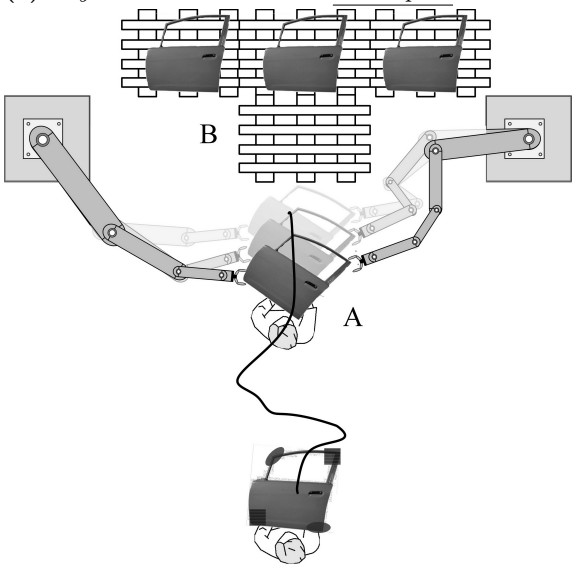
In this chapter, we offered a unified control architecture for compliantly reaching for grabbing a moving object by a multi-arm robotic system. The approach consists of a *virtual object* based-DS control law that generates autonomous motions for a multi-arm robot system. Our approach can generate two types of behaviors: (i) multi-arm *asynchronous* task-space behaviors, where each robot has its own target or desired motion (Figure 4.1a) and (ii) multi-arm *synchronous* task-space behaviors, where the robots' task is to comply with each other's motions to *simultaneously* reach-for a moving object (Figure 4.1b). To provide a smooth transitioning between these two behaviors, we introduce the notion of *synchronization allocation*. Given the motion of the object and the joint workspace of the multi-arm system, each arm is being continuously allocated to a desired behavior. While being allocated to the *synchronous* behavior, control of the robots is taken over by the *virtual object*-DS. By this, we unify two levels of compliance into one level; namely the virtual object must comply with the motion of the object. While allocated to the *asynchronous* behavior, the robots are controlled independently, each with their own goal-directed stable DS. Both behaviors are encoded in a single dynamical system which is expressed as a Linear Parameter Varying (LPV) system subject to stability constraints, that ensure convergence to the object or targets.

Further, we provide compliance in the joint-space by introducing a centralized inverse kinematics (IK) solver under self-collision avoidance (SCA) constraints; formulated as a quadratic program (QP) and solved in real-time. These SCA constraints are introduced as linear inequality constraints in the optimization problem in the form of a continuous "SCA Boundary function" and its gradient.

Note that the work presented in this chapter was published in three papers with Nadia Figueroa, currently PhD student at LASA. This chapter reports



(a) *Asynchronous* Multi-Arm Task-Space Behavior.



(b) *Synchronous* Multi-Arm Task-Space Behavior.

Figure 4.1: Illustration of Multi-arm Task-Space Coordination. (a) *Asynchronous* task-space behavior, where each robot has its own target and is endowed with an independent stable DS to generate desired motions; (b) *Synchronous* task-space behavior, where the robots coordinate with each other to simultaneously reach-for a moving object.

solely on the parts that were developed and implemented by myself. The construction of the self-collision avoidance data-set, which was developed collaboratively and implemented by me, and learning approximating SCA boundary function, which was solely developed and implemented by Nadia Figueroa, are not reported in this thesis. Moreover, the real-world experiments were developed and conducted collaboratively.

Related work for this chapter is reviewed in Section 2.3. This chapter is structured as follows. Section 4.2 formalizes the problem. In Section 4.4, we present the *multi-arm*-DS, including a formalization and convergence proof of the LPV based DS. Section 4.5 introduces the centralized inverse kinematic solver which can handle SCA constraints. The proposed method is then validated with a dual-arm platform for several coordination and reaching scenarios in Section 4.6. Discussions is presented in Section 4.7.

This chapter corresponds to the following publications:

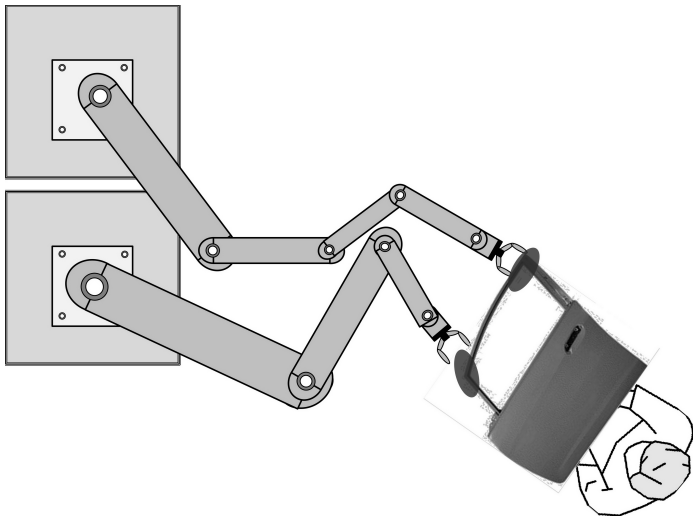


Figure 4.2: Self-Collision Avoidance (SCA) in Joint-Space for both task-space behaviors.

- Mirrazavi Salehian, S. S., Figueroa, N. and Billard, A. (2017) A Unified Framework for Coordinated Multi-Arm Motion Planning. The International Journal of Robotics Research: 2018..
- Mirrazavi Salehian, S. S., Figueroa, N. and Billard, A. (2017) Dynamical System-based Motion Planning for Multi-Arm Systems: Reaching for moving objects. In Proceedings of International Joint Conference on Artificial Intelligence 2017, Melbourne, Australia.
- Mirrazavi Salehian, S. S., Figueroa, N. and Billard, A. (2016) Coordinated multi-arm motion planning: Reaching for moving objects in the face of uncertainty. In Proceedings of Robotics: Science and Systems XVI , Arbor, Michigan, .

4.2 Problem Statement

The main objectives of this chapter is to design a controller which drives the multiple robot system such that

Objective 1: While the robots are allocated to *asynchronous* behavior, each of them should be able to reach the independent targets.

Objective 2: While the robots are allocated to *synchronous* behavior, the robots must simultaneously intercept the object at the desired locations.

Objective 3: The robots should be able to seamlessly transition from *asynchronous* to *synchronous* behaviors and vise versa.

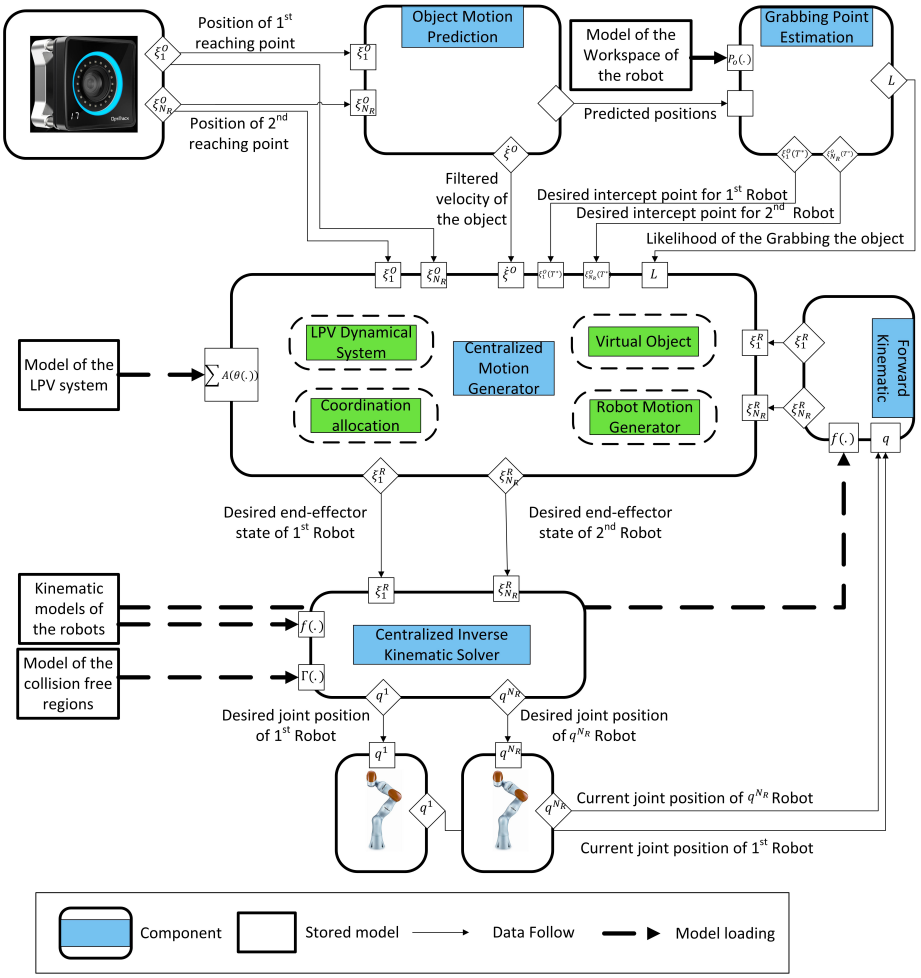


Figure 4.3: Block diagram for coordinated multi-arm motion planning for reaching a large moving object. Where N_R represents the total number of robot arms. T represents the motion prediction duration. Same as the previous chapter, in this chapter, we assume that the low-level controller of the robot is a perfect tracking controller.

Objective 4: All the while, any collision between two arms must be avoided.

In this chapter, a known model of the dynamics of the object or its motion is not assumed. The sole knowledge about the object is its *coupled feasible reaching points*, which are the preferred reaching positions and orientations on the object, specified by the user (see Figure 4.4).

In order to achieve the mentioned objectives, while the target object is in motion, several problems need to be solved simultaneously: (i) prediction of the object's trajectory; (ii) computing intercept points for each arm and (iii) planning coordinated motion of the robotic arms towards their corresponding intercept points.

An overview of the proposed framework is illustrated in Figure 4.3. As seen on the illustration, sub-component "Grabbing Point Estimation" computes the intercept point. It uses an estimate of the reachable workspace of the multi-

Table 4.1: Nomenclature

Variable	Domain	Definition
ε	$\in \mathbb{R}_{>0}$	A small positive number.
k	$\in \mathbb{R}_{>0}$	A large constant positive number.
$\mathbf{k}$	$\in \mathbb{R}_{>0}$	A constant positive number.
δ_j	$\in \mathbb{R}_{>0}$	Minimum likelihood threshold of j^{th} robot's workspace .
δ	$\in \mathbb{R}_{>0}$	Minimum joint likelihood threshold.
T^*	$\in \mathbb{R}_{>0}$	Time when the object is kinematically reachable
N_R	$\in \mathbb{N}$	Number of the available robot arms.
τ_{c_i}	$\in \mathbb{R}_{(0,1)}$	Synchronization parameter of i^{th} robot.
γ	$\in \mathbb{R}_{(0,1)}$	Coordination parameter.
d_n	$\in \mathbb{N}$	Dimension of the states of the virtual/real object or one robot.
d_N	$\in \mathbb{N}$	Dimension of the states of all the robots in total.
d_{s_i}	$\in \mathbb{N}$	Number of scheduling parameters of i^{th} robot.
d_{q_i}	$\in \mathbb{N}$	Number of the joints of i^{th} robot.
d_Q	$\in \mathbb{N}$	Number of the joints of all the robots.
d_{c_i}	$\in \mathbb{N}$	Number of the scheduling parameters of i^{th} robot.
q^i	$\in \mathbb{R}^{d_{q_i}}$	Joint angles of i^{th} robot.
q^i_j	$\in \mathbb{R}$	Angle of j^{th} joint of i^{th} robot.
$f(q^i_j)$	$\in \mathbb{R}^3$	Cartesian position of j^{th} joint of i^{th} robot with respect to the world frame.
ξ^R_j	$\in \mathbb{R}^{d_n}$	Position of the j^{th} end-effector.
ξ^V_j	$\in \mathbb{R}^{d_n}$	Position of j^{th} reaching point on the *virtual object*.
ξ^V	$\in \mathbb{R}^{d_n}$	Position of the *virtual object*
ξ^O	$\in \mathbb{R}^{d_n}$	Position of the real object.
ξ^O_j	$\in \mathbb{R}^{d_n}$	j^{th} feasible reaching point on the real object.
${}^j\xi$	$\in \mathbb{R}^{d_n}$	ξ in the reference frame of j^{th} robot base.
θ^W		Set of GMM parameters of N_R workspace models
x^V	$\in \mathbb{R}^{2d_n}$	States of the virtual object's dynamical system.
x^R_j	$\in \mathbb{R}^{2d_n}$	States of j^{th} end-effector .
x^d_j	$\in \mathbb{R}^{2d_n}$	States of the static target of the asynchronous behavior of j^{th} robot.
θ_i	$\in \mathbb{R}^{d_{s_i}}$	Scheduling parameters for pos./orient. dynamics.
$A_i(.)$	$\in \mathbb{R}^{2d_n \times 2d_n}$	Affine dependent state-space matrices for.
$\Gamma(.)$	$\in \mathbb{R}$	Self-collision boundary.

arm system (learned off-line prior to experiment) and an on-line step in which it continuously measures the object's pose from a visual tracking system. Sub-component "Centralized Motion Generator" uses the predicted intercept points and the current end-effector poses to generate the desired end-effector poses of the N_R-robot multi-arm system. It is based on a centralized controller that exploits a forward model of the virtual object's motion. A detailed description of each of these sub-components is presented in the following sections. For simplicity and practicality, we summarize the most relevant notation used throughout this chapter in Table 4.1 and illustrate them in Figure 4.4. The control flow of the entire framework is illustrated in Figure 4.3.

4.3 Object Trajectory and Intercept Point Prediction

If the motion of the object is predictable, for fulfilling the goal of *synchronous* behavior, one can use a model-based prediction approach (Kim and Billard, 2012) and find feasible intercept postures by extending a single-robot arm feasible posture extraction algorithm (such as (Kim et al., 2014)) to N_R-robot arms. In this case, a simple point-to-point motion can be devised to intercept the object. As we assume that the motion of the object is not accurately predictable (e.g when carried by a blind-folded human operator as in the experiments); using a model-based approach would be impractical and limiting. Instead, once the object starts moving towards the robots, a linear model predicts its progress ahead of time and determines a point along its trajectory where the object will become reachable by all robotic arms. We call this point the *feasible intercept point*.

To find the *feasible intercept point*, the reachable workspace of each robot is modeled via a probabilistic classification model $p_j({}^j\xi_i^O;\theta_j^W) \;\; \forall j \in \{1,\dots,N_R\}$, namely a Gaussian Mixture Model as introduced in Section 3.2.1:

$$p_j({}^j\xi_i^O;\theta_j^W) = \sum_{l=1}^{K_j^w} \pi_{lj}\mathcal{N}({}^j\xi_i^O|\mu_{lj},\Sigma_{lj}) \tag{4.3.1}$$

where $\pi_{lj},\mu_{lj},\Sigma_{lj}$ correspond to the prior, mean and covariance matrix of the $l = \{1\dots K_j^w\}$ Gaussian functions, respectively, estimated by using the Expectation-Maximization algorithm (Bishop, 2007). In order to generate the training dataset, all possible postures of each robot are simulated by testing all possible displacements of their joints. If $p_j({}^j\xi_i^O) > \delta_j$, where δ_j is the *minimum likelihood* threshold and it is determined such that the likelihood of 99% of the training points is higher than it, then ${}^j\xi_i^O$, i.e., the i-th reaching position on the object (sub-script) in the j-th robot's reference frame (left super-script), is classified as a *feasible* position for the j-th robot to reach. As the reachable workspaces of each robot are statistically independent from each other, the joint distribution of all workspaces can be calculated by computing the product of distributions, as follows:

$$p(\{{}^1\xi_1^O,\dots,{}^{N_R}\xi_{N_R}^O\};\Theta^W) = \prod_{j=1}^{N_R} p_j({}^j\xi_j^O;\theta_j^W) \tag{4.3.2}$$

where $\Theta^W = \{\theta_1^W,\dots,\theta_{N_R}^W\}$ is the set of parameters for all robot workspaces and $\{{}^1\xi_1^O,\dots,{}^{N_R}\xi_{N_R}^O\}$ are the reaching positions in each robot's reference frame.

The *minimum joint likelihood threshold is* $\delta = \prod_{j=1}^{N_R} \delta_j$. if

$$\exists T^* : \delta < p({}^1\xi_1^O(T^*), \ldots, {}^{N_R}\xi_{N_R}^O(T^*); \Theta), \tag{4.3.3}$$

the object at T^* ($\xi^O(T^*) = \frac{1}{N_R}\sum_{j=1}^{N_R} \xi_j^O(T^*)$) is classified as the *feasible intercept point*. If more than one point on the predicted trajectory is classified as the *feasible intercept point*, we select the closest one, in Euclidean space position, to the robots' end-effectors.

4.4 Dual-Behavior Coordinated Motion Generator

Once the *feasible intercept point* is found, the motion of i-th robot's end-effector $\forall i \in \{1, \ldots, N_R\}$, is generated by following a Linear Parameter Varying (LPV) dynamical system (DS), composed of both *synchronous* and *asynchronous* behaviors, and coupled to the motion of the *virtual object* (ξ^V) (see Figure 4.5) as follows:

$$\dot{x}_i^R = \dot{\tau}_{c_i}\left(x_i^V - x_i^d\right) + \tau_{c_i}\dot{x}_i^V + \mathbf{A}_i\left(\theta_i(x_i^R)\right)\left(x_i^R - x_i^d - \tau_{c_i}(x_i^V - x_i^d)\right) \tag{4.4.1}$$

, where $x_i^R(t) = \begin{bmatrix}\xi_i^R & \dot{\xi}_i^R\end{bmatrix}^T \in \mathbb{R}^{2d_n}$ and $x_i^V(t) = \begin{bmatrix}\xi_i^V & \dot{\xi}_i^V\end{bmatrix}^T \in \mathbb{R}^{2d_n}$ are the states of the i^{th} end-effector and *virtual object*, respectively.[1] The state of the virtual object is used to guide the robots for the *synchronous* behavior. In the case of the *asynchronous* behavior, each i-th robot has its own static target, denoted as $x_i^d = \begin{bmatrix}\xi_i^d & 0\end{bmatrix} \in \mathbb{R}^{2d_n}$. $0 \leq \tau_{c_i} \leq 1$ is the *synchronization allocation* parameter and is of class $\mathcal{C}^1$. $\theta_i(x_i^R) \in \mathbb{R}^{d_{s_i}\times 1}$ $\forall i \in \{1, \ldots N_R\}$ is a vector of scheduling parameters; $\theta_i(x_i^R) = \begin{bmatrix}\theta_{i1}(x_i^R) & \ldots & \theta_{id_{s_i}}(x_i^R)\end{bmatrix}^T$ and for simplicity of the notations, its argument is dropped in the rest of the chapter. $\mathbf{A}_i(.) : \mathbb{R}^{d_{s_i}\times 1} \to \mathbb{R}^{2d_n\times 2d_n}$ is the affine dependence of state-space matrices on the scheduling parameter and the state vectors:

$$\mathbf{A}_i(\theta_i) = \sum_{k=1}^{d_{s_i}} \theta_{ik} A_{ik} \quad A_{ik} \in \mathbb{R}^{2d_n\times 2d_n} \quad \theta_{ik} \in \mathbb{R}^{1\times 1} \tag{4.4.2}$$

Like Chapter 3, the parameters of the LPV system is approximated via GMM models from kinematically feasible demonstrations presented in Section 3.4.2. The advantage of this technique is that it inherently results in normalized

[1] The motion generator is fully observable.

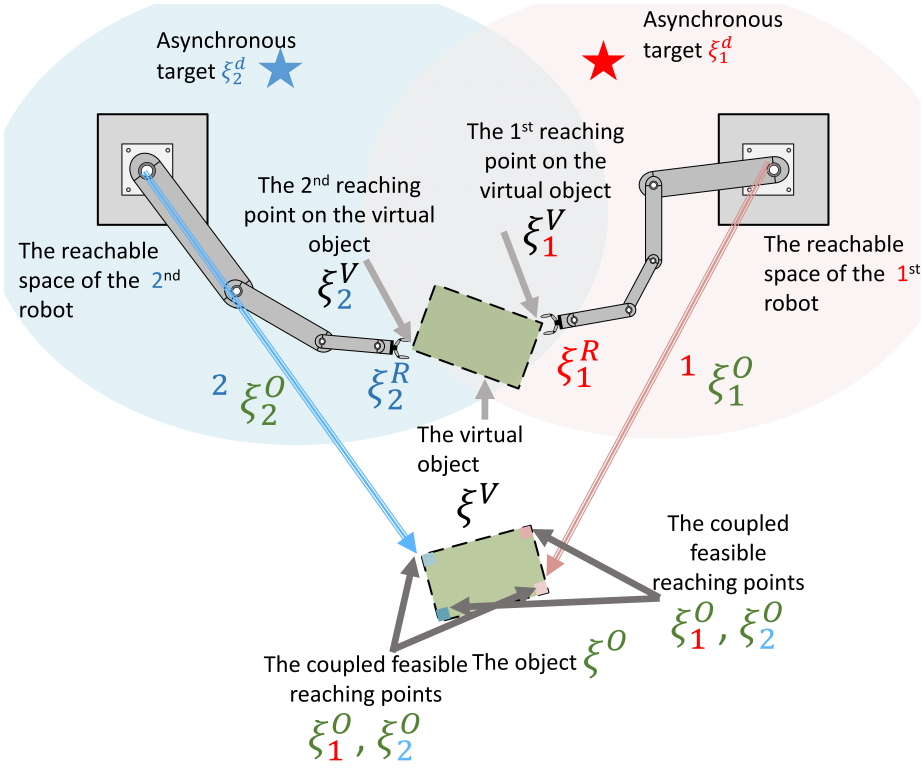


Figure 4.4: An illustration of the variables for $N_R = 2$. The reachable areas are feasible areas for grasping the object. Except for ${}^2\xi^O{}_2$ and ${}^1\xi^O{}_1$, the variables are expressed in the reference frame located on the desired intercept point; i.e., $\xi^O(T^*) = \begin{bmatrix}0 & \dots & 0\end{bmatrix}^T$.

scheduling parameters; i.e., $0 < \theta_{ik} \leq 1,\ \sum_{k=1}^{d_{s_i}} \theta_{ik} = 1,\ \ \forall i \in \{1, \dots, N_R\}$, refer to Section 3.4 for further details on this approximation approach.

Theorem 2. *The dynamical systems given by (4.4.1) asymptotically converge to $\tau_{c_i} x_i^V + (1 - \tau_{c_i}) x_i^d$; i.e*

$$\lim_{t \to \infty} \| x_i^R(t) - \tau_{c_i}(t) x_i^V(t) + (\tau_{c_i}(t) - 1) x_i^d \| = 0 \tag{4.4.3}$$

if there are $P_i^R,\ Q_i^R$ such that:

$$\begin{cases} 0 \prec P_i^R \quad 0 \prec Q_i^R \\ P_i^R A_{ik} + {A_{ik}}^T P_i^R \prec -Q_i^R \quad \forall k \in \{1, \dots, d_{s_i}\} \\ 0 \leq \theta_{ik} \end{cases} \tag{4.4.4}$$

Moreover, by taking $\dot{\tau}_{c_i}(x_i^V - x_i^d) + \tau_{c_i} \dot{x}_i^V - \mathbf{A}_i(\theta_i(x_i^R))(x_i^d + \tau_{c_i}(x_i^V - x_i^d))$ as the input and $x_i^R(t)$ as the output of the dynamical system (4.4.1), (4.4.1) is passive if (4.4.4) is satisfied.

Proof: see Appendix B.1 and B.2.

In (4.4.1), τ_{c_i} determines the level of *synchronization* between the i^{th} robot and the virtual object, see Figure 4.5. Assuming that τ_{c_i} is constant and (4.4.4) is satisfied, when $\tau_{c_i} = 0$, (4.4.1) yields an *asynchronous* DS for reaching towards individual targets:

$$\dot{x}_i^R = \underbrace{\mathbf{A}_i(\theta_i)(x_i^R - x_i^d)}_{\text{Reaching individual target } (x_i^d)} \rightarrow \Big\{ \lim_{t\to\infty} \|x_i^R - x_i^d\| = 0 \tag{4.4.5}$$

Similarly, when $\tau_{c_i} = 1$, (4.4.1) results in a perfect tracking DS of the i^{th} reaching point on the virtual object:

$$\dot{x}_i^R = \underbrace{\dot{x}_i^V + \mathbf{A}_i(\theta_i)(x_i^R - x_i^V)}_{\substack{\text{Tracking } i^{\text{th}} \text{ reaching} \\ \text{position on the virtual object}}} \rightarrow \begin{cases} \lim_{t\to\infty} \|x_i^R - x_i^V\| = 0 \\ \lim_{t\to\infty} \|\dot{x}_i^R - \dot{x}_i^V\| = 0 \end{cases} \tag{4.4.6}$$

To *smoothly transition* between these behaviors, one could calculate $\tau_{c_i}\ \forall i \in \{1, \dots, N_R\}$ with a continuous logistic function. However, we propose the following DS which varies $\tau_{c_i}\ \forall i \in \{1, \dots, N_R\}$ such that $\tau_{c_i} \rightarrow 1$ when the object moves towards the robots and $\tau_{c_i} \rightarrow 0$ when it moves away:

$$\begin{aligned} \dot{\tau}_{c_i}(t) &= \frac{\tau_{c_i}(1 - \tau_{c_i})G(\xi^O(t), \dot{\xi}^O(t))}{\mathbf{k}} \\ \tau_{c_i}(0) &= \epsilon \quad , \forall i \in \{1, \dots, N_R\} \\ G(.) &= -\frac{\dot{\xi}^O(t)^T(\xi^O(t) - \xi^O(T^*))}{\epsilon + \|\xi^O(t) - \xi^O(T^*)\|^2} \end{aligned} \tag{4.4.7}$$

Where, $0 < \epsilon << 1$ is a small positive value. $\mathbf{k} \in \mathbb{R}_{>0}$ is a positive constant that controls for the steepness of the increase or decrease of the parameter.[2] As the initial value of τ_{c_i} is positive and less than 1, (4.4.7) is a bounded dynamical system between $(0, 1)$. $M(.)$ is a function that *coordinates* the robots with the *virtual object*, such that if the real object moves toward the workspaces, the robots perform the *synchronous* behavior, otherwise they fall back to the *asynchronous* behavior. The main advantage of the proposed criterion is its adaptability. $\text{sgn}(\dot{\tau}_{c_i})$ changes with respect to the direction of the object's motion; when the object approaches the robots, $\text{sgn}(\dot{\tau}_{c_i}) \rightarrow (+)$, otherwise, $\text{sgn}(\dot{\tau}_{c_i}) \rightarrow (-)$. Consequently, if the object moves towards the robots, they are *synchronized* with the virtual object. Otherwise, they perform the *asynchronous* behavior.

To appropriately consider the effects of the *synchronization* parameters on the motion of the virtual object, and consequently of the robots, the following

[2] $0 \ll \mathbf{k}$ as having only two behaviors is desirable.

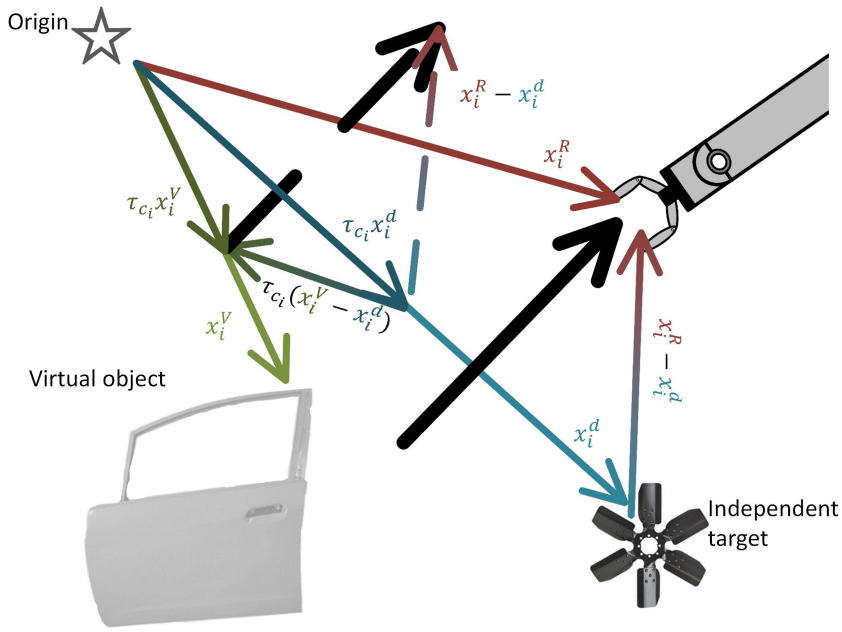


Figure 4.5: An illustration of the variables in (4.4.1). The colors of the variables and the arrows are corresponding; i.e., red represents the arm , green represents the virtual object and blue represents the independent target. The black arrows illustrates $(x_i^R - x_i^d - \tau_{c_i}(x_i^V - x_i^d))$. The dashed lines are used to show how the resultant vector is calculated.

DS is proposed to generate the motion of the virtual object.

$$\dot{x}^V(t) = \frac{1}{1 + \sum_{i=1}^{N_R} \tau_{c_i}} \left(\gamma \dot{x}^O + \dot{\gamma} x^O + A^V \left(x^V - \gamma x^O \right) + \sum_{i=1}^{N_R} U_i \right) \tag{4.4.8}$$

Where, $x^V(t) = \begin{bmatrix} \xi^V(t) & \dot{\xi}^V(t) \end{bmatrix}$ is the state of the virtual object. $0 < \gamma < 1$ is the *coordination* parameter and is of class $\mathcal{C}^1$. U_i is the interaction effect of the motion of the i^{th} end-effector on the virtual object, based on (4.4.1) and (4.4.8):

$$U_i = \dot{x}_i^R - \mathbf{A}_i(\theta_i) \left(x_i^R - x_i^d - \tau_{c_i}(x_i^V - x_i^d) \right) - \dot{\tau}_{c_i} \left(x_i^V - x_i^d \right) \tag{4.4.9}$$

By substituting, (4.4.1) and (4.4.9) into (4.4.8), we have:

$$\dot{x}^V(t) = \frac{1}{1 + \sum_{i=1}^{N_R} \tau_{c_i}} \left(\gamma \dot{x}^O + \dot{\gamma} x^O + A^V (x^V - \gamma x^O) + \sum_{i=1}^{N_R} (\tau_{c_i} \dot{x}_i^V) \right) \tag{4.4.10}$$

Theorem 3. *The dynamical system given by (4.4.10) asymptotically converges to* $\begin{bmatrix}\gamma(t)\xi^O(t) & \gamma(t)\dot{\xi}^O(t)+\dot{\gamma}(t)\xi^O(t)\end{bmatrix}^T$ *i.e.*

$$\begin{aligned}&\lim_{t\to\infty}\|\xi^V(t)-\gamma(t)\xi^O(t)\|=0\\&\lim_{t\to\infty}\|\dot{\xi}^V(t)-(\gamma(t)\dot{\xi}^O(t)+\dot{\gamma}(t)\xi^O(t))\|=0\end{aligned}\tag{4.4.11}$$

if there are P^V, Q^V *such that:*

$$\begin{cases}0\prec P^V\\0\prec Q^V\\P^VA^V+A^{V^T}P^V\prec -Q^V\end{cases}\tag{4.4.12}$$

Moreover, by taking $\gamma\dot{x}^O+\dot{\gamma}x^O-A^V\gamma x^O$ *as the input and* x^V *as the output of the dynamical system (4.4.10), (4.4.10) is passive if (4.4.12) is satisfied.*

Proof: see Appendix B.3, B.4.

Remark 1. *if* $\tau_{c_i}=1$ *, based on (4.4.6) and (4.4.11), the robots asymptotically converge to the reaching points on the object; i.e:*

$$\begin{aligned}&\lim_{t\to\infty}\|\xi_i^R(t)-\gamma(t)\xi_i^O(t)\|=0\\&\lim_{t\to\infty}\|\dot{\xi}_i^R(t)-(\gamma(t)\dot{\xi}_i^O(t)+\dot{\gamma}_i(t)\xi_i^O(t))\|=0\end{aligned}\qquad \text{if } \tau_{c_i}=1\tag{4.4.13}$$

If $\tau_{c_i}=1$ and $(\gamma(t)=\dot{\gamma}(t)=0)$, (4.4.10) generates asymptotically stable motions towards the predicted intercept point: i.e., coordination between the robots is preserved, but the coordination between the robots and the object is lost. If $\tau_{c_i}=1$ and $(\gamma(t)=1,\dot{\gamma}(t)=0)$, (4.4.10) generates asymptotically stable motions towards the real object, even though its motion is not accurately predicted: i.e., perfect *coordination* with the object.[3] However, in this case, there is no guarantee that the virtual object intercepts the real object inside the workspace of the robots; i.e., because of the robots' kinematics constraints, they can not comply with the object's motion any more. Thus, one can vary the values of the coordination parameter between $[0,1]$, such that $\gamma=1$ at the vicinity of the desired intercept time as proposed in (Mirrazavi Salehian et al., 2016b):

$$\dot{\gamma}=\frac{1-\gamma}{\|\xi^O(t)-\xi^O(T^*)\|+\varepsilon}=\frac{1-\gamma}{\|\xi^O(t)\|+\varepsilon},\qquad \gamma(0)=0\tag{4.4.14}$$

. (4.4.14) improves the robustness of the multi-arm reaching motion in face of

[3] We assume that the dynamical system (4.4.10) is fast enough to converge to an acceptable neighborhood around the desired trajectory $\gamma\begin{bmatrix}\xi^O(t) & \dot{\xi}^O(t)\end{bmatrix}^T$ before T^*; i.e., $\|\xi^V(T^*)-\gamma\xi^O(T^*)\|\leq\varepsilon$

inaccuracies in the object's motion prediction, as it ensures that when the object is close enough to the *feasible reaching positions*, the virtual object converges to the real object and perfectly tracks it; i.e., $\gamma(T^*) = 1$. Hence, the robots can simultaneously track the desired reaching points on the object in a coordinated manner. The proposed algorithm can only guarantee collision-avoidance between end-effectors, via the *virtual object*, for *synchronous* behaviors. In the following section we present a centralized inverse kinematics solver, that addresses self-collision avoidance at all times.

4.5 Centralized Inverse Kinematic Solver

To avoid collisions between the joints of the arms, the IK solver must consider not only the kinematic constraints of each robot, but also self-collision constraints. Given that the robots' bases are fixed wrt. each other, we can explore the joint workspace of the robots, in order to model the regions that may lead to collision. Since the space of joint configurations is continuous, the regions of collisions can be approximated by building a continuous map of the feasible (safe) and infeasible (collided) configurations. Assuming that the infeasible regions can be bounded through a continuous and continuously differentiable function $\Gamma(q^{ij}) : \mathbb{R}^{d_{q_i}+d_{q_j}} \rightarrow \mathbb{R}$, where $q^{ij} = [q^i, q^j]^T \in \mathbb{R}^{d_{q_i}+d_{q_j}}$ are the joint angles of the i^{th} and j^{th} robots, respectively. We define $\Gamma(x)$ such that:

$$\begin{aligned} &\text{Collided configurations:} && \Gamma(q^{ij}) < 1 \\ &\text{Boundary configurations:} && \Gamma(q^{ij}) = 1 \\ &\text{Free configurations:} && \Gamma(q^{ij}) > 1 \end{aligned} \tag{4.5.1}$$

A data-driven approach for building $\Gamma(q^{ij})$ is proposed in (Mirrazavi Salehian et al., 2017a). (4.5.1) provides constraints that must be taken into account when solving the inverse kinematics (IK) problem. We propose the following quadratic program to solve the IK:

$$\underbrace{\underset{\dot{\mathbf{q}}}{\text{argmin}}\ \frac{\dot{\mathbf{q}}^T W \dot{\mathbf{q}}}{2}}_{\text{Minimize expenditure}} \tag{4.5.2a}$$

Subject to:

$$\underbrace{\mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \dot{\boldsymbol{\xi}}^R}_{\text{Satisfy the desired end-effector motion}} \tag{4.5.2b}$$

$$\underbrace{\dot{\boldsymbol{\theta}}^{-} \leq \dot{\mathbf{q}} \leq \dot{\boldsymbol{\theta}}^{+}}_{\text{Satisfy the kinematic constraints}} \tag{4.5.2c}$$

$$\begin{aligned} &-\nabla\Gamma^{ij}(q^{ij})^{T}\dot{q}^{ij} \leq \log(\Gamma^{ij}(q^{ij})-1) \\ &\underbrace{\forall(i,j) \in \{(1,2),(1,3),\ldots,(N_R-1,N_R)\}}_{\text{Do not penetrate the collision boundary}} \end{aligned} \tag{4.5.2d}$$

Where, $\mathbf{q} = [q^1,\ldots,q^{N_R}]^T \in \mathbb{R}^{d_\mathbf{q}}$, $d_\mathbf{q} = \sum_{i=1}^{N_R} d_{q_i}$.[4] W is a block diagonal matrix of positive definite matrices. $\mathbf{J} = diag(J_1,\ldots,J_{N_R})$ is the block diagonal matrix of the Jacobian matrices. $\dot{\boldsymbol{\xi}}^R = \begin{bmatrix}\dot{\xi}_1^R & \ldots & \dot{\xi}_{N_R}^R\end{bmatrix}^T \in \mathbb{R}^{d_\mathbf{n}}$, $d_\mathbf{n} = N_R d_n$ is the desired velocity given by (4.4.1). $\dot{\boldsymbol{\theta}}^i = \begin{bmatrix}\dot{\theta}_1^i & \ldots & \dot{\theta}_{N_R}^i\end{bmatrix}$ $\forall i \in \{-,+\}$ and $\dot{\theta}_i^+ \in \mathbb{R}^m$ and $\dot{\theta}_i^- \in \mathbb{R}^m$ are conservative lower and upper bounds of the joint limits, respectively. To integrate the joint limits into the velocity level constraints, (Xia and Wang, 2000) propose the following equation.

$$\begin{aligned} \dot{\theta}_i^- &:= \max\left(\mu(q_i^- - q_i), \dot{q}_i^-\right) \\ \dot{\theta}_i^+ &:= \min\left(\mu(q_i^+ - q_i), \dot{q}_i^+\right) \end{aligned} \tag{4.5.3}$$

With q_i^-, q_i^+, q_i^-, $\dot{q}^-$ as the conservative lower and upper bounds on the joints' positions and velocities. The *intensify coefficients*, $0 < \mu_p$, determine the magnitude of decelerations and are defined such that the feasible region of $\dot{\theta}$ generated by the joint angle limits' projections is not smaller than the real one made by the joint velocity limits. Hence, (Xia and Wang, 2000) proposes to select them not smaller than $\max\{\frac{\dot{q}_i^+ - \dot{q}_i^-}{q_i^+ - q_i^-}\}$.

While the robots are far from the boundary configurations, the value of $\log(\Gamma^{ij}(q^{ij}) - 1)$ is positive which relaxes the inequality constraints; i.e., the robots accurately follow the desired end-effector trajectory. When they are near the boundary configurations, the value of $\log(\Gamma^{ij}(q^{ij})-1)$ is negative. Therefore, constraint (4.5.2d) forces the joint angles to move away from the boundary as they approach it. Since satisfying the collision avoidance and the kinematic constraints is of higher priority than following the desired end-effector motion, in practice, we give higher penalty to (4.5.2c) and (4.5.2d), than to (4.5.2b). In a particular case, when the robots are initiated inside of the boundary (i.e., $\Gamma^{ij}(.) < 1$), $\log(\Gamma^{ij}(.) - 1)$ is not defined. In this case, we replace $\log(\Gamma(.) - 1)$

[4] For example, $N_R = 2$, $\mathbf{q} = q^{12}$. Based on the experience, due to hardware limitations, it is not possible to construct a data set of collision boundaries for more than two 7-DOF KUKA arms at once; e.g., for 3 arms, the size of the data set approximately is $(3*7*3)\times 1000^3$ while it is $(3*7*2)\times 1000^2$ for 2 arms.

with a large negative number which pushes the robots outward towards the boundary.

Equation (4.5.2) is a convex quadratic programming (QP) problem with equality and inequality constraints, hence, there is no closed form solution for it. As the solutions to such linear optimization are solver-dependent, in terms of computation cost, we compare three approaches to solve (4.5.2). The *first* approach formulates (4.5.2) as a system of piecewise-linear equations and uses a DS-based approach to solve them (Xia and Wang, 2000; Zhang et al., 2004; Zhang, 2005). The *second* approach uses Nlopt, a standard non-linear programming solver (Johnson, 2015). The *third* approach uses a solver specifically designed for constrained convex problems. The solver, which is called CVXGEN and introduced in (Mattingley and Boyd, 2012), generates C codes, tailored for the specific formulation of (4.5.2). As the *second* and *third* approaches are ready to use interfaces, in the rest of this chapter we introduce the *first* approach.[5]

Lemma 1. *Linear quadratic programming* (4.5.2) *is equivalent to the following system of piecewise-linear equations.*

$$P_\Omega(u - (Mu + b)) - u = 0 \tag{4.5.4}$$

Moreover, the following dynamical system is asymptotically stable to u^**, where* $u^* \in \mathbb{R}^{d_u}$ *is the solution of* (4.5.4) *.*

$$\dot{u} = (I + M^T)(P_\Omega(u - (Mu + b)) - u) \tag{4.5.5}$$

where

$$M = \begin{bmatrix} W & -\boldsymbol{J}(\boldsymbol{q})^T & -\nabla\Gamma(\boldsymbol{q}) \\ \boldsymbol{J}(\boldsymbol{q}) & 0 & 0 \\ \nabla\Gamma(\boldsymbol{q})^T & 0 & 0 \end{bmatrix} \tag{4.5.6a}$$

$$b = \begin{bmatrix} 0 \\ -\dot{\xi}^R \\ -\log(\Gamma(\boldsymbol{q}) - 1) \end{bmatrix} \tag{4.5.6b}$$

. $u = \begin{bmatrix} \dot{\boldsymbol{q}} & \eta & \nu \end{bmatrix}^T \in \mathbb{R}^{d_u}$*;* $d_u = d_{\mathbf{q}} + d_{\mathbf{n}} + 1$*. As* $0 \preceq W$*,* M *is also positive semi-definite.* $\eta \in \mathbb{R}^{d_N}$ *and* $\nu \in \mathbb{R}$ *are the dual decision vectors.* $P_\Omega(h) =$

[5]The formulation of the Lyapunov stability proof is inspired from (Xia and Wang, 2000; Zhang et al., 2004; Zhang, 2005).

$\begin{bmatrix} P_\Omega(h_1) & \dots & P_\Omega(h_{d_u}) \end{bmatrix}$ *is the element-wise* $\Omega-$ *projection operator defined as*

$$P_\Omega(h_i) = \begin{cases} u_i^- & h_i < u_i^- \\ h_i & u_i^- \leq h_i \leq u_i^+ \\ u_i^+ & u_i^+ < h_i \end{cases} \quad \forall i \in \{1, \dots, d_u\} \tag{4.5.7}$$

. u^+ *and* u^- *are the bounds of the primal-dual decision vector* u *defined as*

$$u^- = \begin{bmatrix} \dot{\boldsymbol{\theta}}^- \\ -\infty \\ 0 \end{bmatrix} \qquad u^+ = \begin{bmatrix} \dot{\boldsymbol{\theta}}^+ \\ +\infty \\ +\infty \end{bmatrix} \tag{4.5.8}$$

and $\Omega = \{u \in \mathbb{R}^{d_u} | u^- \leq u \leq u^+\}$.

Proof: Refer to (Zhang, 2005) and (Xia and Wang, 2000).

Theorem 4. *By taking* u *and* $(I + M^T)P_\Omega(u - (Mu + b))$ *as the output and the input of the system* (4.5.5)*, respectively,* (4.5.5) *is passive.*

Proof: See Appendix B.5.

Remark 2. *Theorems 2, 3 and 4 show that all the proposed dynamical systems are passive. Hence, if the robots and the low level torque controllers are passive, the proposed framework for coordinated Multi-Arm system is passive and stable as it is a feedback system of passive elements.*

4.6 Empirical Validation

The performance of the proposed framework is evaluated on two different real dual-arm platforms. On the first platform, the compliance (in the form of coordination) between the arms and the object is evaluated. The second experimental set up is designed to evaluate the performance of *dual-behavior* and the *self-collision* avoidance. The top view of the set-up is illustrated in Figure 4.6.

4.6.1 First experimental set-up

The proposed framework is implemented on a real dual-arm platform, consisting of two 7 DOF robotic arms, namely a KUKA LWR 4+ and a KUKA IIWA mounted with a 4 DOF Barrett hand and a 16 DOF Allegro hand. The distance between the base of the two robots is $\begin{bmatrix} 0.25 & 1.5 & -0.1 \end{bmatrix}^T$ m. As the synchronized behavior is the only desired behavior in this section, the value of

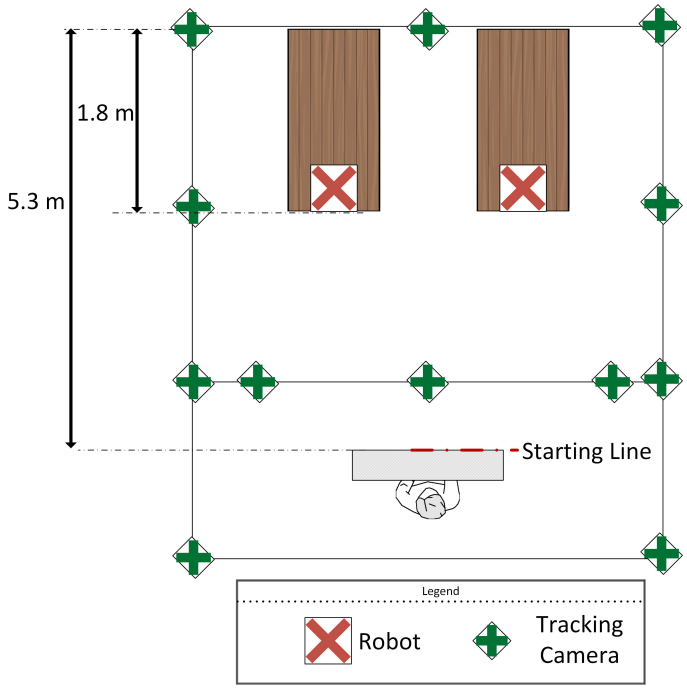


Figure 4.6: Top view illustration of the set-up. The motion of the object is recorded by 12 cameras. The distance between the arms is changed based on the size of the object. More information is available in Table 4.2.

synchronization parameters in (4.4.1) are manually set to one. Moreover, given the distance between the robots, there is a maximum of 30cm intersection between the robots' workspaces; i.e., the arms are ensured to not collide. Hence, instead of centralized IK solver presented in Section 4.5, independent velocity based control method without joint velocity integration IK solver (Nakanishi et al., 2005a) is used to convert the desired end-effector motion into the 7-DOF joints state. In order to avoid high torques, the resulting joint angles are filtered by a critically damped filter. The robot is controlled at a rate of 500 Hz. The fingers are controlled with joint position controllers. All the hardware involved (e.g., arms and hands) are connected to and controlled by one 3.4-GHz i7 PC. The position of the *feasible reaching points* of the objects are captured by an Optitrack motion capture system from Natural point at 240 Hz. Since the control loop is faster than the motion capture system, the predicted position of the object is used as the object position in (4.4.8), when the current position of the object is not available.

Our empirical validation is divided into three parts that demonstrate the controller's ability: (i) compliance within the multi-arm systems; (ii) compliance between the two arms' motions and a large moving object so as to reach and grab it, while introducing unpredictability in the object's motion (by having the object be carried by a blindfolded human) and (iii) rapidly adapt bi-manual motions to intercept a flying object, without using a pre-defined model of the

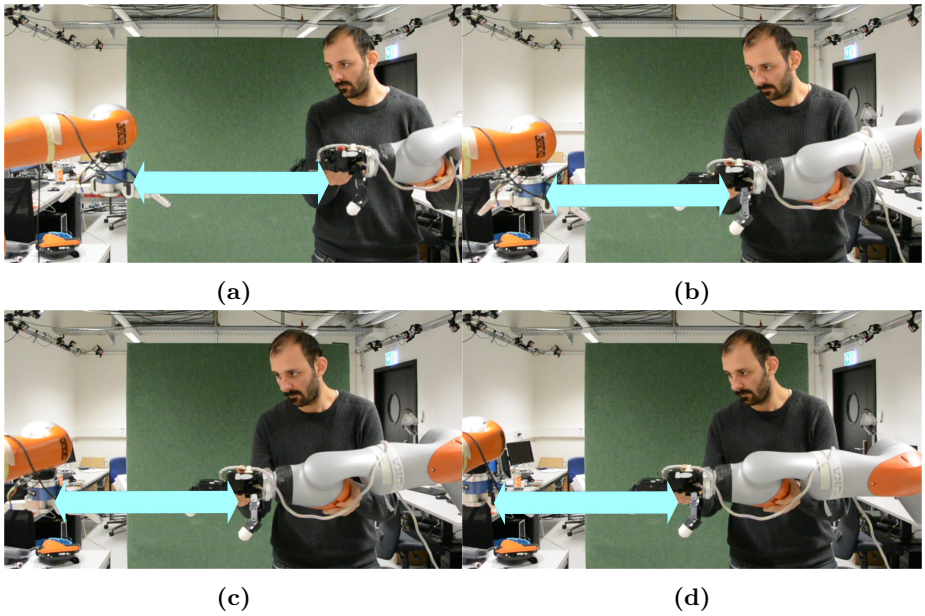


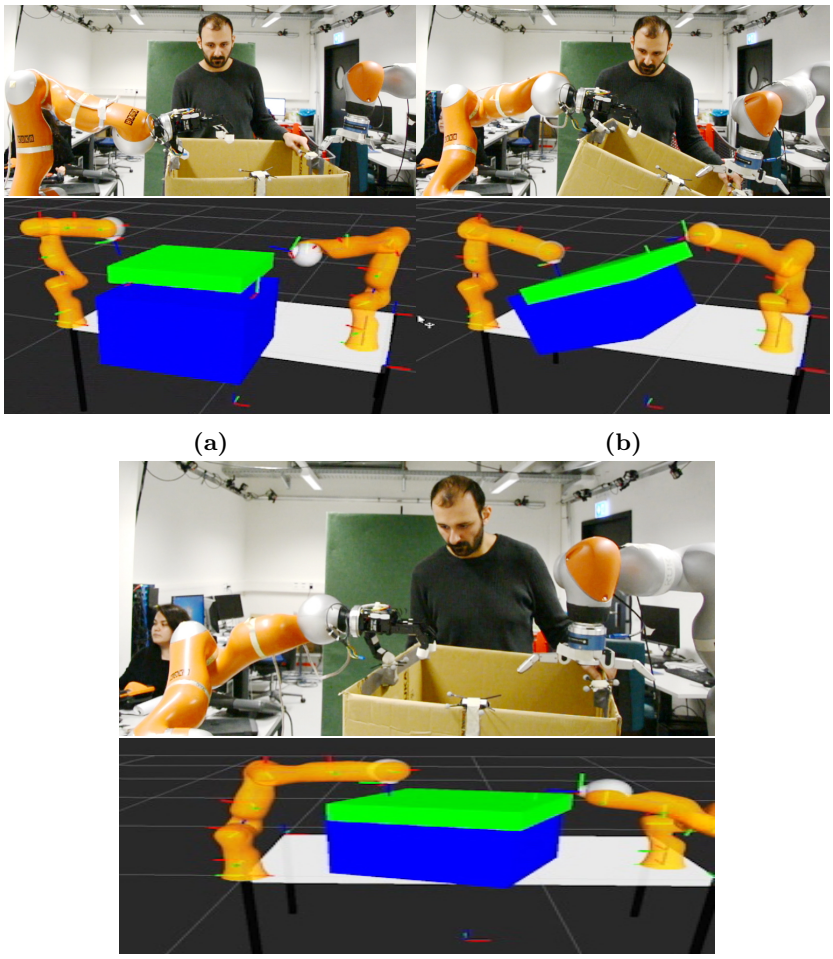
Figure 4.7: Snapshots of the video illustrating coordination of the arms in free space. Both arms are manually assigned to the synchronized behavior and as the object is outside the workspace of the robots, the coordination parameter γ is close to 0. The human operator perturbs one of the arms, which leads the other arm to move in synchrony following the motion of the virtual object attached to the two end-effectors.

object's dynamics. A corresponding video is available on-line: https://youtu.be/UfucwRGa7k8

4.6.1.1 COMPLIANCE CAPABILITIES

The first scenario is designed to illustrate the compliance capabilities of the arms with each other and with the object. We initially show arm-to-arm compliance capabilities by keeping the real object outside of the workspace of the robots, this will force the coordination parameter γ to be 0, favoring arm-to-arm coordination. As the human operator perturbs one of the robot arms, the virtual object is perturbed as well, resulting in a stable synchronous motion of the other unperturbed arm (Figure 4.7). Since we offer a centralized controller based on the virtual object's motion, there is no master/slave arm; thus, when any of the robots are perturbed, the others will synchronize their motions accordingly.

We then present the compliance on the second sub-level; i.e., between the arms and the object which moves inside the workspace of the robots. The object is a large box ($60 \times 60 \times 40cm$) held by a human operator. Two points on the edges of the box are specified as the *feasible reaching points*. When the box is inside the joint workspace of the robots, the operator changes the orientation and the position of the box to show the compliance capabilities between the robots and the object (Figure 4.8). Due to (4.4.8), the motion of the arms compliantly track the object and successfully comply with the motions of others.



(a) **(b)**

(c)

Figure 4.8: Snapshots of the coordination capabilities between the arms and a moving/rotating object. The real object is inside the workspace of the robots; hence, the coordination parameter γ is close to 1 and the arms-to-object coordination is favored. The bottom figures show the real-time visualization of the robots, and the virtual (green) and real object (blue).

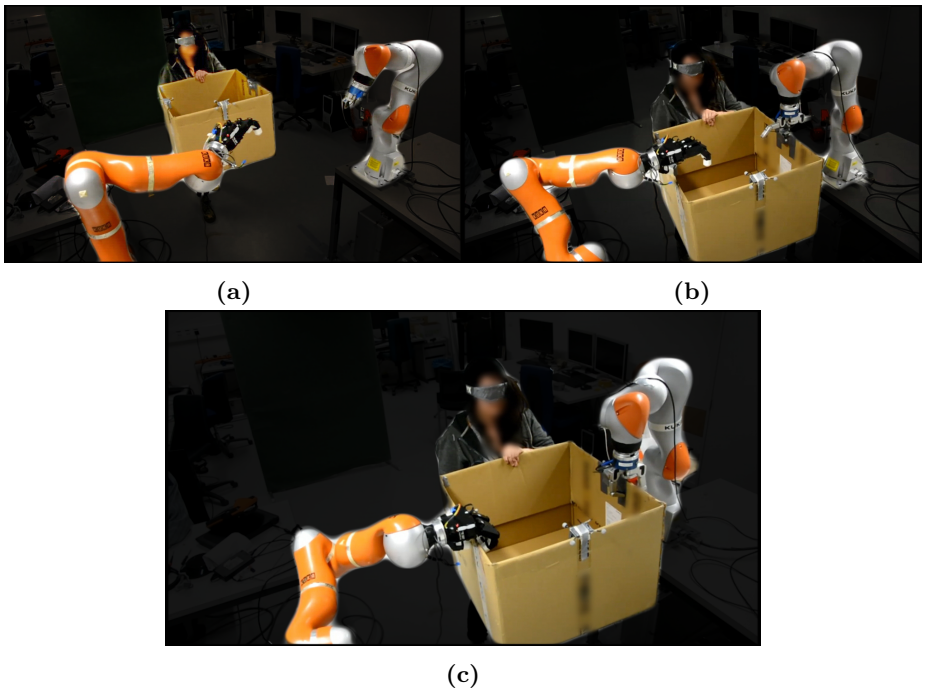
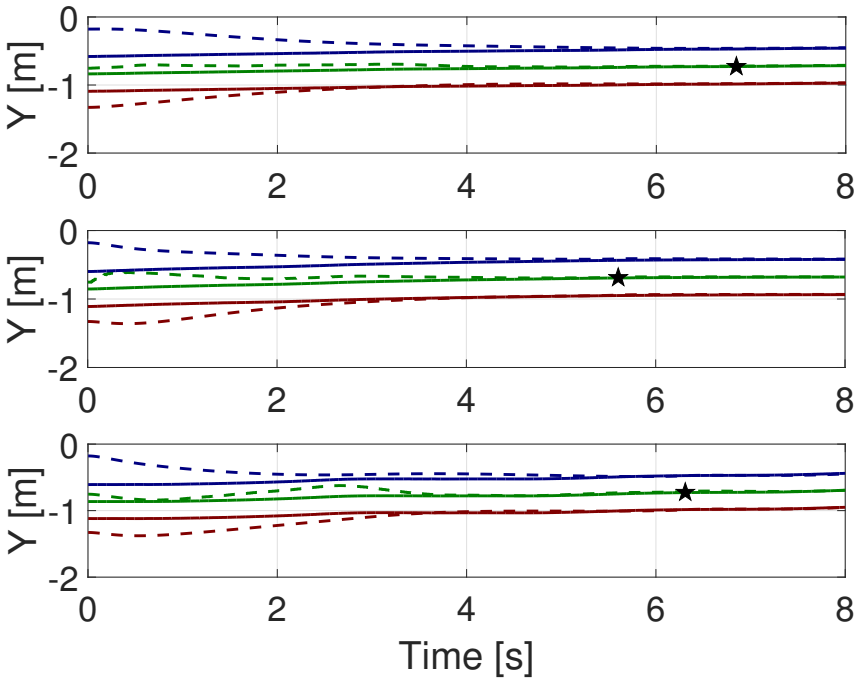


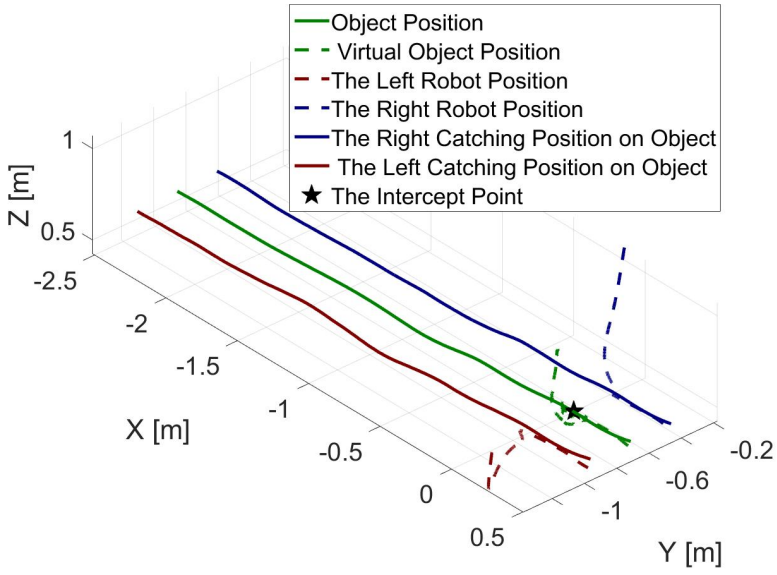
Figure 4.9: Snapshots of the robots' motion when reaching for a moving object, carried by a blindfolded operator. (a) Onset of object trajectory's prediction. (c) Arms have intercepted the object and the fingers have closed on the object.

4.6.1.2 Reaching to Grab a Large Moving Object

In this second scenario, we use the same object as before. Yet, now a blindfolded operator holds the box while walking towards the robots. Once the end-effectors are less than 2 cm away from the *feasible reaching points*, finger closures of the hands are triggered and the box is successfully grabbed from the human. As can be seen in Figure 4.9, the operator is blindfolded to achieve *unpredictable* trajectories and avoid the natural reactions of the humans to help the robots. When the human operator carrying the box is approaching the robots, the virtual object converges to the box and follows it until the desired interception points are reached. The fingers close and the box is grabbed from the human. An example of the desired robot trajectory and the box trajectory are shown in Figure 4.10. As expected, the end-effectors converge to the box and continue to track its motion. The initial value of γ in (4.4.8) is set to 0. While the box is approaching the robots, γ starts increasing and finally reaches $\gamma = 1$ when the object is in the workspace of the robots. Hence, (4.4.8) generates asymptotically stable motions towards the real object instead of the intercept point. Consequently, the prediction of the intercept point does not play a vital role in grabbing the box.



(a)



(b)

Figure 4.10: Different examples of the position of the end-effectors generated by the dynamical system (4.4.8). Only the trajectories along y axis is presented. The illustrated object trajectory is the predicted trajectory of the uncaught object. The prediction of the box's trajectory requires some data to be initialized and uses almost all of the first 0.2 meter of the object in x. As expected, the outputs of (4.4.8) first converges to desired intercept position, since γ is a small value, then it softly intercepts the object's trajectory and follow the object's motion. The robots are stopped if the object is not moving or the fingers are closed.

4.6.1.3 Reaching for Fast Flying Objects

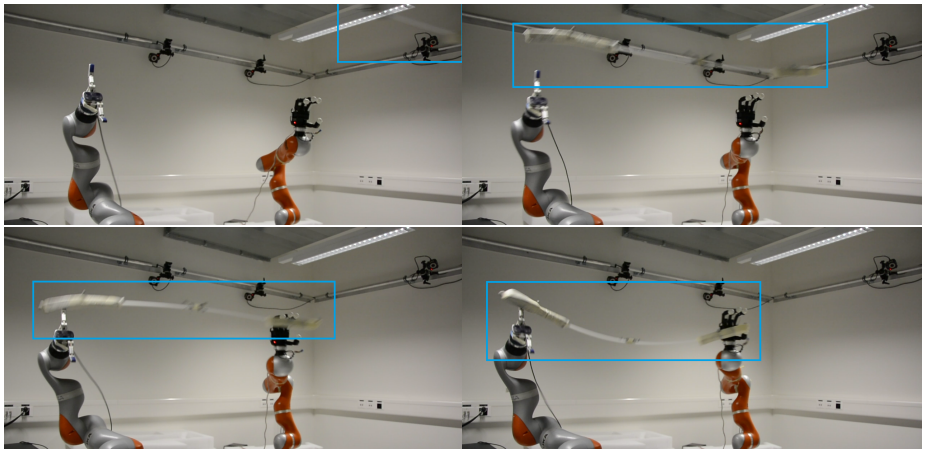
The third scenario is designed to show the capability of the proposed controller in generating fast motions, where a rod ($150 \times 1cm$) is thrown to the robots from $2.5m$ away, resulting in approx. $0.56s$ flying time. The distance between the base of the two robots is reduced to $\begin{bmatrix}0.25 & 1.26 & -0.1\end{bmatrix}^T m$. The first $0.4m$ of the object trajectory initializes the trajectory prediction algorithm. Due to inaccurate prediction of the object trajectory, the feasible intercept points need to be updated and redefined during the reach. The new feasible intercept point is chosen in the vicinity of the previous one to minimize the convergence time. As the motion of the object is fast and the predicted reaching points are not accurate, the initial values of γ in (4.4.14) is set to 0.5. This decreases the convergence duration of the robots to the real object. Snapshots of the real robot experiments are shown in Figure 4.11. Visual inspection of the data and video confirmed that the robots coordinately follow the motion of the object and intercept it at the vicinity of the predicted feasible intercept point.

4.6.1.4 Systematic Assessment

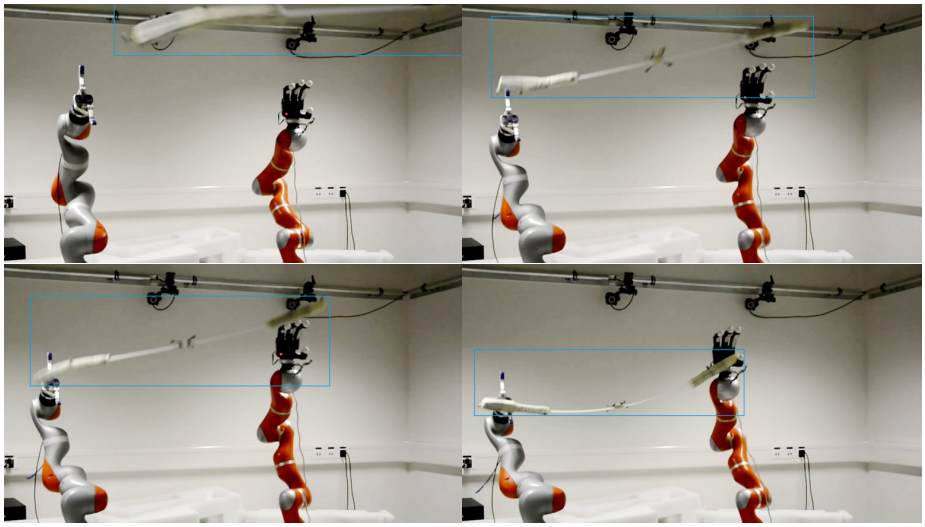
The success rates of our experiments are measured by defining a Boolean metric; i.e., success or failure. A trial is classified as a success if the robot intercepts the object at the desired point within less than $2cm$ error. The success rate is 85% in the box scenario and 37.5% in the rod scenario. Failures are due to either inaccuracies in the measurement of the object's state or the IK solver. To track the object, all the markers must be visible to the cameras. In the box scenario, the object's tracking was obscured partly when the object was covered by the robotic arms or the operator. In the rod scenario, the vision system looses track of the markers approx. 55% of the time. This happens, for example, when the rod rotates rapidly.

Failures caused by the IK solver are mostly observed in the flying rod scenario, where the inability to accurately calculate joint-level motions corresponding to the desired end-effectors' trajectories results in errors in the robots' motions. In over 40 trials, the tracking error between the desired and the actual end-effector position at the intercept point is approximately $1.42 \pm 1.92cm$. The large variance in the error indicates the implemented IK Solvers sensitivity to the robot's joints configuration.

To systematically assess the robustness of the algorithm to unmeasured object positions, a set of simulations was designed to reach for a moving ball. The ball diameter is $1.2m$. The simulation is repeated 130 times in total for two different object velocities; i.e., fast and slow motions. To assess the effects of the unmeasured object positions on the interception error, the desired and the real end-effectors' states are assumed equal. Results from this evaluation indicate that the interception error is directly correlated to the percentage of unmeasured object points and consequently, the velocity of the moving object



(a) The object flight duration is $0.39s$.



(b) The object flight duration is $0.48s$.

Figure 4.11: Snapshots of the arms reaching for a fast moving object. The object is specified by a blue square. The arms move in **same** direction (a) or in **opposite** directions (b) to keep the coordination between the arms and with the object. In order to not damage the robot's hands, the robot hands do not close on the object when the hands intercept the object.

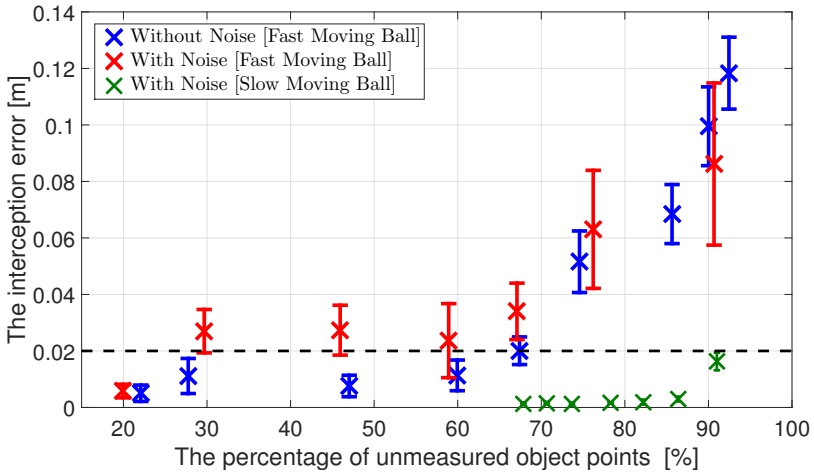


Figure 4.12: The interception error is the average of the minimum distance between the ball and the end-effectors. The throwing positions are randomly chosen within the range of $\begin{bmatrix}-3.5 \pm 0.1 & -1.0 \pm 0.1 & -0.0 \pm 0.1\end{bmatrix}^T m$. The initial ball speeds for fast and slow motions are randomly chosen within range of $8.94 \pm 0.173 \frac{m}{s}$ and $1.63 \pm 0.173 \frac{m}{s}$, respectively. We only consider trials when the ball passes through the robots workspaces. The measurement noise is simulated with pseudo-random values within the range of ± 0.02m. The cut-off success/failure assessment is illustrated by the back dashed line.

(see Figure 4.12). Thus, the faster the object, the more sensitive the system is to the tracking inaccuracies.

4.6.2 Second experimental set-up

The proposed framework is implemented on a dual-arm platform, consisting of two 7 DOF KUKA IIWA robotic arms mounted with a 2 finger Robotiq gripper and a 16 DOF Allegro hand. The robots are controlled via Fast Research Interface (FRI) at the joint impedance mode. The fingers are controlled with joint angle position controllers in two states: Open, Close. All the hardware involved (e.g., arms and hands) are controlled by one 3.4-GHz i7 PC.[6] The position of the *feasible reaching points* of the objects are captured by an Optitrack motion capture system from Natural point at 240 Hz. As the outputs of the vision system are noisy, a Savitzky-Golay filter is used to smooth the position of the object and estimate velocity and acceleration from these position measurements.

The empirical validation is divided into two parts which demonstrate the controller's capabilities in estimation of the desired behavior and accordingly adaptation of the two arms' motions; i.e., move *asynchronously* or *synchronously*. A corresponding video is available on-line: https://youtu.be/LxAWvU2locU.

Moreover, the performance of the framework is systematically assessed in three different levels to show: (i) the success rate of coordinately reaching for a

[6] Due to implementation constraints, one of the arms is connect to another PC. Apart from this connection, no computation is done on the other PC.

moving object, (ii) performance of IK solvers and (iii) sensitivity of the framework to noise. A corresponding video of the systematically assessment is available on-line at https://youtu.be/S5fvr_wZ_W0.

4.6.2.1 Dual Behavior Capabilities.

The first scenario is designed to illustrate the dual-behavior capabilities of the arms. The asynchronous behavior of each robot is to reach a fixed target (see Figure 4.13) or follow the hands of operator 1 who stands between the arms (see Figure 4.14). The synchronized behavior is to coordinately reach an object brought by an operator. The visual inspection of the data shows that when the operator moves the object toward the robots, based on (4.4.7), the value of τ_{c_i} $\forall i \in \{1, 2\}$ smoothly increases to one. Hence, a smooth transition from the *unsynchronized* behavior to the *synchronized* behavior is achieved; see Figure 4.13 and Figure 4.14. As there is no full coordination between the arms while the value of τ_{c_i} $\forall i \in \{1, 2\}$ is less than one, perturbing one arm does not affect the motion of other arm, see Figure 4.14(a),(b),(c),(d). While the arms are allocated to the synchronized behavior, due to (4.4.14), the arms successfully comply with the each other's and the object's motions and intercept the object at the desired points, see Figure 4.14(g),(h).

4.6.2.2 Systematic assessment

Coordination and Adaptation Assessment.

To systematically assess the performance of the proposed framework. We design a handover scenario, where an operator holds an object and moves toward the arms and hand overs the object to the robots. The robots' hands are triggered to close when the distance between the arms and the object is less than $1cm$. The success rates of our experiments are measured by defining a Boolean metric; i.e., success or failure. A trial is classified as a success if the robot intercepts the object at the desired point within less than $1cm$ error. Three car parts are chosen with different sizes and shapes to validate the algorithm, i.e., a bumper, a fender and a front panel, see Figure 4.15. The reaching orientation for each robot is specified by the operator with respect to the orientation of the hands. The experiments were repeated 20 times for each object. The objects were moved and handed-over by the operator from random initial positions with different orientations. The snapshots of the experiments are shown in Figure 4.16 and an example of the motion of the arms and the object is shown in Figure 4.17. The variation of the intercept points are illustrated in Figure 4.18. Data of the experimental results are summarized in Table 4.2. In this table, the first point is the position the object when, for the first time, the *feasible* intercept point is determined. Motion duration is the duration of the objects'

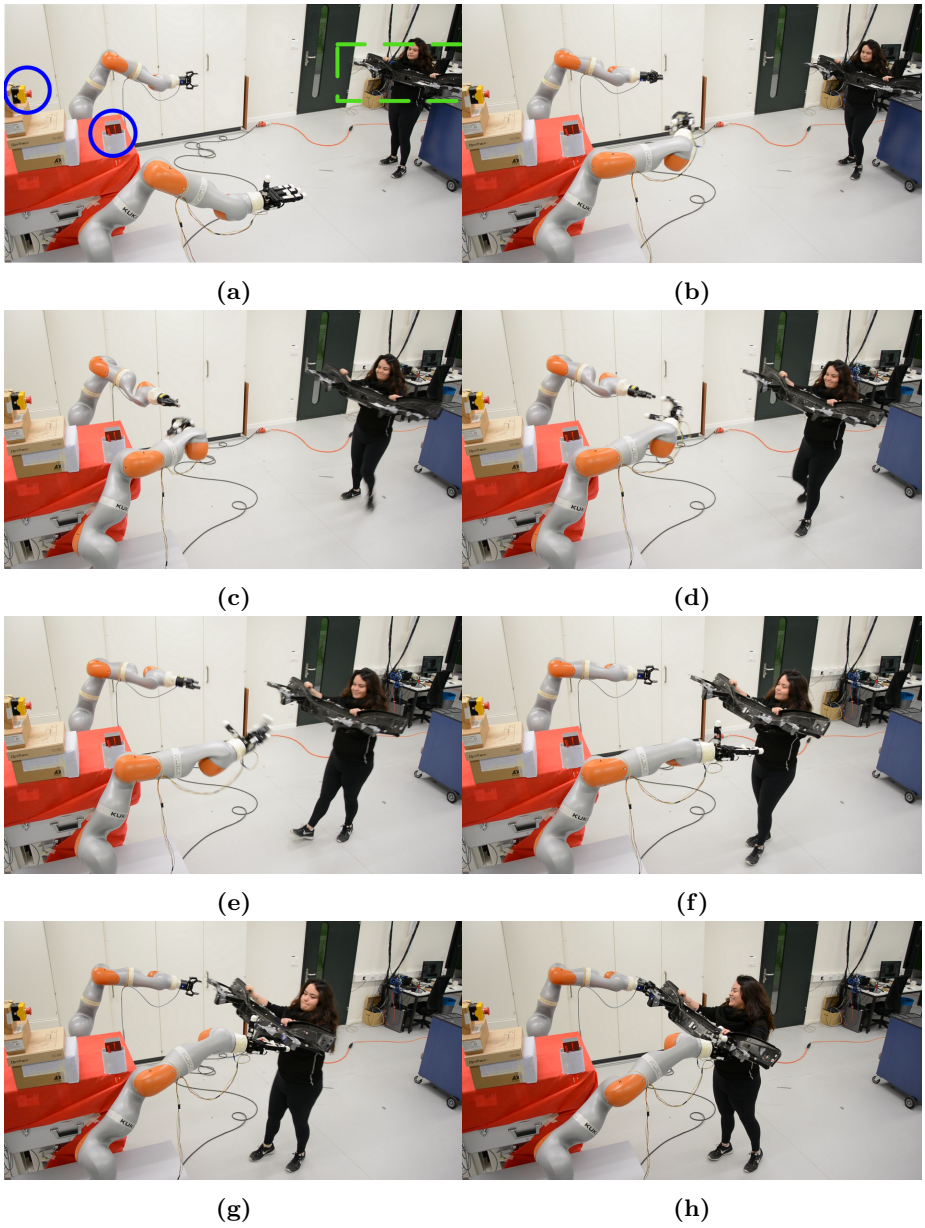
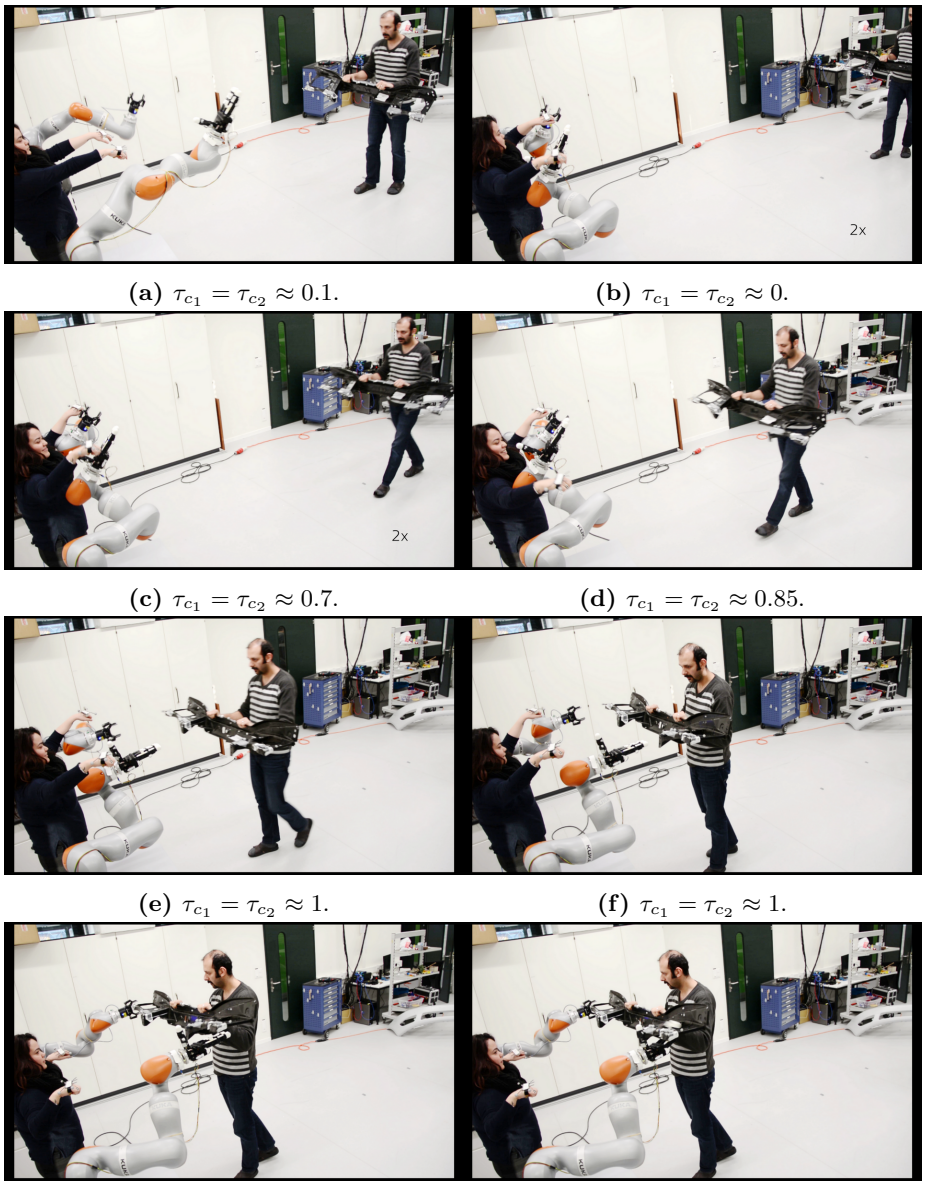


Figure 4.13: Snapshots of the video illustrating the dual behavior capabilities. The target of synchronous and asynchronous behaviors are highlighted in (a) by the green square and the blue circles, respectively. Initially, the robots are allocated to the asynchronous behavior. Hence, the robots move toward the asynchronous targets in (b) and (c). In (d), $\gamma_{c_i} \approx 1 \quad \forall i \in \{1, 2\}$ as the operator moves the object toward the arms; i.e., the synchronous behavior. Consequently, the robots comply with the object's motion and simultaneously reach and intercept it at the desired reaching points.



(a) $\tau_{c_1} = \tau_{c_2} \approx 0.1$. **(b)** $\tau_{c_1} = \tau_{c_2} \approx 0$.

(c) $\tau_{c_1} = \tau_{c_2} \approx 0.7$. **(d)** $\tau_{c_1} = \tau_{c_2} \approx 0.85$.

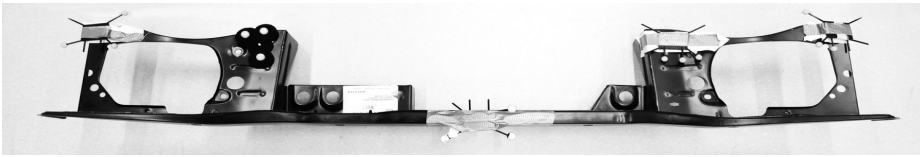
(e) $\tau_{c_1} = \tau_{c_2} \approx 1$. **(f)** $\tau_{c_1} = \tau_{c_2} \approx 1$.

(g) $\tau_{c_1} = \tau_{c_2} \approx 1$. **(h)** $\tau_{c_1} = \tau_{c_2} \approx 1$.

Figure 4.14: Snapshots of the video illustrating of dual behavior capabilities. The asynchronous behavior is to follow the hands of the operator 1 who is inside of the robot workspaces. When the operator 2 moves the object away from or toward the arms, the synchronization parameter smoothly goes to 0 and 1, respectively.

(a) The bumper.

(b) The fender.



(c) The front panel.

Figure 4.15: The car parts which are used for the systematic assessment of the framework.

motions from the first point till the intercept position. The bases distances is the distance between the robots' bases.

The overall success rate of the experiment is 86.7%[7]. Failures are mostly due to inaccuracies in the measurement of the object's state. To find the position of the reaching areas and the orientation of the object, all the five markers must be visible to the cameras. The objects' tracking markers were occluded partly when the objects were covered by the robotic arms or the operator. In 5 out of 8 cases, one or two out of the five markers were not detected accurately when the object was close to the robots, hence either the robots converged to a wrong position or the synchronization parameters were changing undesirably. These two cases can be detected easily. In the first case, the hands are closed where there is no object. In the second case, the robots rapidly move back and forth; i.e., chattering between *synchronous* and *asynchronous* behaviors. In 2 out of 8 cases, the robots started moving very late as the object's predicted motion was completely wrong. In this case, the object was inside of the robots' workspaces when the robots started moving. As the motions of the objects was not extremely fast, the IK solver was able to accurately generate the joint space trajectory and only in 1 out of 8 cases it failed to track the desired end-effectors' motions. In this case, the operator suddenly changed the object's orientation when it was about to be intercepted by the robots. Hence, the reaching points on the object became kinematically infeasible for the robots to reach.

To systematically study the performance of the IK solvers and sensitivity of the framework to unmeasured object positions, two sets of simulations were designed to reach for a moving box. The size of the box is same as the size of the

[7]The cases which the operator moved outside of the robots' worksapces are excluded.

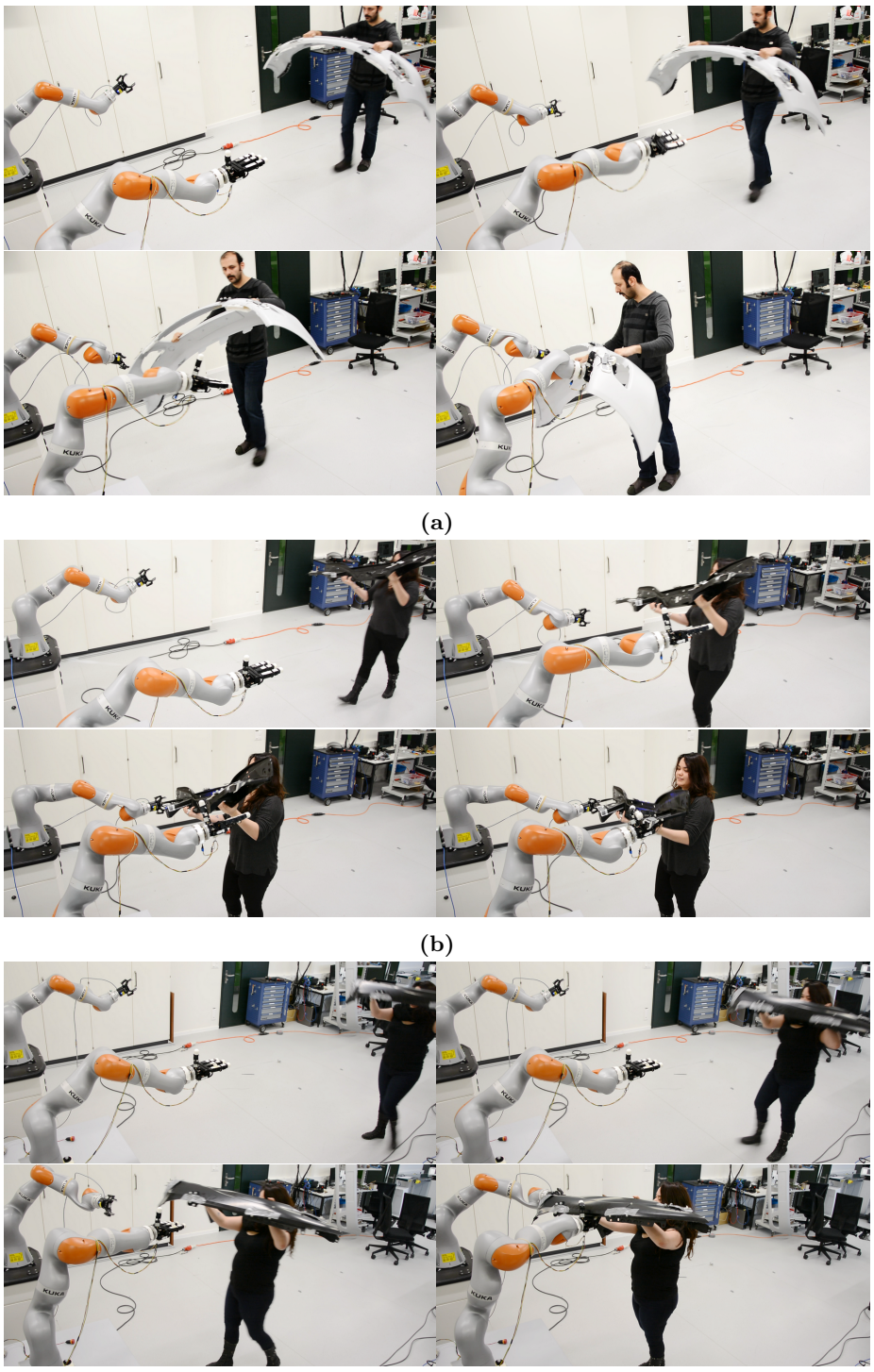


Figure 4.16: Snapshots from systematically assessment experiment. The objects are a bumper, a front panel and a fender in (a), (b) and (c), respectively. The robots are stopped when the hand and the gripper are closed. A corresponding video is available at https://youtu.be/S5fvr_wZ_W0.

Table 4.2: The details of the systematically assessment experiments. All the positions are expressed with respect to the base of the KUKA IIWA 7 robot. The starting positions are randomly chosen by the operator. The robots do not move till the first intercept point is calculated. We call the position of the object at this time the first point.

	Bumper
Weight	2.2
Material	Plastic
The bases distance	$[0.2 \quad -1.4 \quad 0.1]$
Initial position (m)	$[-3.6 \pm 0.2 \quad -0.9 \pm 0.4 \quad 0.6 \pm 0.2]$
First point (m)	$[-2.4 \pm 0.3 \quad -1.0 \pm 0.2 \quad 0.7 \pm 0.1]$
Duration (s)	3.4 ± 1.6
Success rate	**85**%
	Front panel
Weight	2.9
Material	Metallic
The bases distance	$[0.2 \quad -1.4 \quad 0.1]$
Initial position (m)	$[-4.1 \pm 0.3 \quad -0.9 \pm 0.2 \quad 0.4 \pm 0.2]$
First point (m)	$[-2.7 \pm 0.4 \quad -0.8 \pm 0.3 \quad 0.6 \pm 0.1]$
Duration (s)	3.2 ± 1.3
Success rate	**85**%
	Fender
Weight	2.4
Material	Metallic
The bases distance	$[0.0 \quad -1.3 \quad 0.1]$
Initial position (m)	$[-3.8 \pm 0.5 \quad -0.7 \pm 0.3 \quad 0.6 \pm 0.1]$
First point (m)	$[-2.7 \pm 0.4 \quad -0.7 \pm 0.2 \quad 0.7 \pm 0.1]$
Duration (s)	2.3 ± 1.2
Success rate	**90**%

bumper. In both scenarios, the object is moving toward the robots on a straight line. The simulations are conducted in the kuka-rviz environment.

IK Solver Performance.

In the first set of simulations, the performance of the three solvers of (4.5.2) (CVXgen, Nlopt and the dynamical system (4.5.5)) are assessed in terms of computation time and the smoothness of the generated joint motions. The initial velocity of the object is fixed but the initial position is randomly chosen within the range of $\begin{bmatrix} -3.5 \pm 0.05 & -0.45 \pm 0.05 & 0.8 \pm 0.05 \end{bmatrix} m$. The simulation is repeated 5 times for each solver which results in more than 5×35000 data points. The termination tolerance of the solvers is set to 10^{-4}. The computation time of each solver is illustrated in Figure 4.19a. As it was expected, CVXgen is the fastest solver and it takes about 0.000358s for it to solve (4.5.2) in average. The performance of the our implementation of (4.5.5) takes approximately 0.00092s to solve (4.5.2). As initialization of the dynamical system (4.5.5) plays important role in the convergence duration, the standard deviation of the computation

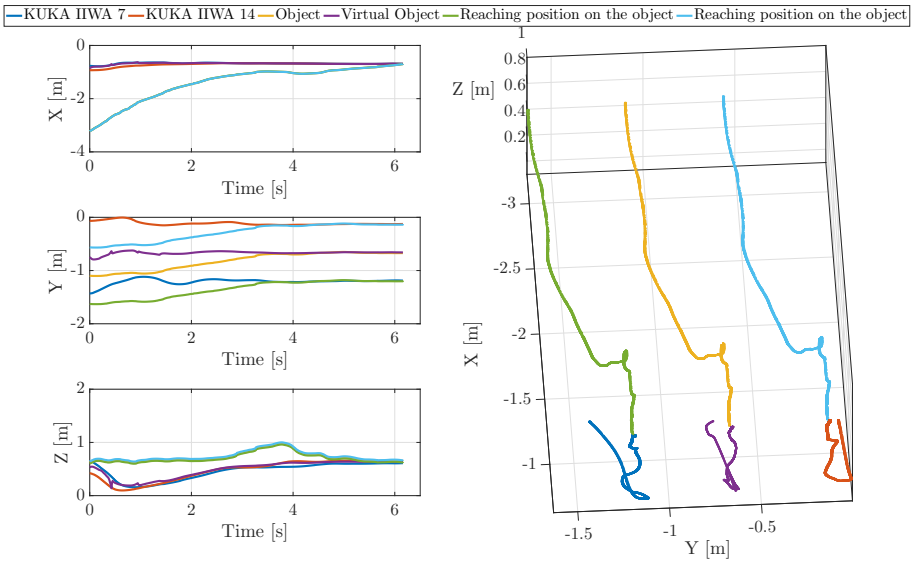


Figure 4.17: The position of the end-effector, the virtual object generated by (4.4.1) and (4.4.10), respectively. These trajectories are illustrated from the first point till the stop point. As expected, both arms intercept the reaching positions on the object at the same time. In order to avoid any internal forces, the robots are stopped once the fingers are closed on the object.

time of (4.5.5) is much higher than other two approaches. The smoothness of the trajectory L is assessed by $\mathcal{S} = \frac{\text{std}(\dot{L})}{\text{mean}(\dot{L})}$, where a smaller value of it indicates a smoother motion. As it is shown in Figure 4.19b, the result of (4.5.5) is much smoother than the other methods. It was expected as (4.5.5) calculates the desired motion at the acceleration level. Hence, the output of (4.5.5) can directly be transmitted to the robots, but the outputs of either Nlpot or CVXgen need to be filtered. As the computation power was the main criterion for choosing the IK solver for us, we mostly used CVXGEN during the experiments.

Sensitivity to noise.

In the second set of simulations, the robustness of the framework to noise and unmeasured object position is assessed. The simulation is repeated 165 times in total for three different object velocities; i.e., $0.25\frac{m}{s}$, $1.25\frac{m}{s}$ and $3.75\frac{m}{s}$. Results from this evaluation indicate that the interception error is directly correlated to the percentage of unmeasured object points and the velocity of the moving object (see Figure 4.20). Thus, the faster the object, the more sensitive the system is to the inaccuracies.

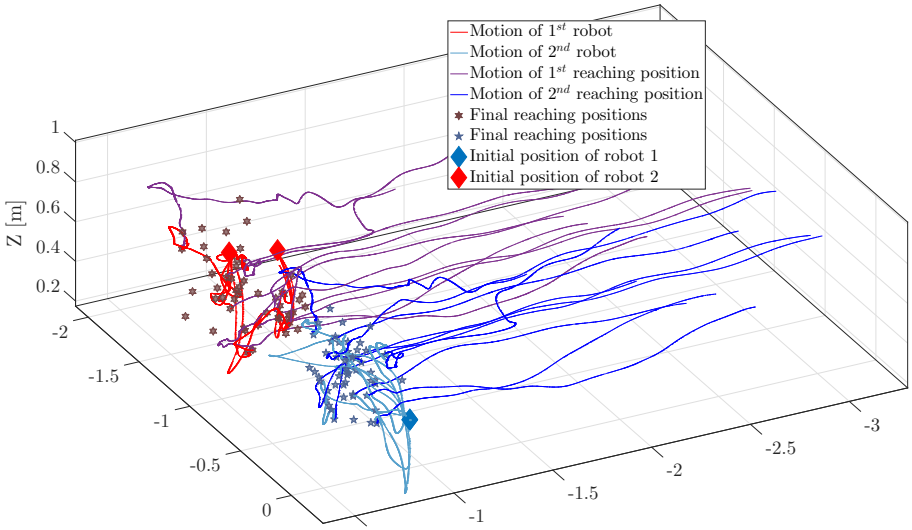
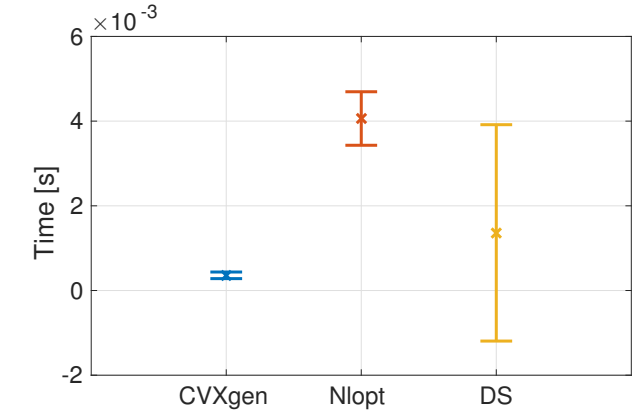


Figure 4.18: Spatial Variation of final intercept points. For clarity, only ten runs (i.e., trajectories) of the objects' and robots' motions are shown. For each run, a different object trajectory and final intercept points were observed. Experimental results verified that the robots intercept the object in *synchrony*.

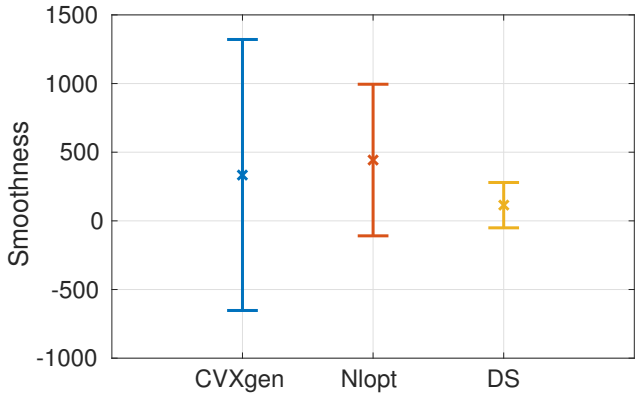
4.7 Conclusion and Discussion

In this chapter, we extend our previous formulation (introduced in Chapter 3) of dynamical systems for generating robots' motions while complying not only with the motion of a moving object but also with the motion of each other. If provided with robotic arms that can travel fast enough, our algorithm can select the most feasible robotic arms to intercept the object in coordination and with the velocities aligned to (or equal to) that of the object while any collision between the arms are avoided. For selecting the most feasible robotic arms, we define a parameter (i.e the *synchronization* parameter) to assign the robots to the appropriate behavior; i.e., the *asynchronous* or *synchronous* behaviors. The *synchronization* parameter varies between zero and one based on the feasibility criteria.

Similar to Chapter 3, in this chapter, we proved the stability and convergence of the proposed dynamical system based motion generator; i.e., (4.4.1) and (4.4.10). However, as no constraint is imposed on the magnitude of the eigenvalues ($|\lambda_{A_{ij}}|$) of A_{ij} $\forall(i,j) \in \{(1,1),(1,2),\ldots,(N_R, d_{s_{N_R}})\}$, the rate of convergence may not be fast enough such that (4.4.10) converges to $\gamma\xi^O$ in time. In two cases, this might cause a failure. In the first case, as the reaching points are assigned to the corresponding robots based on their Euclidean distance to the robots' end-effectors, they might be assigned to different robots during one trial. In the second case, the robots' asynchronous targets are far from the synchronous targets. In these cases, most likely the system fails to converge to the synchronous targets fast enough. To address this challenge, a potential direction



(a) Computation time comparison



(b) Smoothness of the motion comparison

Figure 4.19: The results of performance of the IK solvers in terms of computation time and the smoothness of the motion. DS stands for the dynamical system (4.5.5).

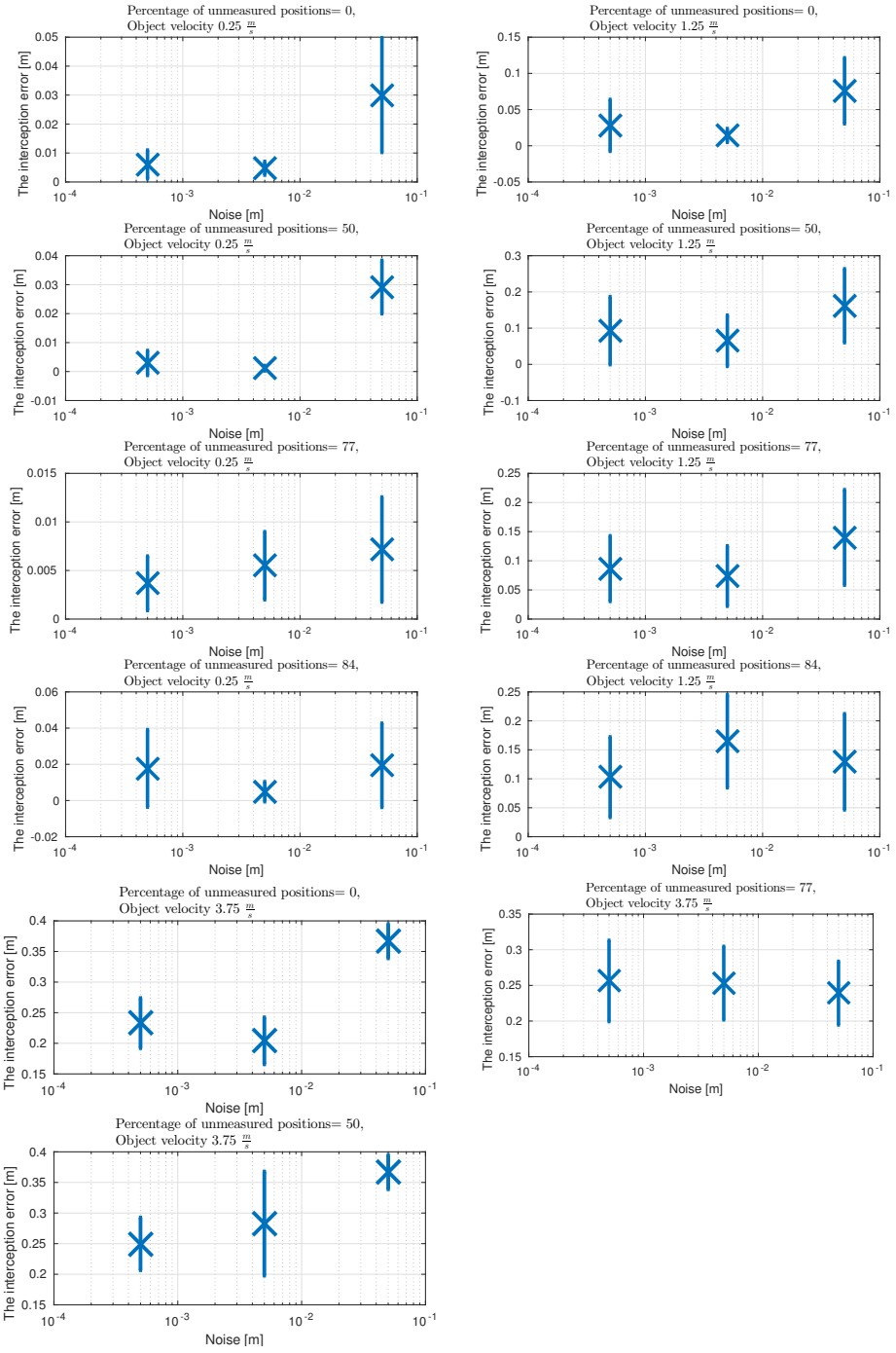


Figure 4.20: The interception error is the average of the minimum distance between the box and the end-effectors. The initial positions of the box are randomly chosen within the range of $\begin{bmatrix}-3.5 \pm 0.1 & -0.45.0 \pm 0.1 & 0.8 \pm 0.1\end{bmatrix}^T m$. The distance between the arms and the size of the box are same as the bumper scenario. The simulations are repeated for each combination of three object's speeds, three noise powers and four percentages of unmeasured positions. We only consider trials when the box passes through the robots workspaces. The measurement noise is simulated with pseudo-random values within the range of ± 0.05, ± 0.005 and ± 0.0005m. The results of the worse case are not illustrated as the robots were not able to follow the object; i.e., the worse case is when the percentage of unmeasured position is 85% and the box's velocity is $3.75 \frac{m}{s}$.

for future work would be to estimate the parameters of (4.4.1) and (4.4.10) with respect to the stability and the convergence rate constraints.

To solve the quadratic programming problem, we used three different approaches. First a dynamical system based approach, Eq.(4.5.5). Second, nonlinear programming solver (Johnson, 2015). The third approach was CVXGEN, introduced in (Mattingley and Boyd, 2012). Each of these approaches has its own advantages and disadvantages. Using the first approach is advantageous in the way that the passivity of the dynamical system can be proven. Hence, the unified framework stays passive and stable as long as the robots are passive. In addition, the first approach result in smoother joint motions. The main advantage of the second approach is its interface. Nlopt is very user friendly and it is possible to test several different solvers, but it is computationally expensive. The main advantage of the third approach is the computational cost. As it has been shown in (Mattingley and Boyd, 2012), CVXGEN is computationally very efficient. The main shortcoming of the third approach is the stability of the closed loop system which can not be proven; however any unstable behaviors during the real world evaluations or the simulations has not been seen.

Similar to Chapter 3, throughout the proofs, we assume that the intercept point is a fixed attractor. However, due to the imperfect prediction of the object trajectory, the feasible intercept postures need to be iteratively updated. Nevertheless, this does not affect the convergence of the system for two main reasons. First, when $\gamma < 1$ the new feasible intercept point is chosen in the vicinity of the previous ones; i.e., the convergence rate is much faster than the rate of update. Second, when the object is reachable, $\gamma = 1$, the virtual object converges to the real object and the position of the intercept point does not affect the convergence. Similarly, we have also assumed that the reaching points on the object do not change or their changing rates are much less than the convergence rate of the dynamical systems.

In Chapter 3 and Chapter 4, we introduced compliant control architectures for actively synchronizing the motion of the robots with a moving object and with each other at different levels. However, the proposed controllers are tailored for the reaching and softly intercepting moving objects scenarios; which is a particular case of non-contact/contact scenarios. In the next chapter, we propose a general solution for actively controlling the motion of the robots to stably performing any non-contact/contact scenarios.

Chapter 5

Stable non-contact/Contact Transitions

5.1 Introduction

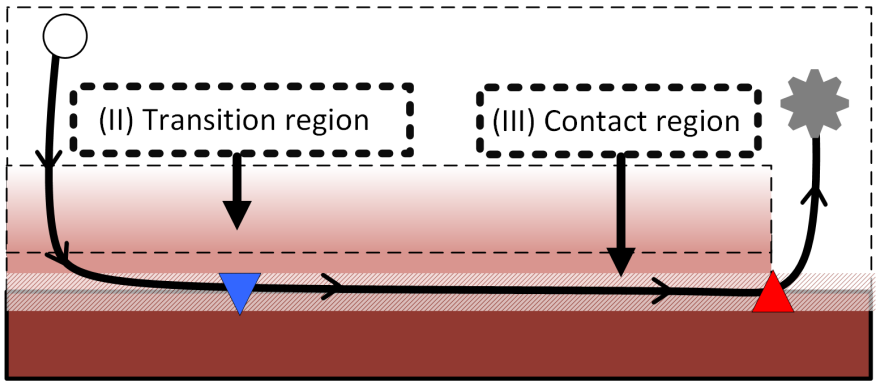
In this chapter, we consider the problem of controlling the robotic manipulators during non-contact/contact transitions. A wide variety of real-world manipulation tasks, such as: milling/ polishing/finishing workpieces (Jinno et al., 1995; Kabir et al., 2017; Pagilla and Yu, 2001b; Khansari et al., 2016), wiping/painting surfaces (Lin et al., 2017; Leidner et al., 2016), peeling or dough rolling Figueroa et al. (2016) includes interactions between a tool and an environment. For such applications, the complete scenario can be categorized into three *regions*: (i) Moving in *free motion* space and approaching the contact surface; i.e., *Free motion* region. (ii) Establishing contact with the surface; i.e., *Transition* region. (iii) Maintaining contact with the surface while moving in the other directions; i.e., *Contact* region; see Figure 5.1. *We call a contact stable if the impact happens only once and the robot remains in contact with the surface after the impact.*

Realizing a stable contact is particularly challenging as the impact leaves infinitesimal time for the robot to react properly to the impact force. It is however necessary to control for a stable contact to avoid that the robot bounces on the surface and damages itself or the environment.

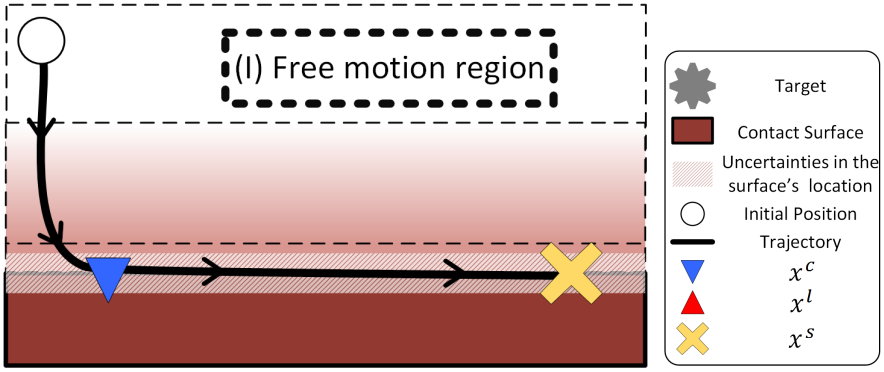
The complexity of the problem increases importantly if the arm and the surface do not dissipate impact energy; i.e perfectly elastic impact. In this case, to successfully establish a contact with a rigid surface, the robot should touch the surface with zero velocity so that the post-contact velocity in the normal direction is zero. Nevertheless, the impacts in real-world scenarios are mainly inelastic, where, based on Newtons's law, the post-contact velocity of an object is a constant fraction of the pre-contact velocity (Jia et al., 2013).[1] In this case, touching the surface with zero, or near to zero, velocity results in a zero post-contact velocity in the normal direction; i.e., the robot remains in contact with the surface after the impact (Pagilla and Yu, 2001a, 2000).

Previously, in Chapter 3 and Chapter 4, we introduced actively complaint

[1] It is worth mentioning that Newton's model is one of the simplest mathematical model which describes an impact. In recent years, Possition's and the energy hypotheses are proposed to address shortcomings of Newton's model. However, if the object is passive, and the surface has no velocity, the post-contact velocity is always equal or less than the pre-contact velocity.



(a) Scenario 1.



(b) Scenario 2.

Figure 5.1: Schematic of three subtasks of an interactive application while the surface's location is uncertain. The arm starts approaching the surface at the *free motion* region. Once it is close enough to the surface, it regulates its velocity to establish a stable contact at the desired contact point (x^c). Then while sliding on the surface, based on the scenario requirements, it either leaves the surface (Figure 5.1a) or reaches the target on the surface (Figure 5.1b) at desired departure (x^l) or stop (x^s) locations, respectively.

control architectures for intercepting a moving object at the desired position with zero relative velocity by single or multi-arm system. However, the proposed controllers were particularly designed for accomplishing the specific tasks. In this chapter, as the *transition* is a local behavior and to comply with the velocity constraints, we propose a strategy consisting of locally modulating the motion of the robot once it is close to the surface such that the robot's velocity is aligned with the contact surface so that a stable contact can be established. The proposed architecture can be integrated into existing DS-based motion control approaches, where they represent the nominal arm behavior. The idea of locally modulating dynamical systems is not novel and it has been previously used in (Khansari-Zadeh and Billard, 2012) and (Kronander et al., 2015) for modulating first order DSs. In this contribution, we use this idea to modulate the motion of a robot such that:

Objective 1 : If the robot contacts the surface, the impact happens only once

and the robot remains in contact after the impact.

Moreover, we show that the proposed controller is capable of modulating the robot's motion such that:

Objective 2 : The robot contacts the surface at a specific point (x^c).

Objective 3 : If the robot is in contact with the surface, it slides on the surface and either

(a) leaves the surface at a specific departure location (x^l), see Figure 5.1a, or

(b) stops at a specific stop location (x^s) on the surface, see Figure 5.1b.

This chapter is began by formalizing our assumptions and problem formulation in Section 5.2. Then in Section 5.3, we develop our compliance modulation function. Performance of the approach is evaluated on real world robot experiments in Section 5.4. This chapter concludes with a discussion in Section 5.5.

This chapter corresponds to the following publication:

- Mirrazavi Salehian, S. S. and Billard, A. A dynamical system based Approach for controlling robotic manipulators during non-contact/contact transitions, 2018. (Submitted).

5.2 Problem Statement

Suppose the contact surface is non-penetrable, passive and planar. Moreover, a continuous function ($\Gamma(x) = N^T x$), which conveys a notation of distance to the surface is available; where N is the unit normal vector to the surface. x denotes the position of the robot's end-effector. By definition, the origin of the coordinate frame is on the surface and the surface corresponds to the plane of equation $N^T x = 0$. Based on this definition, one can categorize the task space into two regions; the *free motion* region when $0 < N^T x$ and the *contact* region when $N^T x = 0$.

We consider the following continuous-time system. As we need to control both position and velocity, the DS must be a function of both of them and the output defines the desired acceleration of the robot.

$$\ddot{x} = \mathsf{M}(x, \dot{x}) f(x, \dot{x}, t) \tag{5.2.1}$$

Where $f(x, \dot{x}, t)$ represents the nominal dynamical system. We assume that the nominal DS is asymptotically stable to a fixed target (x^t) located above the

surface; i.e., $0 < N^T x^t$.[2]

Furthermore, the nominal acceleration is non-zero everywhere except on the target; i.e., $f^T(x,\dot{x})f(x,\dot{x}) \neq 0 \ \forall (x,\dot{x}) = \mathbb{R}^{d\times d} - \{x^t, 0\}$.[3] $\mathsf{M}(x,\dot{x}) \in \mathbb{R}^{d\times d}$ is a modulation function which reshapes the nominal DS such that it complies with the contact surface based on the state of the robot, where we define the modulation function as follows:

$$\begin{aligned} \mathsf{M}(x,\dot{x}) &= Q\Lambda Q^{-1} \\ Q &= \begin{bmatrix} q_1 & \dots & q_d \end{bmatrix} \end{aligned} \tag{5.2.2}$$

Where $q_i \ \forall i \in \{1,\dots,d\}$ form an orthonormal basis in $\mathbb{R}^d$ with q_1 pointing along the normal to the surface; i.e., $q_1 = N$. $\boldsymbol{\lambda}_{ij}(x,\dot{x}) \ \forall i,j \in \{1,\dots,d\}$ are the entries of Λ, where i is the row number and j is the column number. The motion direction, tangential and normal to the surface, can be controlled through the scalar values $\boldsymbol{\lambda}_{ij} \ \forall i,j \in \{1,\dots,d\}$. For example, by setting $\boldsymbol{\lambda}_{1j}(x,\dot{x}) = 0 \ \forall j \in \{1,\dots,d\}$, the acceleration of the robot normal to the surface will be zero; i.e., $N^T\ddot{x} = 0$. Moreover, by setting $\boldsymbol{\lambda}_{ii}(x,\dot{x}) = 1$, $\boldsymbol{\lambda}_{ij}(x,\dot{x}) = 0 \ \forall i,j \in \{1,\dots,d\}$, $i \neq j$, the nominal DS drives the robot in the $q_i{}^{\text{th}}$ direction. We exploit this property and limit the influence of the modulation function to a region in a vicinity of the surface; denoted as the *transition* region.[4]. Using the fact that we have at our disposal the function $\Gamma(x)$ to measure the distance to the surface, we set the *transition* region to be all points such that $\Gamma(x) \leq \rho$, $\rho \in \mathbb{R}_{>0}$. Outside this region, to avoid undesirable modulations, the modulation decreases and vanishes exponentially as a function of the distance to the surface. To modulate locally the dynamics of the DS given by (5.2.1) and (5.2.2), we set that:

$$\boldsymbol{\lambda}_{ij}(x,\dot{x}) = \begin{cases} \lambda_{ij}(x,\dot{x}) & \text{if } \Gamma(x) \leq \rho \\ (\lambda_{ij}(x,\dot{x}) - 1)\, e^{\frac{\rho-\Gamma(x)}{\sigma}} + 1 & \text{if } i = j \ \ \rho < \Gamma(x) \\ \lambda_{ij}(x,\dot{x}) e^{\frac{\rho-\Gamma(x)}{\sigma}} & \text{if } i \neq j \ \ \rho < \Gamma(x) \end{cases} \tag{5.2.3}$$

$\forall i,j \in \{1,\dots,d\}$ Where $0 < \sigma$ defines the speed at which the modulation vanishes in the free motion region. ρ defines the region of the influence of the modulation function. If $\rho < \Gamma(x)$, the robot is far from the contact surface and $\Lambda = I_{d\times d}$; i.e., the robot is driven solely by the nominal dynamical system.

Next we show how by defining $\lambda_{ij} \ \forall i,j \in \{1,\dots,d\}$, a stable contact can be

[2] It is important to note that asymptotically stability of the nominal DS is only required to achieve *Objective 3*.a. To achieve the other objectives(*Objective 1*, *Objective 2* and *Objective 3*.b) it is not necessary for the nominal DS to be stable.

[3] If $M(x,\dot{x})$ is the control input, this assumption is equivalent to the controllability of $\ddot{x} = \mathsf{M}(x,\dot{x})f(x,\dot{x},t)$.

[4] The development of the *transition* region is partly inspired from the potential field obstacle avoidance approaches Khatib (1986).

achieved. Moreover, we define ρ based on the kinematic constraints of the robot. First, we consider a perfect elastic impact between the robot and the contact surface; i.e., the Coefficient Of Restitution (COR)[5] (e) is one. Then, we extend this to a realistic scenario where the impact is inelastic; i.e., $0 \leq e < 1$.

5.3 Compliant Modulation Systems

5.3.1 The elastic impact

Consider a scenario where the impact is perfectly elastic ($e = 1$). In this case, the normal velocities[6] of the robot before and after the impact are equal in amplitudes but pointing to opposite directions. Hence, to achieve a stable contact (*Objective 1*), the normal velocity of the robot at contact must be zero; i.e.

$$N^T \dot{x}(t^*) = 0 \tag{5.3.1}$$

Where, t^* is the time when the robot enters into contact with the surface.

Theorem 5. *For a given initial state* $\{x_0, \dot{x}_0 \in \mathbb{R}^d |\ 0 < N^T x_0 \leq \rho, f(x_0,\ \dot{x}_0) \neq 0\}$*, the motion generated by* (5.2.1) *and* (5.2.2) *contacts the surface with zero normal velocity and satisfies Objective 1, if* $\forall j \in \{1, \ldots, d\}$

$$\lambda_{1j}(x, \dot{x}) = \left(-2\omega N^T \dot{x} - \omega^2 N^T x\right) \mathbf{f}_j(x, \dot{x}, t) \tag{5.3.2}$$

where $\mathbf{f}_j(x, \dot{x}, t) = \frac{f(x,\dot{x},t)^T q_j}{f(x,\dot{x},t)^T f(x,\dot{x},t)}$ *and*

$$\frac{|N^T \dot{x}_0|}{N^T x_0} \leq \omega \tag{5.3.3}$$

Moreover, the motion generated by (5.2.1) *and* (5.2.2) *contacts the surface at* x^c *and satisfies Objective 2, if* $\forall (i, j) \in \{(2,1), (2,2), \ldots, (d,d)\}$

$$\lambda_{ij}(x, \dot{x}) = \left(-2\omega {q_i}^T \dot{x} - \omega^2 {q_i}^T (x - x^*)\right) \mathbf{f}_j(x, \dot{x}) \tag{5.3.4}$$

Where $x^* = x^c$.

Proof: see Appendix C.1.

Theorem 5 provides a function to modulate the motion of the robot's end-effector such that the stable contact can be established at the desired location.

[5] COR is defined as the ratio of velocities after and before an impact, taken along the line of the impact.

[6] For sake of simplify, in the rest of the paper, we call the velocity normal to the surface, the normal velocity.

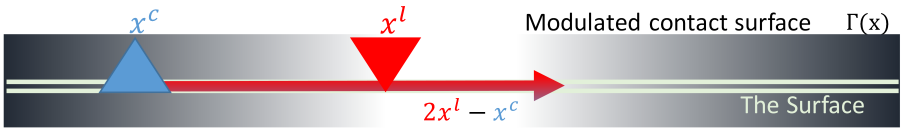


Figure 5.2: An illustration of the variables in (5.3.10). The value of modulated contact surface ($\Gamma(x)$) is depicted by the intensity of the black, where $\rho \leq \Gamma(x)$ in the white areas and $\Gamma(x) < \rho$ in the rest. The colors of the variables and the arrows are corresponding.

However, it is important to note that defining the pre-specified contact location is not necessary for implementing the proposed the modulation framework. For instance, by defining

$$\lambda_{ij}(x, \dot{x}) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \quad \forall i \in \{2, \dots, d\} \quad \forall j \in \{1, \dots, d\} \tag{5.3.5}$$

and $\lambda_{1j}(x, \dot{x})$, $\forall j \in \{1, \dots, d\}$ by (5.3.2), the motion of the nominal DS is modulated only in the normal direction. Hence, if the robot enters the transition region, it stably contacts the surface as the normal velocity of the robot is modulated based on (5.3.2). However, the contact location emerges from the motion generated by the nominal DS.

If the robot starts its motion outside of the transition region, Eq. (5.2.3) states that the modulation function is activated if the robot enters it. Hence, the initial state ($N^T x_0$) in Theorem 5 is equivalent to ρ. However, Theorem 5 depends on the robot's dynamics and is achievable only if the robot can decelerate sufficiently rapidly. Hence, the transition region must be set sufficiently large to meet the robot's physical limits. This is summarized in the following proposition:

Proposition 2. *For a robot with upper bounds $\dot{x}_{max}$ and $\ddot{x}_{max}$ on velocity and acceleration, respectively. Given $N^T x_0 = \rho$, we set ρ and ω in (5.2.3) such that*

$$\rho = \frac{3(N^T \dot{x}_{max})^2}{|N^T \ddot{x}_{max}|} \tag{5.3.6}$$

$$\omega = |\frac{N^T \ddot{x}_{max}}{3N^T \dot{x}_{max}}| \tag{5.3.7}$$

Proof: see Appendix C.2.

Once the robot is in *contact* with the surface, two interactive scenarios can be accomplished. In the first scenario, the robot slides on the surface and leaves it at the specific departure location (x^l); see Figure 5.1a. In the second scenario, the robot slides on the surface till it reaches the desired stop location on the surfaces (x^s); see Figure 5.1b. The latter can be realized by modulating the

nominal DS and x^*. However, the former can be achieved by modulating the nominal dynamical system, x^* and the definition of $\Gamma(x)$. These are summarized in the following propositions.

Proposition 3. *For a given initial state* $\{x_0, \dot{x}_0 \in \mathbb{R}^d |\ N^T x_0 \leq \rho, f(x_0,\ \dot{x}_0) \neq 0\}$, *the motion generated by the nominal DS* (5.2.1) *modulated by* (5.2.2), *where* $\lambda_{ij}(x, \dot{x}),\ \forall (i,j) \in \{(1,1),(1,2),\ldots,(d,d)\}$ *are defined by* (5.3.2) *and* (5.3.4), *contacts the surfaces at* x^c *and then slides on the surface till it asymptotically reaches* x^s *(i.e., satisfaction of Objective 3.b) if* x^* *in* (5.3.4) *is such that:*

$$x^* = \begin{cases} x^c & \text{if } 0 < N^T x \\ x^s & \text{if } \ N^T x = 0 \end{cases} \tag{5.3.8}$$

Where, x^s is defined on the surface.

Proof: see Appendix C.3.

Proposition 4. *For a given initial state* $\{x_0, \dot{x}_0 \in \mathbb{R}^d |\ N^T x_0 \leq \rho, f(x_0,\ \dot{x}_0) \neq 0\}$, *the motion generated by the nominal DS* (5.2.1) *modulated by* (5.2.2), *where* $\lambda_{ij}(x, \dot{x}),\ \forall (i,j) \in \{(1,1),(1,2),\ldots,(d,d)\}$ *are defined by* (5.3.2) *and* (5.3.4), *contacts the surfaces at* x^c *and then leaves it at* x^l *(i.e., satisfaction of Objective 3.a) if* x^* *and* $\Gamma(x)$ *in* (5.3.4) *and* (5.2.3), *respectively, are defined as follows:*

$$x^* = \begin{cases} x^c & \text{if } 0 < N^T x \\ 2x^l - x^c & \text{if } \ N^T x = 0 \end{cases} \tag{5.3.9}$$

$$\Gamma(x) = N^T x + \left(\rho - (x^l - x^c)^T (x^l - x)\right) e^{-(x^l - x)^T \Sigma^{-1} (x^l - x)} \tag{5.3.10}$$

Where, x^l *is defined on the surface and* $\Sigma \in \mathbb{R}^{d \times d}$ *is a positive definite matrix.*

Proof: see Appendix C.4.

Changing x^* based on (5.3.8) or (5.3.9) does not cause oscillations as (5.3.2) is not a function of x^*. Hence, changing x^* does not influence the motion of the robot in the normal direction to the surface. Σ defines the influence of $\left(\rho - (x^l - x^c)^T (x^l - x)\right) e^{-(x^l - x)^T \Sigma^{-1} (x^l - x)}$ over $N^T x$ in (5.3.10); see Figure 5.2. If all entries in Σ are small, its influence will be small and vice-versa.

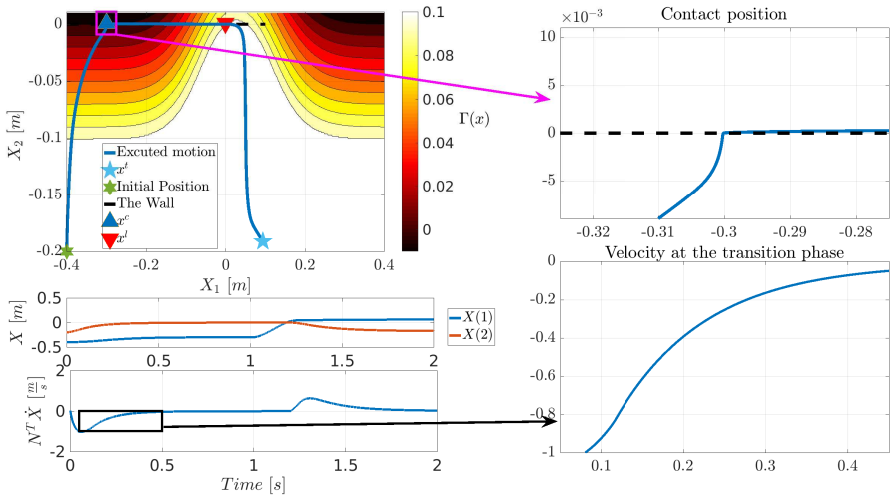


Figure 5.3: An intuitive example showing the behavior of the proposed control in three different regions. Stable contact is achieved by using the proposed modulation function. $\Gamma(x)$ is defined based on (5.3.10). The contact is assumed inelastic. $\rho = 0.1$, $\sigma = 0.01$, $\omega = 0.1$, $\nu = 0.01$ and $\delta_{\dot{x}} = -0.02$. As it is illustrated, at the *transition* region, the velocity of the robot is reduced based on Theorem 6 such that it satisfies (5.3.11). Moreover, in the tangential directions, the robot regulates its velocity based on (5.3.4) and touches the surface exactly at the desired contact point ($x^c = [-0.3 \quad 0]^T$). As it is shown in this example, based on Proposition 4, the robot leaves the surface at the desired leaving point ($x^l = [0 \quad 0]^T$) . $f(.) = \begin{bmatrix} -1 & 0 \\ -40 & -25 \end{bmatrix} \dot{x} + \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} (x - x^t)$, where $x^t = [0.1 \quad -0.2]^T$.

5.3.2 The inelastic impact

In an inelastic impact, due to the action of internal friction, the kinetic energy is not conserved and hence the coefficient of restitution is less than one ; i.e., $0 \leq e < 1$. In this case, we can assume that if the normal velocity of the robot becomes very small ($\delta_{\dot{x}}$), the surface absorbs all the kinetic energy of the arm; i.e., the end-effector remains in contact after the impact[7]. Hence, to achieve *Objective 1*, the velocity of the robot must satisfy the following constraint at impact:

$$\delta_{\dot{x}} \leq N^T \dot{x}(t^*) \leq 0 \tag{5.3.11}$$

Theorem 6. *Assuming the impact is inelastic. For a given initial state* $\{x_0, \dot{x}_0 \in \mathbb{R}^d | \ 0 < N^T x_0 \leq \rho, \ f(x_0, \dot{x}_0) \neq 0\}$, *the dynamical system* (5.2.1) *and* (5.2.2)

[7] This assumption is adopted from Pagilla and Yu (2001a).

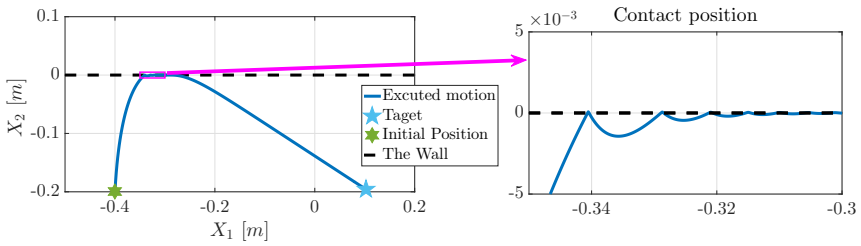


Figure 5.4: The modulation function is disabled by setting $M(.) = I$. Not only the contact is unstable and the robot bounces on the surface, as the impact force is enormous, but also the robot does not contact the surface at the desired location. $f(.) = \begin{bmatrix} -1 & 0 \\ -40 & -25 \end{bmatrix} \dot{x} + \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} (x - x^t)$, where $x^t = [0.1 \ \ -0.2]^T$.

satisfies condition (5.3.11)*, if* $\lambda_{1j}(x, \dot{x}) =$

$$\begin{cases} \omega \left(-N^T \dot{x} + (\delta_{\dot{x}} + \nu) \right) \mathbf{f}_j(x, \dot{x}) & N^T \dot{x} < \delta_{\dot{x}} \quad (5.3.12) \\ \omega \left(\dfrac{\nu}{\delta_{\dot{x}}} N^T \dot{x} - \omega (1 - \dfrac{N^T \dot{x}}{\delta_{\dot{x}}}) N^T x \right) \mathbf{f}_j(x, \dot{x}) & \delta_{\dot{x}} \leq N^T \dot{x} \leq 0 \quad (5.3.13) \\ \omega \left(-2 N^T \dot{x} - \omega N^T x \right) \mathbf{f}_j(x, \dot{x}) & 0 < N^T \dot{x} \quad (5.3.14) \end{cases}$$

where $\mathbf{f}_j(x, \dot{x}) = \frac{f(x,\dot{x},t)^T q_j}{f(x,\dot{x},t)^T f(x,\dot{x},t)}$ *and* ω *is defined based on* (5.3.4) *(or its equivalent* (5.3.7)*) and*

$$\frac{\delta_{\dot{x}} - N^T \dot{x}_0}{e^1 - 1} < \nu \qquad (5.3.15)$$

Proof: see Appendix C.5.

As shown in Table 5.1, $\lambda_1(x, \dot{x})$ defined by (5.3.12)-(5.3.14) is continuous. The main advantages of the modulation function proposed for the inelastic impact over the elastic one, is its ability in handling uncertainties in the surface location:

Proposition 5. *Assuming a planar surface with equation* $N^T x = \eta$*, whose orientation is precisely defined through its normal* (N) *but whose location* (η) *is imprecise but bounded with a known upper bound* $\boldsymbol{\eta}$*, i.e.,* $|\eta| \leq \boldsymbol{\eta} < \rho$*. Moreover, for a given initial state* $\{x_0, \dot{x}_0 \in \mathbb{R}^d | \ N^T \dot{x}_0 < \delta_{\dot{x}} < 0, 0 \leq \boldsymbol{\eta} < N^T x_0 \leq \rho, \}$*, the dynamics of the robot is generated by the nominal DS* (5.2.1) *modulated by* (5.2.2)*, where* $\lambda_{1j}(x, \dot{x})$*,* $\forall (j) \in \{1, 2, \ldots, d\}$ *are defined by* (5.3.12)*. Then, the velocity of the robot when it impacts the surface is bounded and satisfies condition* (5.3.11)*, if* ν *and* ω *are defined as follows:*

$$\nu = -\delta_{\dot{x}}, \quad \omega = \frac{\delta_{\dot{x}} - N^T \dot{x}_0}{N^T x_0 - \boldsymbol{\eta}} \qquad (5.3.16)$$

Proof: see Appendix C.6.

Proposition 5 hence ensures that the contact is stable and *Objective 1* is

Table 5.1: Value of $\lambda_{1j}(x, \dot{x})$ defined by (5.3.12)-(5.3.14) at the conditions' boundaries.

	$\lim_{N^T\dot{x}\to\delta_{\dot{x}}^-}$	$N^T\dot{x} = \delta_{\dot{x}}$	$\lim_{N^T\dot{x}\to\delta_{\dot{x}}^+}$
$\lambda_1(x, \dot{x})$	$\omega^{-1}\nu\mathbf{f}_j(x, \dot{x})$ (5.3.12)	$\omega^{-1}\nu\mathbf{f}_j(x, \dot{x})$ (5.3.13)	$\omega^{-1}\nu\mathbf{f}_j(x, \dot{x})$ (5.3.13)
	$\lim_{N^T\dot{x}\to 0^-}$	$N^T\dot{x} = 0$	$\lim_{N^T\dot{x}\to 0^+}$
$\lambda_1(x, \dot{x})$	$-\omega^{-2}N^T x\mathbf{f}_j(x, \dot{x})$ (5.3.13)	$-\omega^{-2}N^T x\mathbf{f}_j(x, \dot{x})$ (5.3.13)	$-\omega^{-2}N^T x\mathbf{f}_j(x, \dot{x})$ (5.3.14)

Table 5.2: The details of the systematically assessment. All the positions are with respect to the robot's base. The units are defined in the metric system. $x^c = [-0.6\ 0.3\ 0.0]^T$ and $x^l = [-0.66\ 0.31\ 0.07]^T$ and $x^s = [-0.65\ -0.01\ 0.07]^T$. $\delta_{\dot{x}} = -0.1\frac{m}{s}$ and $\rho = 0.2m$. " Contact ", "Leaving/ Stopping" errors are the Euclidean distance between the real and the desired corresponding points. "Pre-contact" and "Pre-transition" velocities are the velocity of the end-effector in the normal direction about entering the *contact* and *transition* regions, respectively.

	Scenario 1
Initial position	$[-0.5 \pm 0.2\ \ -0.0 \pm 0.4\ \ 0.8 \pm 0.1]$
Contact Error	0.0 ± 0.03
Leaving/Stopping Error	0.0 ± 0.01
Pre-contact velocity	0.08 ± 0.06
Pre-transition velocity	2.4 ± 1.4

	Scenario 2
Initial position	$[-0.5 \pm 0.1\ \ -0.0 \pm 0.3\ \ 0.8 \pm 0.1]$
Contact Error	$0.0 \pm 2 \times 10^{-3}$
Leaving/Stopping Error	$0.0 \pm 4 \times 10^{-3}$
Pre-contact velocity	0.07 ± 0.01
Pre-transition velocity	2.5 ± 1.3

satisfied. However, it depends on the fact that the uncertainty on the location of the surface must remain bounded. Moreover, if the location of the contact surface is uncertain, the contact location can not be precisely specified.

The performance of the proposed framework is illustrated by a simple intuitive 2-D example in Figure 5.3 and 5.4 , where, in Figure 5.3, by using the proposed framework, the robot can stably transit to the contact region. Figure 5.4 illustrates an unstable contact where the modulation function is disabled by setting $M(x, \dot{x}) = I$. In this case, as contact velocity is very high, the robot bounces on the surface.

5.4 Experimental Results

We consider a task of wiping the surface of a fender. The performance of the proposed framework is evaluated on a real robotic arm platform; i.e., 7 DOF robotic arm (KUKA IIWA). The robot is controlled in the joint position level at a rate of 200 Hz. The output of the DS (5.2.1) is converted into the joint state using the damped least squares inverse kinematic solver. The robot is equipped with a 6-axis ATI force-torque sensor which is only used for recording forces and not in the controller.

The surface of the fender is approximated by a plane which is calculated by capturing the position of three markers on the surface. The positions are captured by Optitrack motion capture system. The orientation of the end-effector

is constrained to be normal to the contact surface. The impact is inelastic and $\delta_{\dot{x}} = -0.1\frac{m}{s}$.

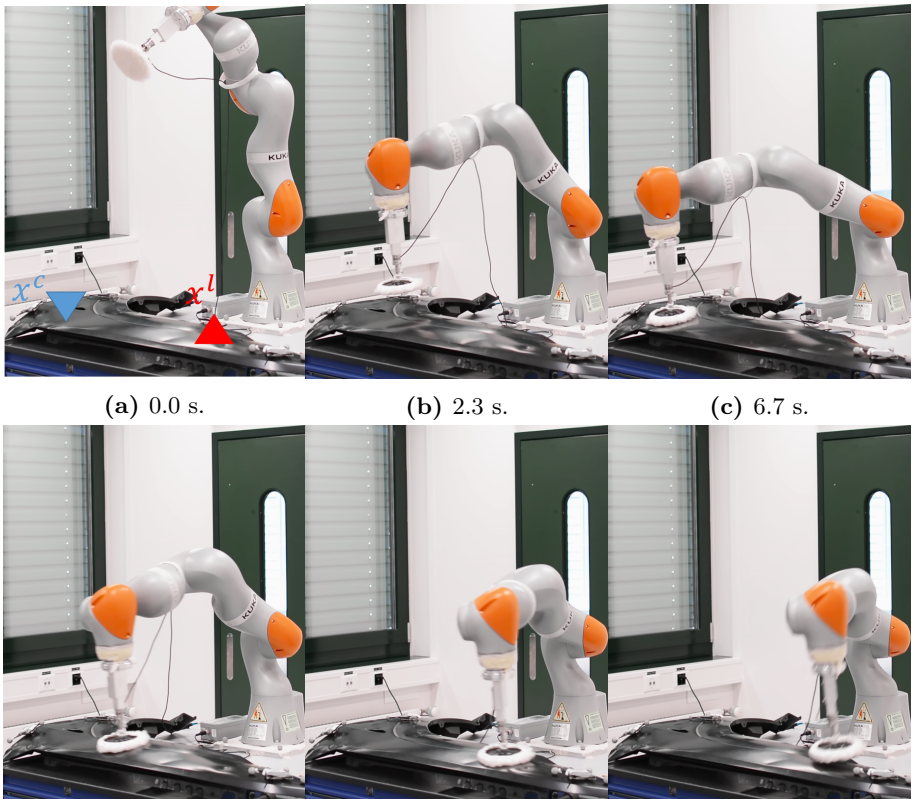
The empirical validation is divided into three parts. In the first part, we systematically assess the performance of the controller in performing the two scenarios illustrated in Figure 5.1a and Figure 5.1b in a known environment. In the second part, the controller's abilities in modulating the robot's motion in a dynamically chancing environments are assessed. In the third part, the performance of the controller is assessed in an uncertain environment.

5.4.1 Systematic assessment

Both scenarios were repeated 30 times where the initial state of the robot is randomly chosen; all the information is summarized in Table 5.2. The location of the surface is fixed. The snapshots of the motion execution of both scenarios are shown in Figure 5.5 and 5.6. Visual inspection of video[8] confirmed that, in all the trials, the robot stably touches the surface at x^c and accomplishes the tasks. However, the inspection of data indicates that in three cases the velocity at impact is higher than 0.1. An example of the motion of the robot is illustrated in Figure 5.7. As it can be seen, the normal velocity of the robot is reduced to $\delta_{\dot{X}}$ in the *transition* region to ensure a stable contact. In Figure 5.8, the variation of the robot's motions is illustrated.

As reported in Table 5.2, the overall position errors at x^c, x^l and x^s are very small and negligible. This indicates that even though the surface is not exactly planar in our implementation, our modulation function is capable of accomplishing the desired tasks. The causes of these inaccuracies can be categorized into three different categories. i) The main cause of this error is the approximation of the contact surface. In this experiment, we assumed that the contact surface is a plane. However, the fender's surface is bumpy. This results in inaccuracies in the measurement of the distance between the robot and the real surface. ii) The second cause of error is the inverse kinematic. Even thought the motion of the robot is not super fast, the IK solver is still unable to generate a very accurate joint-level motion corresponding to the desired end-effector trajectory. The kinematic constrains of the robot is the main source of this inability. iii) The third cause of the error is delays in measuring the joint positions and the communication channels. As the robot runs in the closed loop, any measuring delay causes inaccuracies in specifying the desired motion of the robot. In spite of these, the overall performance of the task execution is satisfying and the robot was able to wipe the surface successfully in all the trials.

[8] http://lasa.epfl.ch/files/Sina_Mirrazavi_Thesis_videos/Chapter_5/Systematic_Assessment.mp4.



(a) 0.0 s. **(b)** 2.3 s. **(c)** 6.7 s.

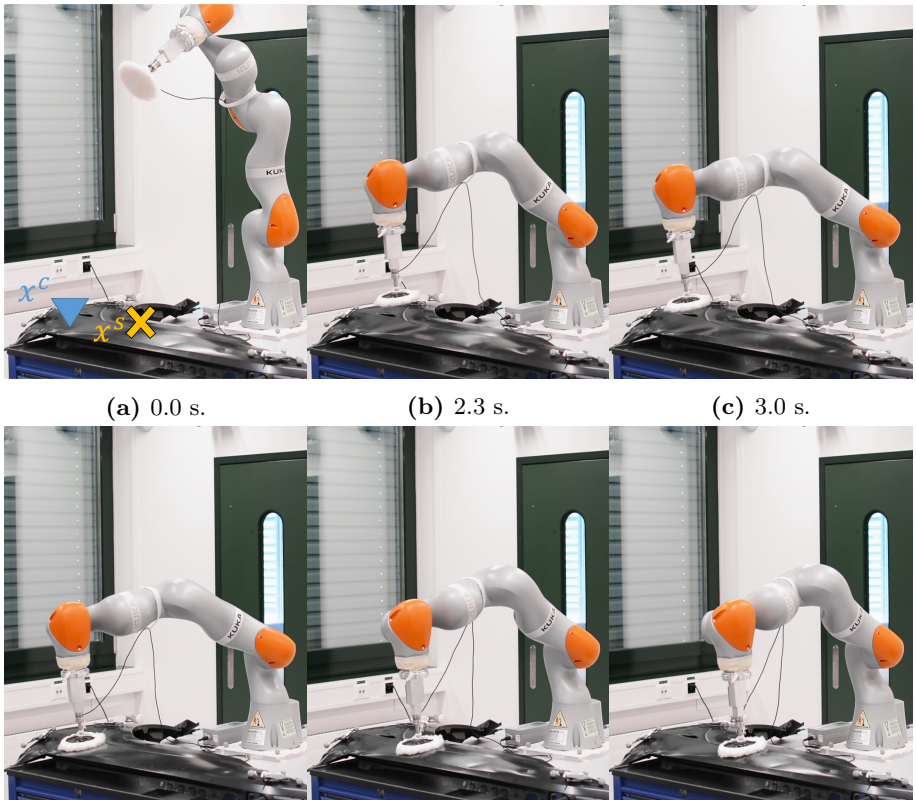
(d) 7.7 s. **(e)** 8.7 s. **(f)** 8.9 s.

Figure 5.5: Snapshots of the motion of the robot while performing Scenario 1, where the goal of the robot is to stably touch the surface at x^c and leave it at x^l. In (a), the initial configuration of the robot, x^c and x^l are shown. While the robot is at the transitions region, in (b), the robot's motion is directed towards x^c and its velocity is modulated such that it stably touches the surface in (c). In (e), the robot reaches x^l. Accordingly, the robot leaves the surface in (f).

5.4.2 Modulation under perturbations

The second part is design to illustrate the capability of the modulation framework in performing Scenario 1 (Figure 5.1a) under perturbations. While the robot is moving from the initial location, a human operator perturbs either the robot or the surface. Perturbations on the robot is applied on its end-effector. Due to the closed loop implementation of (5.2.1), the robot is compliant to the operator. Hence, by changing the state of the robot, the modulation function calculates the best next action, in a real-time, with respect to the currant state of the robot. As it can be seen in the video[9] and Figure 5.9, while the robot is perturbed, the modulation function modifies the motion of the robot such that *Objective 1*, *Objective 2* and *Objective 3*.a are achieved. We then assess the performance of the controller in a dynamically changing environment. Once the robot started moving, the operator changes the fender's position as well as its orientation (Figure 5.10). Due to the fact that modulation function is

[9] http://lasa.epfl.ch/files/Sina_Mirrazavi_Thesis_videos/Chapter_5/Main.mp4.



(a) 0.0 s. **(b)** 2.3 s. **(c)** 3.0 s.

(d) 9.17 s. **(e)** 11.8 s. **(f)** 20.0 s.

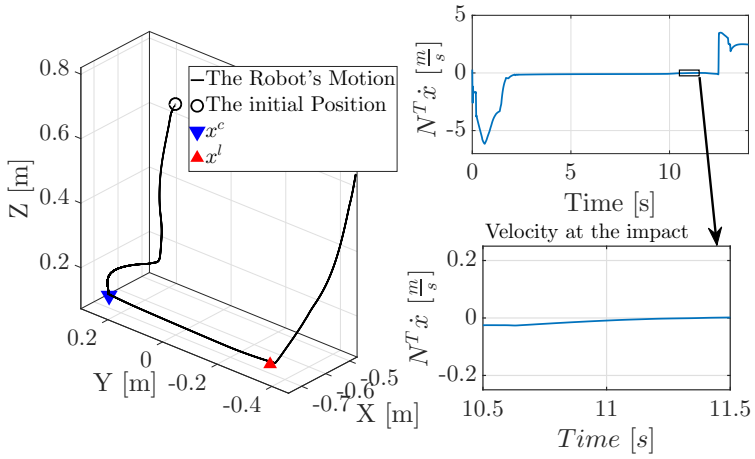
Figure 5.6: Snapshots of the motion of the robot while performing Scenario 2, where the goal of the robot is to touch the surface at x^c and stop on the surface at x^s. In (a), the initial configuration of the robot, x^c and x^s are shown. While the robot is at the transitions region, in (b) and (c), the robot's motion is directed towards x^c and its velocity is modulated such that it stably touches the surface in (d). In (f) the robot reaches x^s.

inherently a linear system, it is computationally efficient. Hence, it can instantaneously modify the robot's motion wrt. the current state of the surface.

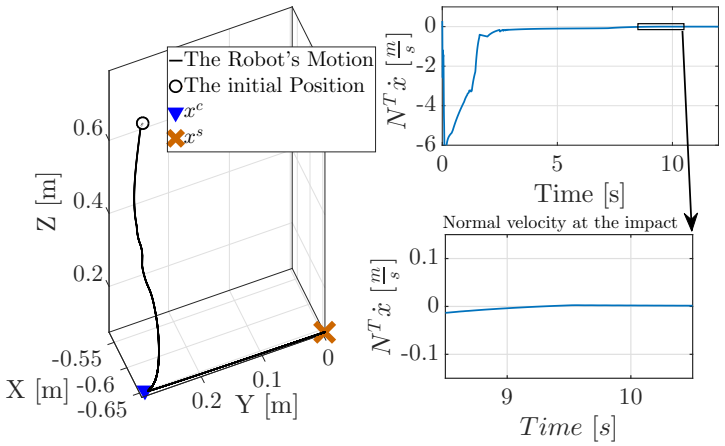
5.4.3 Modulation under uncertainties

In the third part, we assess the performance of the controller in an uncertain environment while performing Scenario 1 (Figure 5.1a). Uncertainties are modeled as a random noise on the location of the surface, where $\boldsymbol{\eta} = 0.15m$. As $\delta_{\dot{x}} = -0.1\frac{m}{s}$ and $N^T x_0 = \rho = 0.2$ and $N^T \dot{x}_{max} = -4\frac{m}{s}$, based on (5.3.16), $\nu = 0.1$ and $\omega = 78$. The experiment was repeated 30 times, see the accompanying video[10] and Table 5.3. In all the 30 trials, the impacts were stable. However, as expected, the robot does not exactly contact the surface at x^c. Moreover, as a force/tactile sensor has not been used, there is no way to realize that a contact has happened. Hence, in 28 out of 30 cases, the robot does not slide on surface, after the contact, to reach x^l. In other two cases, η was approximately

[10] http://lasa.epfl.ch/files/Sina_Mirrazavi_Thesis_videos/Chapter_5/Systematic_Assessment.mp4.



(a) Scenario 1



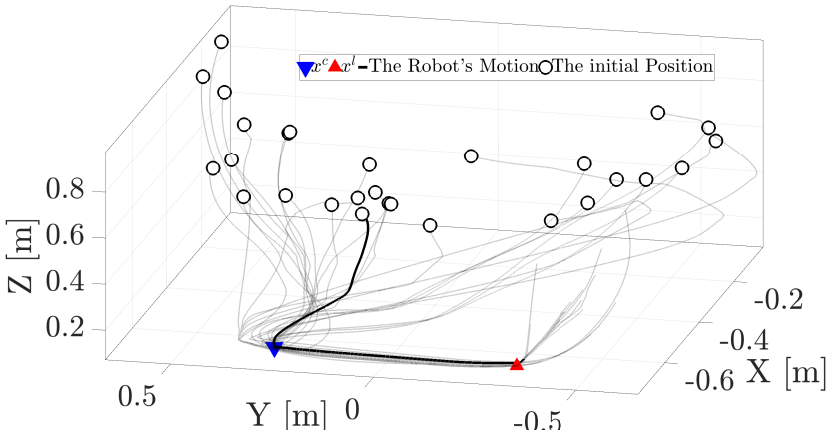
(b) Scenario 2

Figure 5.7: The position of the end-effector, generated by (5.2.1), x^c and $x^l(x^s)$ are shown in the left figures. $\delta_{\dot{x}} = -0.1\frac{m}{s}$. The normal velocity of the robot at the impact region is illustrated in the bottom right figures.

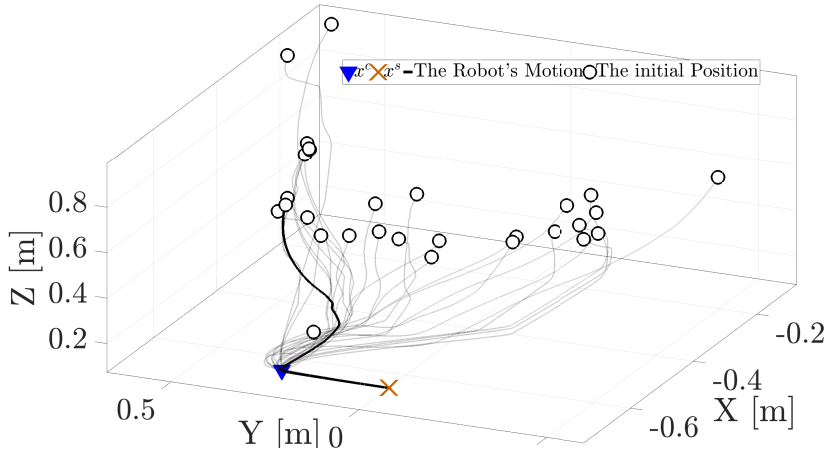
0. Snapshots of the scenario where $\eta = 0.15$ are illustrated in Figure 5.11.

5.5 Conclusion

In this chapter, we proposed an actively compliant control architecture for establishing an stable contact with a surface. The proposed controller consists of locally modulating the robot's motion during non-contact/contact transitions. By using this, the robot reduces its velocity to a certain threshold before coming into contact with the surface such that the post-contact velocity becomes zero; i.e., the impact is being stable and the robot does not bounce on the surface. Furthermore, by modulating the motion of the robot in the tangential directions,



(a) Scenario 1



(b) Scenario 2

Figure 5.8: Spatial variational of the initial, contact and leaving/stopping points. Experimental results verify that the contact is stable and the robot touches the surface at the desired points.

we showed that the contact location can be specified. Moreover, while the robot slides on the surface, it can either leave or stop on the surface at the desired departure or stop points, respectively.

Since the stability of the impact is ensured by bounding the pre-impact velocity at the kinematic level, a careful reader might wonder: why not directly control the impact force. The answer is that in order to directly control the impact force, one need to control the motion of the robot at the dynamic level, which requires manipulators with a high-performance model-based torque controller. It is noteworthy that by controlling the pre-impact velocity, we indirectly control the impact force; i.e., a zero pre-impact velocity results in a zero impact force.

Throughout the proofs, we assume that x^* is a fixed target. However, in two

Table 5.3: The details of the systematically validating the performance of the controller in an uncertain environment. $x^c = [-0.74,\ 0.27,\ 0.05]^T$ and $x^l = [-0.68,\ -0.32,\ 0.07]^T$. $\delta_{\dot{x}}$ is $-0.1\frac{m}{s}$ and $\rho = 0.2m$.

	Scenario 1
Initial position	$[-0.5 \pm 0.1 \ \ -0.1 \pm 0.3 \ \ 0.8 \pm 0.2]$
η	0.06 ± 0.04
Contact Error	0.11 ± 0.13
Pre-contact velocity	0.06 ± 0.05
Pre-transition velocity	1.3 ± 0.13

cases it is not constant. In the first case, x^* is changed wrt. the state of the robot in Proposition 3 and 4. Nevertheless, this does not affect the performance of the system for two reasons as changing x^* based on (5.3.8) or (5.3.9) does not affect the motion of the robot in the normal direction. Hence, the switch between $0 < N^T x$ and $N^T x = 0$ happens only once. In the second case, x^* changes while the surface is perturbed. In this case, as the modulation function is very fast to compute and its convergence rate is faster than the update rate, it can properly react, in real-time, to the perturbations; see Section 5.4.2.

As the sole knowledge of the surface is its location, any inaccuracies in the position measurements deteriorate the performance of the controller. To address this, we present Proposition 5 to improve the robustness of the system in face of uncertainties in the location of the surface. This, however, fails in identifying the true location of the surface once the robot contacts the surface and hence it would not be possible to correctly switch between $0 < N^T x$ and $N^T x = 0$ constraints. One can easily address this problem by using force or tactile sensors.

As a final note, even though, the performance of the system is successfully evaluated on a bumpy surface, in this chapter, the contact surface is assumed planar. One way to improve the framework can be considering convex/non-convex shaped surfaces. Another potential direction to improve the performance of the proposed system would be to integrate the proposed controller with the impedance varying controllers. In this way, the modulated DS is capable of generating the desired motion of the robot, as well as different desired impedance behaviors defined by local compliant regions in the state space.

(a) 0.0 s. **(b)** 0.96 s. **(c)** 1.29 s. **(d)** 2.75 s.

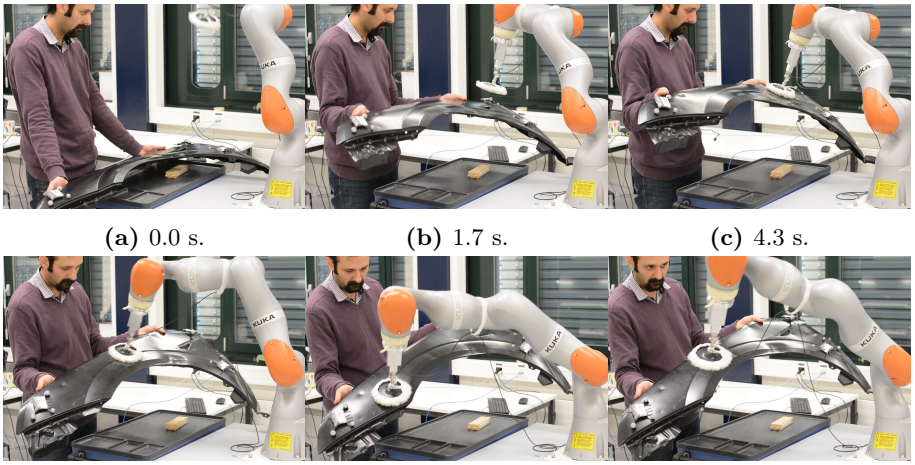
(e) 3.48 s. **(f)** 5.35 s. **(g)** 5.72 s. **(h)** 6.1 s.

(i) 6.47 s. **(j)** 8.31 s. **(k)** 9.01 s. **(l)** 9.74 s.

(m) 11.8 s. **(n)** 14.81 s. **(o)** 15.94 s. **(p)** 16.32 s.

(q) 17.07 s. **(r)** 17.45 s. **(s)** 17.81 s. **(t)** 18.18 s.

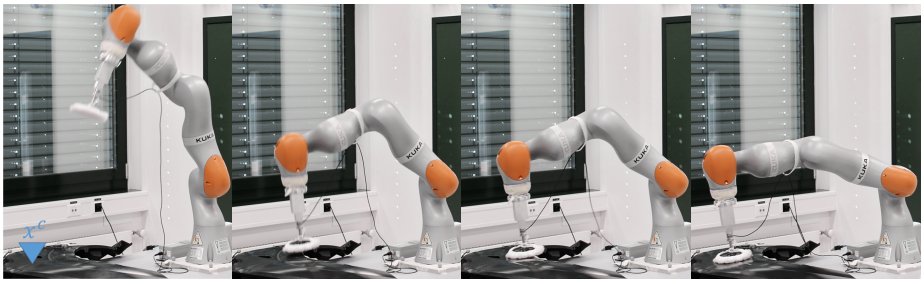
Figure 5.9: Snapshots of the motion of the robot while performing Scenario 1 under perturbations. In (a), the initial configuration of the robot, x^c and x^l are shown.



(a) 0.0 s. **(b)** 1.7 s. **(c)** 4.3 s.

(d) 8.3 s. **(e)** 10.2 s. **(f)** 11.1 s.

Figure 5.10: Snapshots of the motion of the robot in a dynamically changing environment. The location and the orientation of the surface are changing over the motion execution. (a) is the initial location. In (c), the robot contacts the surface at x^c. In (d), the robots slides on the surface while the surface's orientation is changed. In (e), the robot reaches x^l and, consequently, leaves the surface as depicted in (f).



(a) 0.0 s. **(b)** 3.0 s. **(c)** 3.3 s. **(d)** 5.0 s.

Figure 5.11: Snapshots of the motion of the robot in an uncertain environment. As $\eta = 15$ and $\rho = 20$, the effective transition region is $5cm$; the robot enters this region in (b). Hence, the arm does not contact the surface at x^c. However, as the contact was stable, the robot slides on the surface till it reaches x^c in (c)-(d).

Chapter 6

Conclusions

"Perhaps the most important principle for the good algorithm designer is to refuse to be content."

Alfred V. Aho
1941 – present

In this chapter, we summarize our main contributions in this thesis. Furthermore, we discuss the limitations and the potential future directions of the work presented in this thesis.

6.1 Main Contributions

Our key contribution in this thesis lies in presenting motion generators to accomplish interactive tasks, where being compliant to the state of the environment is crucial. Toward this end, we exploit the properties of autonomous dynamical systems and propose motion-control strategies that actively – and in real-time– comply with the state of the environments.

First, in Chapter 3, we have presented a compliant motion-control strategy for catching flying objects. We formulate the controller as a LPV-based dynamical system for generating the end-effector's motion, where the parameters of the LPV system are approximated via Gaussian Mixture Models (GMM) from a set of the fastest kinematically feasible motions of the robot. Furthermore, we propose a computationally efficient optimization algorithm for estimating the parameters from the demonstration. We –theoretically[1] and empirically– show that the arm driven by the proposed controller intercepts the object at the desired location, at the right time, and with the desired velocity, specifically a velocity aligned with that of the object. Aligning the velocity of the arm with the object at the interception results in having the arm move with the object for a short period of time. This improves the robustness of the system against temporal and spatial uncertainties. Moreover, to maximize the softness at the

[1] The stability and the convergence proofs were developed collaboratively with Mahdi Khoramshahi, currently a PhD student at LASA.

interception subject to the kinematic constraints of the robot, we suggest a closed-loop optimal-control problem.

Second, in Chapter 4, we focus on generating the motion of a multi-arm system such that the robots reach, in coordination with each-other, for a moving object. To accomplish this scenario, the motion of each arm must comply with the motion of the other arms, in both task and joint spaces, and with the motion of the object. We propose a virtual-object-based dynamical system to coordinate the motion of the arms with each other and the resultant motion of the arms with that of the object. Furthermore, with the purpose of coordinating the arm motions in the joint-space level, we propose a centralized inverse kinematic solver that is formulated as a quadratic program problem subject to kinematics and collision-avoidance constraints. We theoretically show that the proposed controllers are able to satisfy the objectives of the task. Moreover, the performance of the controller is systematically validated by sets of real-world experiments[2].

Third, in Chapter 5, we address the need for control algorithms when the robotic manipulators come into a contact with a rigid surface. To stably transition from a free motion region to a contact region, the pre-impact velocity of the robot should be modified such that the post-impact velocity becomes zero, i.e., the robot does not bounce on the surface. To this end, we propose a local modulation function for adapting the nominal motion of the robot before entering into contact so that its velocity aligns with the surface. Furthermore, by modulating the motion of the robot in the tangential directions to the surface, we show that the contact location can be specified. Moreover, when the robot slides on the surface, it can either leave or stop on the surface at the desired departure or stop points, respectively.

6.2 Limitations and Future Work

At the end of each chapter, we discuss the limitations and drawbacks of the corresponding frameworks. In this section, we summarize them and elaborate more on the important limitations, as well as possible working directions to address these limitations.

Joint-space task-oriented dynamical Systems

In Chapter 3 and Chapter 5, we have defined the motion of the robot at the task-space level. We assume that there is an accurate and computationally efficient inverse kinematic solver that maps the motion of the robot to the corresponding joints' motions. However, as presented in Section 3.5, it is not always true in real-world scenarios. This limitation is, in particular, critical in fast mo-

[2]The real-world experiments were developed and conducted collaboratively with Nadia Figueroa, currently PhD student at LASA.

tions, as small deviations from the desired path result in situations where the object is hit by the undesired parts of the hands/grippers and bounces away. One way to address this problem is to integrate the "inverse kinematic (IK) solver" and the "motion generator" blocks into a unified block, the proposed approach is capable of generating motions that satisfy the kinematic constraints of the robot.

Force control

In this thesis, we have addressed the need for control architectures for accomplishing interactive scenarios when the mechanical compliance is negligible in comparison with the interaction forces. The proposed control architectures generate the desired robot actions, based on the current state of the environment which can be the object (in Chapter 3), the object and the other robots (in Chapter 4) or the contact surface (in Chapter 5). There are two limitations inherent to this assumption: (i) the state of the environment must be accurately measurable, (ii) accurately controlling interaction forces is not possible.

As discussed in Section 3.5 and Section 4.6, to track the object, all the markers must be visible to the cameras. However, the object's tracking was occasionally obscured partly when the object was very close to the arms, hence covered by them. This is also true in Chapter 5, where the contact surface is approximated with a plane where uncertainties in the surface orientation could not be handled and they directly deteriorate the performance.

Apart from uncertainties, in Chapter 4, we control the motion of the arm from the initial condition (palm open, robots far from the object) to the point when the arms reach the object and the fingers are about to close on the object. Hence, there are no interaction forces (which would arise once the robots are in contact with the object). Once the fingers close on the object, the robot-object system becomes a closed kinematic chain. In this case, devising an appropriate force controller is necessary to coordinate the robots. It is also true, in Chapter 5, where the forces exerted on the contact surface are not controlled.

One potential solution would be to integrate force-feedback into the motion generators, i.e., admittance control architectures. In this way, one can define the desired reference trajectory of the robot in accordance to the measured forces. However, this requires the availability of force sensors that can be mounted on the end-effectors and compensating for all the measurements that are not caused by the interaction with the environment.

Another potential solution would be the unification of motion generators and the impedance-varying controllers. In this way, the DS is capable of generating the desired motion of the robot, as well as different desired impedance behav-

iors defined by local compliant regions in the state space. Hence, the interacting force can be controlled by changing the impedance behavior of the robot.

Tailored motion generators

In all the three studied scenarios, we have designed the controllers, based on the tasks' requirements that are defined by the authors. Hence, the controllers are capable of fulfilling only these specific objectives in the specific tasks. Even though by tailoring the control laws, we theoretically and empirically show that (under certain assumptions) the objectives can be realized, tailoring control laws has one inherent shortcoming: the task's objectives must be precisely definable.

This concern is less problematic when the task is simple and can be decomposed into a sequence of goal-oriented sub-tasks where each of them can be accomplish by designing a control law. Consequently, the entire task can be accomplished by switching between these controllers. However, this issue is more problematic when the task description is vague, e.g., cook a good pasta or gasp an unknown shape object.

One potential solution to this problem is Inverse Reinforcement Learning where the reward/objective functions are approximated from expert demonstrations (Zhifei and Joo, 2012). Hence, there is no need to manually specify the objectives. Even though this might seem to be a solution for learning complex behavior problems, it comes with a challenge: approximating the reward/objective functions from a set of demonstrations is an ill-posed problem as any demonstration might seem optimal under a constant reward.

Communication delays and Dynamics of the robot

Throughout the thesis, we have assumed that (i) the delays in the communication channels are zero (II) there is a low-level controller at the position-level, which fully compensates for the dynamics of the robot + tools mounted on top of it. However, these are not true in the real-world experiments and we spent a considerable amount of time dealing with them. The delays in closed-loop systems can cause one main problem: the robot works much slower than its nominal speed. Hence, in Chapter 3, the DS works in a closed-loop mode with the IK solver and not with the real-robot. The second assumption can change the desired behavior of the system once the payload of the robot changes. In this case, the payload acts as an external perturbation, which deteriorates the performance of the controller.

One potential way to address these issues, is to identify the dynamics of the robot and to propose dynamical systems that encapsulate the low-level controller of the robots so that the dynamical limitations of the robot can be handled as well. By this, the motion of the arm can directly be translated into joint torques, and any changes in the payload of the robot can be seen and properly addressed.

Another way to address these issues is to adapt the parameters of the motion

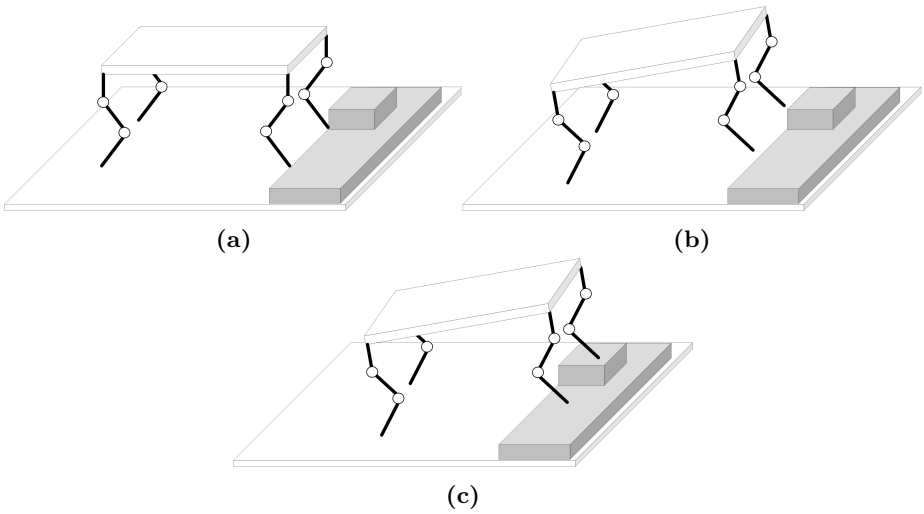


Figure 6.1: Illustration of a scenario where a quadruped robotic platform is galloping on an uneven floor. In this case, each leg of a quadruped robotic platform, in synchronization with the rest of the legs, should touch the floor at the right place, otherwise, the robot might fall.

control law with respect to the tracking error. By this, the identification of the dynamics of the robot is not required and the control law is, incrementally, redefined such that it compensates for the communication delays and the dynamics uncertainties.

Applications to other domains

Even though the proposed control architectures are implemented solely on manipulators, their usages are not limited to these robotic platforms. For example, in Chapter 3 and Chapter 4, we propose motion generators by which a single or multi-robotic arms can intercept a reference trajectory on time at a desired point. By defining this reference trajectory as a limit cycle, one can use the same framework to generate the motion of a quadruped robotic platform, where each leg, in synchronization with the rest of the legs, should touch the floor at the right place, e.g., see Fig. 6.1. The proposed modulation framework, in Chapter 5, can be used in locomotion scenarios to make sure that the steps could be stable and the impact force is minimized.

Another application of the proposed motion generator, in Chapter 4, can be the modeling of the motion of human arms. There has been a number of interesting hypotheses and theories that explain how the human arms are controlled and how the motions of two arms are related. It has been generally accepted in the literature that the motions of two arms are closely coordinated (Tayler and Davids, 1997) and the coordination of hand movements is controlled, optimized with respect to the task-specific goals (Todorov and Jordan, 2002). These two can be achieved by introducing the notion of the virtual object that provides coordination within the arms and with the task-specific goals: intercepting the

object at the desired points.

Appendices

Appendix A

Appendices for chapter 3

A.1 Proof of Theorem 1

Based on (3.3.8), (3.3.5) can be re-written as follows:

$$\begin{aligned}
&\begin{bmatrix} \dot{\xi}(t) - \gamma(t)\dot{\xi}^O(t) - \dot{\gamma}(t)\xi^O(t) \\ \ddot{\xi}(t) - \gamma(t)\ddot{\xi}^O(t) - \ddot{\gamma}(t)\xi^O(t) - \dot{\gamma}(t)\dot{\xi}^O(t) \end{bmatrix} \\
&= \begin{bmatrix} 0_{D\times D} & I_{D\times D} \\ \mathbf{A}_1(\theta) & \mathbf{A}_2(\theta) \end{bmatrix} \begin{bmatrix} \xi(t) - \gamma(t)\xi^O(t) \\ \dot{\xi}(t) - \gamma(t)\dot{\xi}^O(t) - \dot{\gamma}(t)\xi^O(t) \end{bmatrix} \\
&= \sum_{k=1}^{K} \theta_k \begin{bmatrix} 0_{D\times D} & I_{D\times D} \\ A_2^k & A_1^k \end{bmatrix} \begin{bmatrix} \xi(t) - \gamma(t)\xi^O(t) \\ \dot{\xi}(t) - \gamma(t)\dot{\xi}^O(t) - \dot{\gamma}(t)\xi^O(t) \end{bmatrix}
\end{aligned} \tag{A.1.1}$$

Where, $0_{D\times D} \in \mathbb{R}^{D\times D}$ and $I_{D\times D} \in \mathbb{R}^{D\times D}$ are zero and identical matrices, respectively. We propose a Lyapunov function as follows:

$$V = \frac{1}{2} \begin{bmatrix} \xi(t) - \gamma(t)\xi^O(t) \\ \dot{\xi}(t) - \gamma(t)\dot{\xi}^O(t) - \dot{\gamma}(t)\xi^O(t) \end{bmatrix}^T P \begin{bmatrix} \xi(t) - \gamma(t)\xi^O(t) \\ \dot{\xi}(t) - \gamma(t)\dot{\xi}^O(t) - \dot{\gamma}(t)\xi^O(t) \end{bmatrix} \tag{A.1.2}$$

As P is positive definite, V is positive definite, radially unbounded, continuous and continuously differentiable. Substituting (A.1.1) into the derivative of

V with respect to time yields:

$$\begin{aligned}
\dot{V} &= \frac{dV}{dt} \\
&= \sum_{k=1}^{K} \underbrace{\theta_k}_{>0} \begin{bmatrix} \xi(t) - \gamma(t)\xi^O(t) \\ \dot{\xi}(t) - \gamma(t)\dot{\xi}^O(t) - \dot{\gamma}(t)\xi^O(t) \end{bmatrix}^T \Bigg(\\
&\qquad \underbrace{\begin{bmatrix} 0_{D\times D} & I_{D\times D} \\ A_2^k & A_1^k \end{bmatrix}^T P + P \begin{bmatrix} 0_{D\times D} & I_{D\times D} \\ A_2^k & A_1^k \end{bmatrix}}_{\prec 0} \\
&\qquad\qquad \Bigg) \begin{bmatrix} \xi(t) - \gamma(t)\xi^O(t) \\ \dot{\xi}(t) - \gamma(t)\dot{\xi}^O(t) - \dot{\gamma}(t)\xi^O(t) \end{bmatrix} \\
&\leq 0
\end{aligned} \tag{A.1.3}$$

Therefore, dynamical system (3.3.5) is globally stable; i.e., ξ and $\dot{\xi}$ are bounded as ξ^O, $\dot{\xi}^O$, γ and $\dot{\gamma}$ are bounded. Since $\ddot{V}$ is finite, Barbalet's lemma indicates that the attractor is globally asymptotically stable; i.e:

$$\lim_{t\to\infty} \|\dot{\xi}(t) - (\gamma\dot{\xi}^O(t) + \dot{\gamma}(t)\xi^O(t))\| = 0, \qquad \lim_{t\to\infty} \|\xi(t) - \gamma(t)\xi^O(t)\| = 0 \tag{A.1.4}$$

■

To conclude, Theorem 1 is proved.

Appendix B

Appendices for Chapter 4

B.1 Proof of Theorem 2, Part A

We propose a Lyapunov function

$$V = \frac{1}{2}(x_i^R(t) - \tau_{c_i}(t)x_i^V(t) + (\tau_{c_i}(t) - 1)x_i^d)^T P_i^R \\ (x_i^R(t) - \tau_{c_i}(t)x_i^V(t) + (\tau_{c_i}(t) - 1)x_i^d) \tag{B.1.1}$$

V is positive definite, radially unbounded, continuous, and continuously differentiable. The derivative of V with respect to time is

$$\begin{aligned} \dot{V} = \frac{dV}{dt} =& \frac{1}{2}\Big(\big((\dot{x}_i^R(t) - \dot{\tau}_{c_i}(t)x_i^V(t) - \tau_{c_i}(t)\dot{x}_i^V(t) + \\ & \dot{\tau}_{c_i}(t)x_i^d)^T P_i^R(x_i^R(t) - \tau_{c_i}(t)x_i^V(t) + \\ & (\tau_{c_i}(t) - 1)x_i^d)\big) + \big(x_i^R(t) - \tau_{c_i}(t)x_i^V(t) + \\ & (\tau_{c_i}(t) - 1)x_i^d)^T P_i^R(\dot{x}_i^R(t) - \dot{\tau}_{c_i}(t)x_i^V(t) \\ & - \tau_{c_i}(t)\dot{x}_i^V(t) + \dot{\tau}_{c_i}(t)x_i^d)\big)\Big) \end{aligned} \tag{B.1.2}$$

By substituting (4.4.1) into (B.1.2), we have:

$$
\begin{aligned}
\dot{V} =& \frac{1}{2}\Big(\big(\mathbf{A}_i(\theta_i)(x_i^R - x_i^d - \tau_{c_i}(x_i^V - x_i^d))\big)^T P_i^R \\
& (x_i^R(t) - \tau_{c_i}(t)x_i^V(t) + (\tau_{c_i}(t) - 1)x_i^d)\big) + \\
& \big(x_i^R(t) - \tau_{c_i}(t)x_i^V(t) + (\tau_{c_i}(t) - 1)x_i^d\big)^T P_i^R \\
& (\mathbf{A}_i(\theta_i)(x_i^R - x_i^d - \tau_{c_i}(x_i^V - x_i^d)))\Big) \\
=& \frac{1}{2}\Big((x_i^R - x_i^d - \tau_{c_i}(x_i^V - x_i^d))^T \sum_{k=1}^{d_{s_i}} \theta_{ik} A_{ik}^T P_i^R \\
& (x_i^R - x_i^d - \tau_{c_i}(x_i^V - x_i^d))+ \\
& \big(x_i^R - x_i^d - \tau_{c_i}(x_i^V - x_i^d)\big)^T P_i^R \\
& (\sum_{k=1}^{d_{s_i}} \theta_{ik} A_{ik}^T (x_i^R - x_i^d - \tau_{c_i}(x_i^V - x_i^d)))\Big) \\
=& (x_i^R - x_i^d - \tau_{c_i}(x_i^V - x_i^d))^T \\
& \sum_{k=1}^{d_{s_i}} \underbrace{\theta_{ik}}_{>0} \Big(\underbrace{A_{ik}^T P_i^R + P_i^R A_{ik}}_{\prec -Q_i^R} \Big) \\
& (x_i^R - x_i^d - \tau_{c_i}(x_i^V - x_i^d)) \\
& \leq 0
\end{aligned}
\tag{B.1.3}
$$

Therefore, dynamical system (4.4.1) is globally stable; i.e., $x_i^R(t) - x_i^d - \tau_{c_i}(x_i^V - x_i^d))x_i^d$ and its time derivative is bounded. Since $\ddot{V}$ is finite, Barbalat's lemma Khalil (2002) indicates that the attractor is globally asymptotically stable; i.e:

$$
\lim_{t\to\infty} \|x_i^R(t) - \tau_{c_i}(t)x_i^V(t) + (\tau_{c_i}(t) - 1)x_i^d\| = 0 \tag{B.1.4}
$$

■, c.q.f.d.

B.2 Proof of Theorem 2, Part B

Consider the following storage function:

$$
V_i = \frac{1}{2}(x_i^R)^T P_i^R (x_i^R) \quad \forall i \in \{1, \ldots, N_R\} \tag{B.2.1}
$$

Clearly (B.2.1) is positive definite, radially unbounded, continuous and continuously differentiable. To simplify the notations, in this section, we consider

$\mathbf{Y}_i = P_i^R x_i^R$ and $\mathbf{U}_i = \dot{\tau}_{c_i}(x_i^V - x_i^d) + \tau_{c_i}\dot{x}_i^V - \mathbf{A}_i(.)(x_i^d + \tau_{c_i}(x_i^V - x_i^d))$ as the output and the input of (4.4.1). To prove the passivity of (B.2.1), we need to show that

$$\frac{dV}{dt} + \psi(.) \leq \mathbf{U}_i{}^T \mathbf{Y}_i \qquad \exists \psi(.),\ 0 \leq \psi(.) \tag{B.2.2}$$

The derivative of V with respect to time is as follow:

$$\begin{aligned}
\dot{V} &= \frac{1}{2}(\dot{x}_i^R)^T P_i^R (x_i^R) + \frac{1}{2}(x_i^R)^T P_i^R (\dot{x}_i^R) \\
&= \frac{1}{2}\Bigg(\dot{\tau}_{c_i}(x_i^V - x_i^d) + \tau_{c_i}\dot{x}_i^V + \\
&\sum_{k=1}^{d_{s_i}} \theta_{ik} A_{ik}(x_i^R - x_i^d - \tau_{c_i}(x_i^V - x_i^d))\Bigg)^T P_i^R(x_i^R) + \\
&\frac{1}{2}(x_i^R)^T P_i^R \Bigg(\dot{\tau}_{c_i}(x_i^V - x_i^d) + \tau_{c_i}\dot{x}_i^V + \\
&\sum_{k=1}^{d_{s_i}} \theta_{ik} A_{ik}(x_i^R - x_i^d - \tau_{c_i}(x_i^V - x_i^d))\Bigg) \\
&= (x_i^R)^T \sum_{k=1}^{d_{s_i}} \underbrace{\theta_{ik}}_{0\geq} \underbrace{(A_{ik}^T P_i^R + P_i^R A_{ik})}_{\prec -Q_i^R} x_i^R + \mathbf{U}_i^T \mathbf{Y}_i
\end{aligned} \tag{B.2.3}$$

By defining $\psi(.) = -(x_i^R)^T \sum_{k=1}^{d_{s_i}} \theta_{ik}(A_{ik}^T P_i^R + P_i^R A_{ik}) x_i^R$, (B.2.2) is satisfied. Furthermore, as the memoryless system $x_i^R = (P_i^R)^{-1}\mathbf{Y}_i$ is passive, the dynamical system given by (4.4.1) is passive when $\dot{\tau}_{c_i}(x_i^V - x_i^d) + \tau_{c_i}\dot{x}_i^V - \mathbf{A}_i(\theta_i(x_i^R))(x_i^d \tau_{c_i}(x_i^V - x_i^d))$ and $x_i^R(t)$ are the input the output, respectively. ■, c.q.f.d.

B.3 Proof of Theorem 3, Part A

As $\dot{x}_i^V = \dot{x}^V, \forall i \in \{1, \ldots, N_R\}$, (4.4.10) can be written as:

$$\dot{x}^V(t) = \gamma\dot{x}^O + \dot{\gamma}x^O + A^V(x^V - \gamma x^O) \tag{B.3.1}$$

We propose a Lyapunov function as follows:

$$V = \frac{1}{2}(x^V(t) - \gamma x^O)^T P^V (x^V(t) - \gamma x^O) \tag{B.3.2}$$

V is positive definite, radially unbounded, continuous and continuously differentiable. Substituting, (B.3.1) into the derivative of V with respect to time

results in:

$$\begin{aligned}\dot{V} =&\frac{dV}{dt} = \frac{1}{2}\Big((x^V(t) - \gamma x^O)^T P^V A^V (x^V(t) - \gamma x^O) \\ &+ (x^V(t) - \gamma x^O)^T {A^V}^T P^V (x^V(t) - \gamma x^O)\Big) \\ =&(x^V(t) - \gamma x^O)^T \underbrace{(P^V A^V + {A^V}^T P^V)}_{-Q^V}(x^V(t) - \gamma x^O) \\ \leq&0\end{aligned} \tag{B.3.3}$$

Therefore, dynamical system (B.3.1) and (4.4.10) are globally stable; i.e., x^V and $\dot{x}^V$ are bounded as γx^O, $\dot{\gamma} x^O$ and $\gamma \dot{x}^O$ are bounded. Since $\ddot{V}$ is finite, Barbalat's lemma Khalil (2002) indicates that the attractor is globally asymptotically stable; i.e:

$$\begin{aligned}&\lim_{t\to\infty} \|\xi^V(t) - \gamma(t)\xi^O(t)\| = 0 \\ &\lim_{t\to\infty} \|\dot{\xi}^V(t) - (\gamma(t)\dot{\xi}^O(t) + \dot{\gamma}(t)\xi^O(t))\| = 0\end{aligned} \tag{B.3.4}$$

■, c.q.f.d.

B.4 Proof of Theorem 3, Part B

Consider the following storage function:

$$V = \frac{1}{2}(x^V)^T P^V (x^V) \tag{B.4.1}$$

Clearly (B.4.1) is positive definite, radially unbounded, continuous and continuously differentiable. To simplify the notations, in this section, we consider $\mathbf{Y} = P^V x^V$ and $\mathbf{U} = \gamma \dot{x}^O + \dot{\gamma} x^O - A^V \gamma x^O$ as the output and the input of (4.4.10). To prove the passivity of (B.4.1), we need to show that

$$\frac{dV}{dt} + \psi(.) \leq \mathbf{U}^T \mathbf{Y} \qquad \exists \psi(.),\ 0 \leq \psi(.) \tag{B.4.2}$$

The derivative of V with respect to time is as follow:

$$\begin{aligned}
&\dot{V} = \frac{1}{2}(\dot{x}^V)^T P^V (x^V) + \frac{1}{2}(x^V)^T P^V (\dot{x}^V) \\
&= \frac{1}{2}\Big(\gamma \dot{x}^O + \dot{\gamma} x^O + A^V (x^V - \gamma x^O)\Big)^T P^V x^V + \\
&\frac{1}{2}(x^V)^T P^V \Big(\gamma \dot{x}^O + \dot{\gamma} x^O + A^V (x^V - \gamma x^O)\Big) \\
&= (x^V)^T \underbrace{(P^V A^V + (A^V)^T P^V)}_{\prec -Q^V} x^V + \mathbf{U}^T \mathbf{Y}
\end{aligned} \tag{B.4.3}$$

Hence, $\psi(.) = -(x^V)^T (P^V A^V + (A^V)^T P^V) x^V$. Furthermore, as the memoryless system $x^V = (P^V)^{-1} \mathbf{Y}$ is passive, the dynamical system given by (B.3.1) and consequently (4.4.10) are passive when $\gamma \dot{x}^O + \dot{\gamma} x^O - A^V \gamma x^O$and x^V are the input the output, respectively. ■, c.q.f.d.

B.5 Proof of Theorem 4

Consider the following storage function.

$$V = \frac{1}{2}u^T u \tag{B.5.1}$$

(B.5.1) is positive definite, radially unbounded, continuous and continuously differentiable. To simplify the notations, in this section, we consider $\mathbf{Y} = u$ and $\mathbf{U} = (I + M^T)P_\Omega(u - (Mu + b))$ as the output and the input of (4.5.5). To prove the passivity of (4.5.5), we need to show that

$$\frac{dV}{dt} + \psi(u) \leq Y^T \mathbf{U} \qquad \exists \psi(u),\ 0 \leq \psi(u) \tag{B.5.2}$$

The derivative of V with respect to time is as follow:

$$\begin{aligned} &\dot{V} = u^T \dot{u} \\ &\dot{V} = u^T (I + M^T)(P_\Omega(u - (Mu + b) - u) \\ &\dot{V} + u^T (I + M^T)u = u^T (I + M^T)P_\Omega(u - (Mu + b)) \end{aligned} \tag{B.5.3}$$

Hence, $\psi(u) = u^T(I + M^T)u$, which indicates the passivity of (4.5.5). ■, c.q.f.d.

Appendix C

Appendices for Chapter 5

C.1 Proof of Theorem 5

By definition, Q is an orthonormal matrix; i.e $QQ^T = I,\ Q^T = Q^{-1}$. Moreover, as $q_i\ \forall i \in \{1, \dots, d\}$ form an orthonormal basis in $\mathbb{R}^d$, $\forall w \in \mathbb{R}^d$:

$$w = \sum_{i=1}^{d} q_i {q_i}^T w \tag{C.1.1}$$

Substituting (5.3.2) and (5.2.2) into (5.2.1) and multiplying both sides of the resultant equation by $q_1^T = N^T$ and $x^* = x^c$ yields:

$$\begin{aligned}
N^T\ddot{x} =& N^T Q\Lambda Q^{-1} f(x,\dot{x},t) \\
=& \sum_{j=1}^{d} \lambda_{1j}(x,\dot{x}) q_j^T f(x,\dot{x},t) \\
=& \sum_{j=1}^{d} \frac{-2\omega N^T\dot{x} - \omega^2 N^T x}{f(x,\dot{x},t)^T f(x,\dot{x},t)} f(x,\dot{x},t)^T q_j q_j^T f(.) \\
=& \frac{-2\omega N^T\dot{x} - \omega^2 N^T x}{f(.)^T f(.)} f(.)^T \sum_{j=1}^{d} q_j q_j^T f(x,\dot{x},t) \\
=& \left(-2\omega N^T\dot{x} - \omega^2 N^T x\right) \frac{f(x,\dot{x},t)^T f(x,\dot{x},t)}{f(x,\dot{x},t)^T f(x,\dot{x},t)} \\
=& -2\omega N^T\dot{x} - \omega^2 N^T x
\end{aligned} \tag{C.1.2}$$

Which is a second order linear differential equation. The solution of (C.1.2) for a given initial state $\{x_0, \dot{x}_0\}$ is:

$$N^T x = e^{-t\omega}(N^T x_0 + (N^T x_0\omega + N^T\dot{x}_0)t) \tag{C.1.3}$$

Based on (5.3.3), as $\frac{|N^T\dot{x}_0|}{N^T x_0} \leq \omega$ and $0 < N^T x_0$, $0 \leq N^T x_0\omega + N^T\dot{x}_0$. Hence,

(C.1.3) is zero only when t tends to infinity; i.e.

$$\lim_{t\to+\infty} N^T x = 0 \tag{C.1.4}$$

Moreover, the time derivative of (C.1.3) at $t = +\infty$ is zero; i.e.,

$$\begin{aligned}\lim_{t\to+\infty} N^T \dot{x} &= \lim_{t\to+\infty} e^{-t\omega}(N^T\dot{x}_0 - (N^T x_0\omega + N^T\dot{x}_0)\omega t)\\ &= 0\end{aligned} \tag{C.1.5}$$

Hence, the motion generated by 5.2.1 and 5.2.2 with respect to (5.3.2) and (5.3.3), enters the contact surface with zero normal velocity. Hence, *Objective 1* is satisfied.

Similar to (C.1.2), substituting (5.3.4) and (5.2.2) into (5.2.1) and multiplying both sides of the resultant equation by q_i, $\forall i \in \{2,\dots,d\}$ yields:

$$\begin{aligned}q_i^T\ddot{x} =& q_i^T Q\Lambda Q^{-1} f(.)\\ =& \sum_{j=1}^{d} \lambda_{ij}(x,\dot{x}) q_j^T f(.)\\ =& -2\omega {q_i}^T\dot{x} - \omega^2 {q_i}^T(x - x^*)\end{aligned} \tag{C.1.6}$$

Which is a second order linear differential equation. Similar to (C.1.3), the solution of (C.1.6) for a given initial state $\{x_0, \dot{x}_0\}$ converges to ${q_i}^T x^*$ when t tends to infinity; i.e.

$$\lim_{t\to+\infty} \|{q_i}^T x - {q_i}^T x^*\| = 0 \tag{C.1.7}$$

Where, in this theorem, $x^* = x^c$. As (C.1.7) holds $\forall i \in \{2,\dots,D\}$ and the rate of change of (C.1.2) and (C.1.6) are the same, the motion reaches x^c when $t = +\infty$, the dynamical system (5.2.1) and (5.2.2) with respect to (5.3.3) and (5.3.4) contacts the surface at x^c; i.e., *Objective 2* is satisfied. ■

C.2 Proof of Proposition 2

As the modulation function is activated once the robot enters the transition region, $\rho = N^T x_0$. Substituting (5.3.2) and (5.2.2) into (5.2.1) with $\omega = \frac{|N^T \dot{x}_0|}{\rho}$ and multiplying both sides of the resultant equation by N^T yields:

$$\begin{aligned} N^T \ddot{x}_{max} =& -2\omega N^T \dot{x}_0 - \omega^2 N^T x_0 = -2\omega N^T \dot{x}_0 - \omega^2 \rho \\ =& -2\frac{|N^T \dot{x}_0|}{\rho} N^T \dot{x}_0 - (\frac{N^T \dot{x}_0}{\rho})^2 \rho \Rightarrow \\ \rho =& \frac{-2|N^T \dot{x}_0| N^T \dot{x}_0 - (N^T \dot{x}_0)^2}{N^T \ddot{x}_{max}} \end{aligned} \tag{C.2.1}$$

To be safe, we take the upper bound of (C.2.1), i.e., :

$$\rho = \frac{3(N^T \dot{x}_{max})^2}{|N^T \ddot{x}_{max}|} \tag{C.2.2}$$

Substituting (C.2.2) into $\omega = \frac{|N^T \dot{x}_0|}{\rho}$ yields:

$$\omega = |\frac{N^T \ddot{x}_{max}}{3N^T \dot{x}_{max}}| \tag{C.2.3}$$

■

C.3 Proof of Proposition 3

Substituting (5.3.4) and (5.2.2) into (5.2.1) yields an asymptotically stable dynamical system to x^* in the tangential direction to the surface as shown in (C.1.7). Hence, once the robot is in a contact with the surface ($N^T x = 0$), by setting $x^* = x^s$, the robot asymptotically converges to the desired stop location.

C.4 Proof of Proposition 4

x^l is located in the middle of x^c and $2x^l - x^c$.Once the robot is in contact with the surface ($N^T x = 0$), as the motion generated by (5.3.4) moves on a straight line towards $x^* = 2x^l - x^c$[1], it passes x^l. Moreover, the modulation part of (5.3.10) is less than ρ for all the points between x^c and x^l as $\forall \theta \in [0, 1)$ and $x = x^c + \theta(x^l - x^c)$

$$\begin{aligned}
&\left(\rho - (x^l - x^c)^T (x^l - x)\right) e^{-(x^l - x)^T \Sigma^{-1} (x^l - x)} = \\
&\left(\rho - (x^l - x^c)^T (x^l - x^c - \theta(x^l - x^c))\right) e^{-(1-\theta)^2 (x^l - x^c)^T \Sigma^{-1} (x^l - x^c)} = \\
&\left(\rho \underbrace{-(1-\theta)(x^l - x^c)^T (x^l - x^c)}_{0<}\right) \underbrace{e^{-(1-\theta)^2 (x^l - x^c)^T \Sigma^{-1} (x^l - x^c)}}_{<1} < \rho
\end{aligned} \tag{C.4.1}$$

However, if $\lim_{\theta \to 1^+}$, the modulation part of (5.3.10) is greater than ρ. Hence, once the robot passes x^l, the modulation function is deactivated based on (5.2.3) and the nominal dynamical system leaves the surface to converge to x^t. However, it is important to note that if Σ is very small, even thought it leaves the surface at x^l, the motion might not converge to x^t. ■

C.5 Proof of Theorem 6

Substituting (5.3.12)-(5.3.14) and (5.2.2) into (5.2.1) and multiplying both sides of the resultant equation by $q_1^T = N^T$ yields:

$$N^T \ddot{x} = \begin{cases} -\omega \left(N^T \dot{x} - (\delta_{\dot{x}} + \nu)\right) & N^T \dot{x} < \delta_{\dot{x}} \quad \text{(C.5.1)} \\ \dfrac{\omega^2 N^T x + \nu\omega}{\delta_{\dot{x}}} N^T \dot{x} - \omega^2 N^T x & \delta_{\dot{x}} \leq N^T \dot{x} \leq 0 \quad \text{(C.5.2)} \\ -2\omega N^T \dot{x} - \omega^2 N^T x & 0 < N^T \dot{x} \quad \text{(C.5.3)} \end{cases}$$

As shown in Table. 5.1, $N^T \ddot{x}$ defined by (C.5.1)-(C.5.3) is continuous. Based on $N^T \dot{x}_0$, the proof of Theorem 6 needs to be done in three different velocity regions. In the **third** region, $0 < N^T \dot{x}_0$. Hence, based on (C.5.3):

$$N^T \ddot{x} = -2\omega N^T \dot{x} - \omega^2 N^T x \tag{C.5.4}$$

Which is equal to (C.1.2). Hence, as shown in Appendix C.1, as long as $\frac{|N^T \dot{x}_0|}{N^T x_0} \leq \omega$, the motion generated by (C.5.4) reaches the surface with zero

[1] As $\forall i \in \{2, \dots, d\}$, the gains of the second order critically damped DS (C.1.6) are equal, the rate of converges of (C.1.6) is equal in all the directions. Hence, the generated motion by (C.1.6) is a straight line.

velocity. Hence, *Objective 1* is satisfied.

In the **second** region, $\delta_{\dot{x}} \leq N^T\dot{x} \leq 0$. Hence, based on (C.5.2),

$$N^T\ddot{x} = \frac{\omega^2 N^T x + \nu\omega}{\delta_{\dot{x}}} N^T\dot{x} - \omega^2 N^T x \tag{C.5.5}$$

The aforementioned DS yields that if $N^T\dot{x} = \delta_{\dot{x}}$, $N^T\ddot{x} = \omega\nu$, where, based on (5.3.15), $0 < \omega\nu$. This means that once $N^T\dot{x}$ enters this region, it does not cross the velocity boundary at $\delta_{\dot{x}}$; i.e., it does not get less than $\delta_{\dot{x}}$. Moreover, while the robot is above the surface (i.e., $0 < N^T x$) and $N^T\dot{x} = 0$, the normal acceleration is negative; i.e., $N^T\ddot{x} \leq 0$. Hence, the robot's normal velocity can not get higher than zero. To sum up, if $N^T\dot{x}$ is in this region, the robot moves towards the contact surface with the velocity between 0 and $\delta_{\dot{x}}$. Hence, *Objective 1* is satisfied.

In the **first** region, $N^T\dot{x}_0 < \delta_{\dot{x}}$. Hence, based on (C.5.1),

$$N^T\ddot{x} = -\omega(N^T\dot{x} - (\delta_{\dot{x}} + \nu)) \tag{C.5.6}$$

The solution of (C.5.6) for a given initial state $\{x_0, \dot{x}_0\}$ is given by:

$$N^T x(t) = \frac{(\nu + \delta_{\dot{x}} - N^T\dot{x}_0)e^{-\omega^{-1}t} + \omega(N^T x_0 + \nu t + \delta_{\dot{x}} t) + (\nu + \delta_{\dot{x}})t + N^T\dot{x}_0}{\omega} \tag{C.5.7a}$$

$$N^T\dot{x}(t) = (N^T\dot{x}_0 - \nu - \delta_{\dot{x}})e^{-\omega t} + \delta_{\dot{x}} + \nu \tag{C.5.7b}$$

Both (C.5.7a) and (C.5.7b) are monotonic profiles; i.e., if $0 < N^T x_0$ and $N^T\dot{x}_0 < \delta_{\dot{x}}$, (C.5.7a) is monotonically decreasing and (C.5.7b) is monotonically increasing. Hence, based on (C.5.7b), $N^T\dot{x}(t^*) = \delta_{\dot{x}}$ at

$$t^* = -ln(\frac{\nu}{\nu + \delta_{\dot{x}} - N^T\dot{x}_0})\omega^{-1} \tag{C.5.8}$$

Given $N^T\dot{x}_0 < \delta_{\dot{x}}$, substituting (5.3.3), (5.3.15) and (C.5.8) into (C.5.7a) yields

$$\begin{aligned} &N^T x(t^*) \\ &= (\delta_{\dot{x}} + \nu)(-\omega^{-1}\ln(\frac{\nu}{\nu + \delta_{\dot{x}} - N^T\dot{x}_0})) + (N^T\dot{x}_0 - \delta_{\dot{x}})\omega^{-1} + N^T x_0 \\ &= (-(\delta_{\dot{x}} + \nu)\ln(\frac{\nu}{\nu + \delta_{\dot{x}} - N^T\dot{x}_0}) - \delta_{\dot{x}} + N^T\dot{x}_0)\omega^{-1} + N^T x_0 \end{aligned} \tag{C.5.9}$$

As $-\frac{N^T\dot{x}_0}{N^Tx_0} \leq \omega$, $N^Tx(t^*)$ defined by (C.5.9) is bounded as follows:

$$\begin{aligned}&\frac{N^Tx_0\delta_{\dot{x}}}{N^T\dot{x}_0}\left(\ln\left(\frac{\nu}{\nu+\delta_{\dot{x}}-N^T\dot{x}_0}\right)+1\right)\\&+\frac{N^Tx_0\,\nu}{N^T\dot{x}_0}\ln\left(\frac{\nu}{\nu+\delta_{\dot{x}}-N^T\dot{x}_0}\right)\leq N^Tx(t^*)<N^Tx_0\end{aligned}\tag{C.5.10}$$

By defining $0<\frac{\delta_{\dot{x}}-N^T\dot{x}_0}{e^1-1}<\nu$, the lower bound of (C.5.10) will be positive:

$$0<\underbrace{\frac{N^Tx_0\delta_{\dot{x}}}{N^T\dot{x}_0}}_{0<}\underbrace{\left(\ln\left(\frac{\nu}{\nu+\delta_{\dot{x}}-N^T\dot{x}_0}\right)+1\right)}_{0<}+\underbrace{\frac{N^Tx_0\,\nu}{N^T\dot{x}_0}}_{<0}\underbrace{\ln\left(\frac{\nu}{\nu+\delta_{\dot{x}}-N^T\dot{x}_0}\right)}_{<0}\tag{C.5.11}$$

Hence, the robot's normal velocity is $\delta_{\dot{x}}$ before it gets into the contact. Moreover, as $0<N^T\ddot{x}$ if $N^T\dot{x}=\delta_{\dot{x}}$, the robot moves toward the contact surface with $\delta_{\dot{x}}\leq N^T\dot{x}$. To sum up, in all three regions, the proposed modulation function regulates the normal velocity of the robot such that it satisfies (5.3.11) before the robot gets into the contact with the surface. ■

C.6 Proof of Proposition 5

To satisfy (5.3.11) when the location of the surface is uncertain, we need to study the worse scenario; namely when $\eta=\boldsymbol{\eta}$. In this case, to achieve *Objective 1*, the robot's normal velocity must be $\delta_{\dot{x}}$ at $N^Tx=\boldsymbol{\eta}$. Hence, (C.5.9) should lower bounded by $\boldsymbol{\eta}$:

$$(-(\delta_{\dot{x}}+\nu)\ln(\frac{\nu}{\nu+\delta_{\dot{x}}-N^T\dot{x}_0})-\delta_{\dot{x}}+N^T\dot{x}_0)\omega^{-1}+N^Tx_0=\boldsymbol{\eta}\tag{C.6.1}$$

Moreover, Theorem 6 requires that

$$0<\nu\leq-\delta_{\dot{x}},\qquad 0<\omega\tag{C.6.2}$$

Equation (C.6.1) with respect to (C.6.2) does not have a unique solution. Hence, one can use numerical solvers to minimize ω with respect to (C.6.1) and (C.6.2). However, as $\boldsymbol{\eta}<N^Tx_0$ and $N^T\dot{x}_0<\delta_{\dot{x}}$, one can set $\nu=-\delta_{\dot{x}}$ and $\omega=\frac{\delta_{\dot{x}}-N^T\dot{x}_0}{N^Tx_0-\boldsymbol{\eta}}$. This, based on our experience, results in an acceptable performance. To conclude, by defining (5.3.16), $N^T\dot{x}(t)=\delta_{\dot{x}}$ at $N^Tx(t)=\boldsymbol{\eta}$. Hence, $\forall\eta\in[-\boldsymbol{\eta},\boldsymbol{\eta}]$, based on (5.3.13), the robot's velocity at the contact is $\delta_{\dot{x}}\leq N^T\dot{x}$; i.e., (5.3.11) is satisfied. ■

Appendix D

Student Projects Supervised by the Author

In this appendix, the list of the projects which were supervised by the author is provided.

Master Project[1], Spring 2015

Student: Guillaume Clivaz

Title: Controlling a redundant robot for the task space manipulations

Description:
Redundant manipulators have been designed to improve manipulability, flexibility and dexterity. One of the most fundamental issues regarding redundant manipulators is task space manipulation; i.e. finding a joint level command corresponding to the desired end-effector's trajectory. There are two general solutions to solve this problem. The first option involves, directly solving the inverse kinematics problem by finding a feasible joint trajectory given the end-effector's trajectory. An alternative option is to use a task space inverse dynamic controller. The aim of this project is to control a redundant robot in the task space, subject to equality and inequality/bound constraints such as joint limits, angular velocity and acceleration limits. The student will first review literature on related topics, then the proposed control architecture will be implemented in C/C++ in a simulator in Linux environment and then with the real robot, a 7 DOF arm from KUKA.

Master Project[2], Spring 2016

Student: Sylvain Proux

Title: Object-Throwing with a Robot Arm

Description:
Dynamic motions such as catching, juggling, hitting and throwing require accurate motion planning and motor control. In this work, we consider the problem of precisely throwing an object (namely a ball) to a predefined target. Throwing an object precisely is a challenging problem and requires the solution of three challenging problems. First, the release point needs to be determined with respect to the workspace of the robot and the target

[1] This project was co-supervised by Dr. Alireza Karimi and Nadia Figueroa
[2] This project was co-supervised by Nadia Figueroa

position. To address this challenge, a potential direction is estimating a set of trajectories that pass through the desired target position. Then one of these trajectories is chosen to provide a release point (position and velocity) which satisfies the kinematic constraints of the robot. The second challenge is coordinating hand and arm motions. A throwing motion needs to coordinate the arm motion with the hand opening such that the fingers open at a desired point. This motion is modeled through a reverse coupling of the arm and finger motions, where the fingers are closed at the beginning of the motion and then will open at the desired release point. The third challenge is generating a throwing motion. To address this problem, a DS (Dynamical System) needs to be devised to generate a throwing motion from the initial point to the release point with respect to the robot constraints. The throwing motion will be implemented and tested on a 7DOF KUKA LWR robot arm.

Master Project[3], Spring 2017

Student: Camille Lechot

Title: Smart Adaptive Control of Venetian Blinds and Electric Lighting System based on novel High Dynamic Range (HDR) Vision Sensor

Description:
Smart control of the dynamic façade and electric lighting system of an occupied building is a challenging task that requires considering numerous factors such as façade geometry, sun shading characteristics, occupant's visual and thermal comfort. Moreover, the one-fit-all solutions have revealed serious deficiencies with regard to user acceptance of the technology. Well-designed sun control and shading system can dramatically reduce building peak heat gain and cooling requirement, improve the natural lighting quality of building interiors and reduce the rejection rate of the automatic system. Recently, a novel approach for assessing and integrating glare indices and non-image forming effects of light in building automation is developed through a calibrated High Dynamic Range (HDR) imaging sensor (Figure 1). Researchers have shown that the human-building interactions (namely expressed "wishes") are the best source of information for personalizing the automation system and adapting it to each occupant's needs. The goal of this project is to develop a human based controller for commanding electric lighting and shadings based on the lightning conditions. To achieve this, at first, the student needs to familiarize him/herself with literatures for control strategies of venetian blinds and electric lighting. Then, based on the data collected from human actions (the wishes), a proper decision making controller for commanding the electric lighting and the shadings needs to be proposed. Finally, the performance of the proposed system are systematically verified, if time permits. Some funding might be at disposal for tests

[3]This project was co-supervised by Ali Motamed

in case of promising preliminary outcomes. This project is carried out in collaboration with LESO-PB. Please contact both responsible for getting more information.

Semester Project[4], Fall 2015

Student: Ben Hattar Benjamin Henri

Title: Implementation of the singularity free inverse kinematic solution for redundant robots

Description:

Redundant manipulators have been designed to improve manipulability, flexibility and dexterity. One of the most fundamental issues regarding redundant manipulators is solving the inverse kinematics problem; i.e. finding a feasible joint trajectory given the end-effector's trajectory. Solutions to the inverse kinematics problem can be formulated to handle joint physical limits, singularities, obstacle avoidance and to optimize various performance criteria, while conducting the primary end-effector motion task. The aim of this project is to implement a singularity-free inverse kinematics solution for a 7 Degree of Freedom –DoF– KUKA LWR, subject to equality and inequality/bound constraints such as joint limits, angular velocity and acceleration limits. The student will first implement this in C/C++ in a simulator in linux environment and then with the real robot, a 7 DOF arm from KUKA.

Semester Project[5], Fall 2015

Student: Alberto Arrighi

Title: Studying the arms motions in the catching a flying object scenario

Description:

Dynamic motions such as catching, hitting and throwing require accurate trajectory and motor control generation. Humans can instantly execute these motions with excellent accuracy and speed. For example, the problem of catching an object requires ?getting the hands to the right place at the right time?. To successfully catch an object, humans move their hands in coordination to intercept and stop the object. The aim of this project is to study the motion of human hands when catching a large object with both hands. This project consists of two main phases. In the first phase, the student should record the kinematic of hand motion and build a data set of typical postures of arms, hands and fingers and the object. The motions are recorded by using the Optitrack motion capture system, available in the lab. In the second phase, the data are analyzed to find potential correlations

[4]This project was co-supervised by Nadia Figueroa
[5]This project was co-supervised by Nadia Figueroa

between the relative postures of the arms & hands and the position of the object in the person's workspace.

Semester Project[6], Fall 2016
Student: Nadir Benjamin Ramtoula
Title: Motion study for bimanual catching of a flying object

Description:
Bimanual catching of a flying object is an intriguing problem of visuomotor coordination. In robotics, such a task is an interesting problem that has many possible applications (e.g. dynamic human-robot interaction with large objects or catching a falling object in dangerous situations), however, at the same time, it is a challenging problem that involves fast motor control with time constraint based on the perceived visual information. Humans are capable of coordinating the two arms in a very smooth and efficient way – especially if we consider how slow human brains are compared to the computational capabilities of robots these days. It is still largely unknown how humans coordinate their arms and generate desired trajectories in fast adaptive scenarios such as catching a flying ball with both hands. The simple and efficient strategy that humans use in their motions has large potential in the robotic applications where it can be used in designing a fast and effective controller.

The purpose of this study is to analyze the characteristics of the hand trajectories in terms of the temporal constraint that is present in catching a flying object scenario. The goal of this project is to study:

(i) how the remaining time before catching (time to contact, TTC) affects the trajectories of the hands, (ii) whether humans estimate the TTC and utilize it in motion generations, and (iii) how the trajectories of the two hands are coordinated in given TTC.

In order to achieve the goals, the student should perform a set of experiments where the motions of two arms are recorded using a motion capture system for different catching scenarios. The student is expected to get familiarized himself/herself with related topics including human motions in interceptive actions through reading the literature. Once the data is collected from experiments, the student should analyze the data in terms of the trajectory, movement onset time, movement duration, tangential velocity at impact, normal velocity to the object, and etc. The student should analyze the effect of the TTC on the trajectories of hands in the aspects mentioned above. One possible way to explore is to compare the trajectories for catching a moving object with the trajectories for reaching a stationary object.

[6]This project was co-supervised by Kevin Gonyop Kim

Internship Project, Spring-Summer 2015
Student: Edouard Lagrue
Title: Bi-manual Catching in flight objects

Internship Project, Spring-Summer 2016
Student: Kevin Gonyop Kim
Title: Human-inspired Bi-manual Catching

Internship Project[7], Fall-Winter 2016
Student: Neda Taymourtash
Title: Soft Catching with a Humanoid Robot

Description:
Dynamic motions such as catching, juggling, hitting , and throwing require accurate motion planning and motor control. Catching objects in flight is a particularly challenging task as the time to move toward the object is extremely short, usually lasting about a quarter of a second. The time at impact is even shorter, leaving less than a few milliseconds for the hands to close on the object to secure it tightly in the grip. Soft catching strategy has been proposed as a promising avenue for improving the success rate of catching [1]. The soft catching strategy consists of having the robot move with the object for a short period of time. This leaves more time for the fingers to close on the object and avoids failure due to imprecise control of the time and position at which the hand intercepts the object. The aim of this project is to implement the soft catching algorithm proposed by [1] on the COMAN humanoid robot. The COMAN robot is 95cm tall, weighs 31kg and has 25 DOF. COMAN can walk and balance using inertial sensors in the pelvis and chest, and its series elastic joint design makes it robust against impacts and external disturbances. The student will first familiarize him/herself with the implemented version of the algorithm on the KUKA IIWA robot in C++. Then, the algorithm will be implemented in C/C++ in the COMAN simulator under Linux environment and consequently with the real COMAN robot.

[1] S. S. M. Salehian, M. Khoramshahi and A. Billard, "A Dynamical System Approach for Softly Catching a Flying Object: Theory and Experiment," in IEEE Transactions on Robotics, vol. 32, no. 2, pp. 462-471, April 2016.

[7] This project was co-supervised by Nadia Figueroa

References

F. Aghili. Adaptive control of manipulators forming closed kinematic chain with inaccurate kinematic model. *IEEE/ASME Transactions on Mechatronics,*, 18 (5):1544–1554, Oct 2013.

Juan Manuel Ahuactzin, Kamal Gupta, and Emmanuel Mazer. Manipulation planning for redundant robots: A practical approach. *The International Journal of Robotics Research*, 17(7):731–747, 1998. doi: 10.1177/027836499801700704.

Ibrahim Al-Bluwi, Thierry Siméon, and Juan Cortés. Motion planning algorithms for molecular simulations: A survey. *Computer Science Review*, 6(4): 125–143, 2012.

Alin Albu-Schäffer, Sami Haddadin, Ch Ott, Andreas Stemmer, Thomas Wimböck, and Gerhard Hirzinger. The dlr lightweight robot: design and control concepts for robots in human environments. *Industrial Robot: an international journal*, 34(5):376–385, 2007.

Andrea Alessandretti, A Pedro Aguiar, and Colin N Jones. Trajectory-tracking and path-following controllers for constrained underactuated vehicles using model predictive control. In *Control Conference (ECC), 2013 European*, pages 1371–1376. IEEE, 2013.

N. M. Amato and Y. Wu. A randomized roadmap method for path and manipulation planning. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 113–120 vol.1, Apr 1996. doi: 10.1109/ROBOT.1996.503582.

H. Bai and J. T. Wen. Cooperative load transport: A formation-control perspective. *IEEE Transactions on Robotics*, 26(4):742–750, Aug 2010. ISSN 1552-3098.

Bassam Bamieh and Laura Giarre. Identification of linear parameter varying models. *International journal of robust and nonlinear control*, 12(9):841–853, 2002.

J. Barraquand, B. Langlois, and J. C. Latombe. Numerical potential field techniques for robot path planning. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1012–1017 vol.2, June 1991. doi: 10.1109/ICAR.1991.240539.

D References

Jérôme Barraquand and Jean-Claude Latombe. Robot motion planning: A distributed representation approach. *The International Journal of Robotics Research*, 10(6):628–649, 1991. doi: 10.1177/027836499101000604.

Georg Bätz, Arhan Yaqub, Haiyan Wu, Kolja Kühnlenz, Dirk Wollherr, and Martin Buss. Dynamic manipulation: Nonprehensile ball catching. In *Control & Automation (MED), 2010 18th Mediterranean Conference on*, pages 365–370. IEEE, 2010.

Berthold Bäuml, Oliver Birbach, Thomas Wimböck, Udo Frese, Alexander Dietrich, and Gerd Hirzinger. Catching flying balls with a mobile humanoid: System overview and design considerations. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 513–520. IEEE, 2011.

Bastien Berret, Enrico Chiovetto, Francesco Nori, and Thierry Pozzo. Manifold reaching paradigm: how do we handle target redundancy? *Journal of neurophysiology*, 106 4:2086–102, 2011.

Aude Billard. On the mechanical, cognitive and sociable facets of human compliance and their robotic counterparts. *Robotics and Autonomous Systems*, 88:157 – 164, 2017. ISSN 0921-8890. doi: http://dx.doi.org/10.1016/j.robot.2016.08.030.

Aude G. Billard, Sylvain Calinon, and Rüdiger Dillmann. *Learning from Humans*, pages 1995–2014. Springer International Publishing, Cham, 2016.

M. C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007. ISBN 978-0387310732.

Caroline Blocher, Matteo Saveriano, and Dongheui Lee. Learning stable dynamical systems using contraction theory. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2017 14th International Conference on*, pages 124–129. IEEE, 2017.

RC Bonitz and Tien C Hsia. Internal force-based impedance control for cooperating manipulators. *IEEE Transactions on Robotics and Automation*, 12(1): 78–89, 1996.

Michael Brady. *Robot motion: Planning and control*. MIT press, 1982.

O. Brock and O. Khatib. High-speed navigation using the global dynamic window approach. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 1, pages 341–346 vol.1, 1999. doi: 10.1109/ROBOT.1999.770002.

Oliver Brock, James Kuffner, and Jing Xiao. *Motion for Manipulation Tasks*, pages 615–645. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-30301-5. doi: 10.1007/978-3-540-30301-5_27.

B. Brogliato, S. I. Niculescu, and P. Orhant. On the control of finite-dimensional mechanical systems with unilateral constraints. *IEEE Transactions on Automatic Control*, 42(2):200–215, Feb 1997. ISSN 0018-9286.

Daniel Bullock and Stephen Grossberg. The vite model: A neural command circuit for generating arm and articulator trajectories. *Dynamic patterns in complex systems*, pages 305–326, 1988.

R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research*, 18(6):534–555, 1999.

Giorgio C Buttazzo, Benedetto Allotta, and Felice P Fanizza. Mousebuster: A robot for real-time catching. *IEEE Control Systems*, 14(1):49–56, 1994.

B. Bäuml, T. Wimböck, and G. Hirzinger. Kinematically optimal catching a flying ball with a hand-arm-system. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2592–2599, Oct 2010.

F. Caccavale, P. Chiacchio, A. Marino, and L. Villani. Six-dof impedance control of dual-arm cooperative manipulators. *IEEE/ASME Transactions on Mechatronics*, 13(5):576–586, Oct 2008.

Fabrizio Caccavale and Masaru Uchiyama. Cooperative manipulators. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 701–718. Springer Berlin Heidelberg, 2008.

Fabrizio Caccavale and Masaru Uchiyama. *Springer Handbook of Robotics, Chapter 39, Cooperative Manipulation*, pages 989–1006. Springer International Publishing, Cham, 2016.

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. Special issue on robot learning by observation, demonstration and imitation*, 37(2):286–298, 2007.

Sylvain Calinon, Antonio Pistillo, and Darwin G Caldwell. Encoding the time and space constraints of a task in explicit-duration hidden markov model. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3413–3418. IEEE, 2011.

Heather Carnahan and Bradford J. McFadyen. Visuomotor control when reaching toward and grasping moving targets. *Acta Psychologica*, 92(1):17 – 32, 1996. ISSN 0001-6918.

S. Carpin and G. Pillonetto. Robot motion planning using adaptive random walks. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 3, pages 3809–3814 vol.3, Sept 2003. doi: 10.1109/ROBOT.2003.1242181.

Ralph H Castain and Richard P Paul. An on-line dynamic trajectory generator. *The International Journal of Robotics Research*, 3(1):68–72, 1984.

Liu Chengqing, M. H. Ang, H. Krishnan, and Lim Ser Yong. Virtual obstacle concept for local-minimum-recovery in potential-field based navigation. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pages 983–988 vol.2, 2000. doi: 10.1109/ROBOT.2000.844728.

T. Chettibi, H.E. Lehtihet, M. Haddad, and S. Hanchi. Minimum cost trajectory planning for industrial robots. *European Journal of Mechanics - A/Solids*, 23(4):703 – 715, 2004. ISSN 0997-7538. doi: https://doi.org/10.1016/j.euromechsol.2004.02.006.

Pasquale Chiacchio and Stefano Chiaverini, editors. *Complex Robotic Systems,Chapter 3 Kinematic control of dual-arm systems*, pages 79–97. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-540-40904-5. doi: 10.1007/BFb0035185.

Wonyun Choi and J. C. Latombe. A reactive architecture for planning and executing robot motions with incomplete knowledge. In *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on*, pages 24–29 vol.1, Nov 1991. doi: 10.1109/IROS.1991.174421.

B. Chrètien, A. Escande, and A. Kheddar. Gpu robot motion planning using semi-infinite nonlinear programming. *IEEE Transactions on Parallel and Distributed Systems*, 27(10):2926–2939, Oct 2016. ISSN 1045-9219.

X. Chu, Q. Hu, and J. Zhang. Path planning and collision avoidance for a multi-arm space maneuverable robot. *IEEE Transactions on Aerospace and Electronic Systems*, PP(99):1–1, 2017. ISSN 0018-9251. doi: 10.1109/TAES.2017.2747938.

S. J. Chung and J. J. E. Slotine. Cooperative robot control and concurrent synchronization of lagrangian systems. *IEEE Transactions on Robotics*, 25 (3):686–700, June 2009. ISSN 1552-3098.

P. Cigliano, V. Lippiello, F. Ruggiero, and B. Siciliano. Robotic ball catching with an eye-in-hand single-camera system. *IEEE Transactions on Control Systems Technology*, 23(5):1657–1671, Sept 2015. ISSN 1063-6536.

David C Conner, Alfred A Rizzi, and Howie Choset. Integrated planning and control for convex-bodied nonholonomic systems using local feedback control policies. *Robotics Institute*, page 124, 2006.

D. Constantinescu and E.A. Croft. Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. *Journal of Robotic Systems*, 17(5):233–249, 2000.

John J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1989. ISBN 0201095289.

Elizabeth A Croft, RG Fenton, and B Benhabib. Time-optimal interception of objects moving along predictable paths. In *Assembly and Task Planning, 1995. Proceedings., IEEE International Symposium on*, pages 419–425. IEEE, 1995.

Joost C Dessing, Daniel Bullock, C.(Lieke) E Peper, and Peter J Beek. Prospective control of manual interceptive actions: comparative simulations of extant and new model constructs. *Neural Networks*, 15(2):163 – 179, 2002. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(01)00136-8.

Jörn Diedrichsen and Noreen Dowling. Bimanual coordination as task-dependent linear control policies. *Human movement science*, 28(3):334–347, 2009.

K. Dupree, C. H. Liang, G. Hu, and W. E. Dixon. Adaptive lyapunov-based control of a robot and mass-spring system undergoing an impact collision. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(4):1050–1061, Aug 2008. ISSN 1083-4419.

Economist. Working with robots. our friends electric robotics: A new breed of robots is being designed to collaborate with humans, working alongside them to make them more productive, 2013. URL https://www.economist.com/news/technology-quarterly/21584455-robotics-new-breed-robots-being-designed-collaborate-humans.

Economist. The growth of industrial robots. they mainly build cars and computer chips, for now, 2017a. URL https://www.economist.com/blogs/graphicdetail/2017/03/daily-chart-19?zid=291&ah=906e69ad01d2ee51960100b7fa502595.

Economist. Will robots replace human workers?, 2017b. URL https://www.youtube.com/watch?v=SimCBnbPdP4.

Economist. Robots in the rustbelt. the future lies in automation. factories are upgrading, but still lag far behind the rich world, 2017c. URL https://www.economist.com/news/special-report/21720073-factories-are-upgrading-still-lag-far-behind-rich-world-future-lies.

M. Elbanhawi and M. Simic. Sampling-based robot motion planning: A review. *IEEE Access*, 2:56–77, 2014. ISSN 2169-3536. doi: 10.1109/ACCESS.2014.2302442.

Zlatko Emedi and Alireza Karimi. Fixed-structure LPV Discrete-time Controller Design with Induced l2-Norm and H2 Performance. *International Journal of Contol*, 2015. doi: 10.1080/00207179.2015.1081986.

A. Escande, S. Miossec, and A. Kheddar. Continuous gradient proximity dis-

tance for humanoids free-collision optimized-postures. In *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pages 188–195, Nov 2007.

A. Escande, S. Miossec, M. Benallegue, and A. Kheddar. A strictly convex hull for computing proximity distances with continuous gradients. *IEEE Transactions on Robotics*, 30(3):666–678, June 2014. ISSN 1552-3098.

C. Fang, A. Rocchi, E. M. Hoffman, N. G. Tsagarakis, and D. G. Caldwell. Efficient self-collision avoidance based on focus of interest for humanoid robots. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 1060–1066, Nov 2015.

Timm Faulwasser and Rolf Findeisen. Nonlinear model predictive control for constrained output path following. *IEEE Transactions on Automatic Control*, 61(4):1026–1039, 2016.

J. Fiala, M. Kocvara, and M. Stingl, 2013. URL http://web.mat.bham.ac.uk/kocvara/penlab/.

F. Ficuciello, L. Villani, and B. Siciliano. Variable impedance control of redundant manipulators for intuitive human;robot physical interaction. *IEEE Transactions on Robotics*, 31(4):850–863, Aug 2015. ISSN 1552-3098.

N. Figueroa and A. Billard. Learning complex manipulation tasks from heterogeneous and unstructured demonstrations. In Proceedings of Workshop on Synergies between Learning and Interaction. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2017.

Nadia Figueroa, Ana Lucia Pais Ureche, and Aude Billard. Learning complex sequential tasks from demonstration: A pizza dough rolling case study. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, pages 611–612. IEEE Press, 2016.

D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics Automation Magazine*, 4(1):23–33, Mar 1997. ISSN 1070-9932. doi: 10.1109/100.580977.

H. Frank, D. Barteit, M. Meyer, A. Mittnacht, G. Novak, and S. Mahlknecht. Optimized control methods for capturing flying objects with a cartesian robot. In *2008 IEEE Conference on Robotics, Automation and Mechatronics*, pages 160–165, Sept 2008. doi: 10.1109/RAMECH.2008.4681368.

Udo Frese, B Bauml, Steffen Haidacher, Günter Schreiber, Ingo Schäfer, M Hahnle, and Gerd Hirzinger. Off-the-shelf vision for a robotic ball catcher. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 3, pages 1623–1629. IEEE, 2001.

A. Gams, A. Ude, and J. Morimoto. Accelerating synchronization of movement primitives: Dual-arm discrete-periodic motion of a humanoid robot. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 2754–2760, Sept 2015. doi: 10.1109/IROS.2015.7353755.

JL Garcia-Lopez, I Salgado, and I Chairez. Two-layer dynamic neural field learning law basec on controlled lyapunov functions. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 4496–4503. IEEE, 2017.

S. S. Ge and Y. J. Cui. New potential functions for mobile robot path planning. *IEEE Transactions on Robotics and Automation*, 16(5):615–620, Oct 2000. ISSN 1042-296X. doi: 10.1109/70.880813.

M. Gharbi, J. Cortés, and T. Siméon. Roadmap composition for multi-arm systems path planning. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2471–2476, Oct 2009.

Elena Gribovskaya, S. M. Khansari Zadeh, and Aude Billard. Learning Nonlinear Multivariate Dynamics of Motion in Robotic Manipulators [accepted]. *International Journal of Robotics Research*, 2010. ISSN 0278-3649.

Ludovic Hamon. Virtual reality and programming by demonstration: teaching a robot to grasp a dynamic object by the generalization of human demonstrations. *Presence*, 20(3):241–253, 2011.

D. J. F. Heck, A. Saccon, N. van de Wouw, and H. Nijmeijer. Switched position-force tracking control of a manipulator interacting with a stiff environment. In *2015 American Control Conference (ACC)*, pages 4832–4837, July 2015.

M. Hermann, T. Pentek, and B. Otto. Design principles for industrie 4.0 scenarios. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 3928–3937, Jan 2016.

M. Hersch, F. Guenter, S. Calinon, and Aude Billard. Dynamical system modulation for robot learning via kinesthetic demonstrations. *IEEE Transactions on Robotics*, 24(6):1463–1467, Dec 2008. ISSN 1552-3098.

Micha Hersch and Aude Billard. Reaching with multi-referential dynamical systems. *Autonomous Robots*, 25(1):71–83, Aug 2008.

Neville Hogan. Impedance control: An approach to manipulation. *Journal of dynamic systems, measurement, and control*, 107:17, 1985.

John M Hollerbach. Dynamic scaling of manipulator trajectories. *Journal of Dynamic Systems, Measurement, and Control*, 106(1):102–106, 1984.

W. Hong and J J. E. Slotine. Experiments in hand-eye coordination using active vision. In *Lecture Notes In Control And Information Sciences*, pages 130–139. Springer-Verlag, 1995.

Barbara Hove and Jean-Jacques E Slotine. Experiments in robotic catching. In *American Control Conference, 1991*, pages 380–386. IEEE, 1991.

David Hsu, J-C Latombe, and Rajeev Motwani. Path planning in expansive

configuration spaces. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 3, pages 2719–2726. IEEE, 1997.

David Hsu, Jean-Claude Latombe, and Hanna Kurniawati. On the probabilistic foundations of probabilistic roadmap planning. *The International Journal of Robotics Research*, 25(7):627–643, 2006.

Damir Hujic, Elizabeth A Croft, Gene Zak, Robert G Fenton, James K Mills, and Beno Benhabib. The robotic interception of moving objects in industrial settings: Strategy development and experiment. *IEEE/ASME Transactions On Mechatronics*, 3(3):225–239, 1998.

A. Hussein, M.M. Gaber, E. Elyan, and C. Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys*, 50(2), 2017. doi: 10.1145/3054912.

International Fderation of Robotics IFR. Executive summary world robotics 2016 industrial robots, 2016. URL https://ifr.org/img/uploads/Executive_Summary_WR_Industrial_Robots_20161.pdf.

Auke Jan Ijspeert, Jun Nakanishi, and Stefan Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1398–1403. IEEE, 2002.

Yan-Bin Jia, Matthew T Mason, and Michael A Erdmann. Multiple impacts: A state transition diagram approach. *The International Journal of Robotics Research*, 32(1):84–114, 2013. doi: 10.1177/0278364912461539.

Maolin Jin, Sang Hoon Kang, P. H. Chang, and Eunjeong Lee. Nonlinear bang-bang impact control: A seamless control in all contact modes. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 557–564, April 2005.

M. Jinno, F. Ozaki, T. Yoshimi, K. Tatsuno, M. Takahashi, M. Kanda, Y. Tamada, and S. Nagataki. Development of a force controlled robot for grinding, chamfering and polishing. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 2, pages 1455–1460 vol.2, May 1995. doi: 10.1109/ROBOT.1995.525481.

Steven G Johnson. The NLopt nonlinear-optimization package, 2015. URL http://ab-initio.mit.edu/nlopt.

Ariyan M Kabir, Krishnanand N Kaipa, Jeremy Marvel, and Satyandra K Gupta. Automated planning for robotic cleaning using multiple setups and oscillatory tool motions. *IEEE Transactions on Automation Science and Engineering*, 2017.

M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal. Stomp: Stochastic trajectory optimization for motion planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13,

2011.

F. Kanehiro, E. Yoshida, and K. Yokoi. Efficient reaching motion planning and execution for exploration by humanoid robots. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1911–1916, Oct 2012.

R. Katoh and M. Mori. Control method of biped locomotion giving asymptotic stability of trajectory. *Automatica*, 20(4):405 – 414, 1984. ISSN 0005-1098. doi: http://dx.doi.org/10.1016/0005-1098(84)90099-2.

Lydia E. Kavraki and Steven M. LaValle. *Motion Planning*, pages 109–131. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.

Hassan Khalil. *Nonlinear Systems*. Prentice Hall, 2002. ISBN 0-13-067389-7.

Mohammad Khansari, Ellen Klingbeil, and Oussama Khatib. Adaptive human-inspired compliant contact primitives to perform surface–surface contact under uncertainty. *The International Journal of Robotics Research*, 35(13):1651–1675, 2016. doi: 10.1177/0278364916648389. URL http://ijr.sagepub.com/content/35/13/1651.abstract.

S. M. Khansari-Zadeh and A. Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27 (5):943–957, 2011.

S.-M. Khansari-Zadeh and A. Billard. A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, 32:433–454, 2012. ISSN 0929-5593. 10.1007/s10514-012-9287-y.

S. M. Khansari-Zadeh, K. Kronander, and A. Billard. Learning to play minigolf: A dynamical system-based approach. *Advanced Robotics*, 2012.

Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986.

Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

S. Kim and A. Billard. Estimating the non-linear dynamics of free-flying objects. *Robotics and Autonomous Systems*, 60(9):1108–1122, 2012.

S. Kim, A. Shukla, and Aude Billard. Catching objects in flight. *IEEE Transactions on Robotics*, 30(5):1049–1065, Oct 2014. ISSN 1552-3098. doi: 10.1109/TRO.2014.2316022.

Seungsu Kim, Elena Gribovskaya, and Aude Billard. Learning Motion Dynamics to Catch a Moving Object. In *10th IEEE-RAS International Conference on Humanoid Robots*, 2010.

Jens Kober, Matthew Glisson, and Michael Mistry. Playing catch and juggling with a humanoid robot. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pages 875–881. IEEE, 2012.

Daniel E Koditschek and Elon Rimon. Robot navigation functions on manifolds with boundary. *Advances in Applied Mathematics*, 11(4):412 – 442, 1990. ISSN 0196-8858.

M. Koike, T. Senoo, K. Murakami, and M. Ishikawa. Plastic deformation control based on time-varying impedance adjustment. In *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 106–111, Dec 2016.

Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 1398–1404 vol.2, Apr 1991. doi: 10.1109/ROBOT.1991.131810.

Petar Kormushev, Sylvain Calinon, and Darwin G Caldwell. Robot motor skill coordination with em-based reinforcement learning. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3232–3237. IEEE, 2010.

K. Kronander, S. M. Khansari Zadeh, and A. Billard. New, 2011.

Klas Kronander, Mohammad Khansari, and Aude Billard. Incremental motion learning with locally modulated dynamical systems. *Robotics and Autonomous Systems*, 70:52–62, 2015.

R. Lampariello, D. Nguyen-Tuong, C. Castellini, G. Hirzinger, and J. Peters. Trajectory planning for optimal robot catching in real-time. In *IEEE International Conference on Robotics and Automation*, pages 3719–3726, 2011.

D. Lau, J. Eden, and D. Oetomo. Fluid motion planner for nonholonomic 3-d mobile robots with kinematic constraints. *IEEE Transactions on Robotics*, 31 (6):1537–1547, Dec 2015. ISSN 1552-3098. doi: 10.1109/TRO.2015.2482078.

Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.

Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

Steven M LaValle and James J Kuffner Jr. Rapidly-exploring random trees: Progress and prospects. 2000.

D. Lee and Y. Nakamura. Mimesis scheme using a monocular vision system on a humanoid robot. In *Proceedings 2007 IEEE International Conference*

on Robotics and Automation, pages 2162–2168, April 2007. doi: 10.1109/ROBOT.2007.363641.

DN Lee, DS Young, PE Reddish, S Lough, and TMH Clayton. Visual timing in hitting an accelerating ball. *The Quarterly Journal of Experimental Psychology*, 35(2):333–346, 1983.

Eunjeong Lee, Juyi Park, K. A. Loparo, C. B. Schrader, and Pyung Hun Chang. Bang-bang impact control using hybrid impedance/time-delay control. *IEEE/ASME Transactions on Mechatronics*, 8(2):272–277, June 2003. ISSN 1083-4435.

Jinseok Lee, Yunyoung Nam, Sangjin Hong, and Weduke Cho. New potential functions with random force algorithms using potential field method. *Journal of Intelligent & Robotic Systems*, 66(3):303–319, May 2012. ISSN 1573-0409.

Daniel Leidner, Alexander Dietrich, Michael Beetz, and Alin Albu-Schäffer. Knowledge-enabled parameterization of whole-body control strategies for compliant service robots. *Autonomous Robots*, 40(3):519–536, Mar 2016. ISSN 1573-7527. doi: 10.1007/s10514-015-9523-3.

A. Lemme, K. Neumann, R.F. Reinhart, and J.J. Steil. Neural learning of vector fields for encoding stable dynamical systems. *Neurocomputing*, 141:3–14, 2014.

Andre Lemme, Klaus Neumann, Felix Reinhart, and Jochen J Steil. Neurally imprinted stable vector fields. 2013.

C. Liang, S. Bhasin, K. Dupree, and W. E. Dixon. A force limiting adaptive controller for a robotic system undergoing a non-contact to contact transition. In *2007 46th IEEE Conference on Decision and Control*, pages 3555–3560, Dec 2007.

Nejc Likar, Bojan Nemec, and Leon Žlajpah. Virtual mechanism approach for dual-arm manipulation. *Robotica*, FirstView:1–16, 8 2013. ISSN 1469-8668.

Chunshin Lin, Porong Chang, and J Luh. Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *IEEE Transactions on automatic control*, 28(12):1066–1074, 1983.

Chyi-Yeu Lin and Yi-Pin Chiu. The dsp based catcher robot system with stereo vision. In *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 897–903, July 2008. doi: 10.1109/AIM.2008.4601780.

Hsiu-Chin Lin, Joshua Smith, Keyhan Kouhkiloui Babarahmati, Niels Dehio, and Michael Mistry. A projected inverse dynamics approach for dual-arm cartesian impedance control. *arXiv preprint arXiv:1707.00484*, 2017.

Z Lin, Vladimir Zeman, and Rajnikant V Patel. On-line robot trajectory planning for catching a moving object. In *Robotics and Automation, 1989. Pro-*

ceedings., 1989 IEEE International Conference on, pages 1726–1731. IEEE, 1989.

Stephen R. Lindemann and Steven M. LaValle. *Current Issues in Sampling-Based Motion Planning*, pages 36–54. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

V. Lippiello and F. Ruggiero. 3d monocular robotic ball catching with an iterative trajectory estimation refinement. In *2012 IEEE International Conference on Robotics and Automation*, pages 3950–3955, May 2012a. doi: 10.1109/ICRA.2012.6224994.

V. Lippiello, F. Ruggiero, and B. Siciliano. 3D monocular robotic ball catching. *Robotics and Autonomous Systems*, 61(12):1615–1625, 2013.

Vincenzo Lippiello and Fabio Ruggiero. Monocular eye-in-hand robotic ball catching with parabolic motion estimation. *IFAC Proceedings Volumes*, 45 (22):229 – 234, 2012b. ISSN 1474-6670. 10th IFAC Symposium on Robot Control.

Y.-H. Liu and S. Arimoto. Decentralized adaptive and nonadaptive position/force controllers for redundant manipulators in cooperations. *International Journal of Robotics Research*, 17(3):232–247, 1998.

J. Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

T. Lozano-Perez. Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, C-32(2):108–120, Feb 1983. ISSN 0018-9340.

J. Y. S. Luh and Y. F. Zheng. Constrained relations between two coordinated industrial robots for motion control. *International Journal of Robotics Research*, 6(3):60–70, September 1987.

Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127 – 149, 2009. ISSN 1574-0137. doi: https://doi.org/10.1016/j.cosrev.2009.03.005.

Anirudha Majumdar and Russ Tedrake. Funnel libraries for real-time robust feedback motion planning. *The International Journal of Robotics Research*, 36(8):947–982, 2017.

M. Martino and M. E. Broucke. A reach control approach to bumpless transfer of robotic manipulators. In *53rd IEEE Conference on Decision and Control*, pages 25–30, Dec 2014. doi: 10.1109/CDC.2014.7039354.

Jacob Mattingley and Stephen Boyd. Cvxgen: A code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.

David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM

Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

James K Mills. Manipulator transition to and from contact tasks: A discontinuous control approach. In *IEEE International Conference on Robotics and Automation*, pages 440–446. IEEE, 1990.

James K Mills and David M Lokhorst. Stability and control of robotic manipulators during contact/noncontact task transition. *IEEE Transactions on Robotics and Automation*, 9(3):335–345, 1993.

Seyed Sina Mirrazavi Salehian, Nadia Barbara Figueroa Fernandez, and Aude Billard. A dynamical system approach for softly catching a flying object: Theory and experiment. *IEEE Transactions on Robotics*, 32(2):462–471, April 2016a. ISSN 1552-3098.

Seyed Sina Mirrazavi Salehian, Nadia Barbara Figueroa Fernandez, and Aude Billard. Coordinated multi-arm motion planning: Reaching for moving objects in the face of uncertainty. In *Proceedings of Robotics: Science and Systems*, 2016b.

Seyed Sina Mirrazavi Salehian, Nadia Barbara Figueroa Fernandez, and Aude Billard. A unified framework for coordinated multi-arm motion planning. *The International Journal of Robotics Research*, 2017a.

Seyed Sina Mirrazavi Salehian, Nadia Barbara Figueroa Fernandez, and Aude Billard. Dynamical system-based motion planning for multi-arm systems: Reaching for moving objects. In *International Joint Conference on Artificial Intelligence*, 2017b.

Michael Mistry and Ludovic Righetti. Operational space control of constrained and underactuated systems. *Robotics: Science and systems VII*, pages 225–232, 2012.

K. Murakami, Y. Yamakawa, T. Senoo, and M. Ishikawa. Motion planning for catching a light-weight ball with high-speed visual feedback. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 339–344, Dec 2015.

Sean Murray, Will Floyd-Jones, Ying Qi, Daniel Sorin, and George Konidaris. Robot motion planning on a chip. In *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016.

J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal. Comparative experiments on task space control with redundancy resolution. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3901–3908, 2005a.

J. Nakanishi, R. Cory, M. Mistry, J. Peters, and S. Schaal. Comparative experiments on task space control with redundancy resolution. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3901–3908,

2005b.

Akio Namiki and Masatoshi Ishikawa. Robotic catching using a direct mapping from visual information to motor command. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 2400–2405. IEEE, 2003.

K. Neumann, A. Lemme, and J.J. Steil. Neural learning of stable dynamical systems based on data-driven lyapunov candidates. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1216–1222, 2013. doi: 10.1109/IROS.2013.6696505.

Klaus Neumann and Jochen J. Steil. Learning robot motions with stable dynamical systems under diffeomorphic transformations. *Robotics and Autonomous Systems*, 70:1 – 15, 2015. ISSN 0921-8890.

Koichi Nishiwaki, A Ionno, Koichi Nagashima, Masayuki Inaba, and Hirochika Inoue. The humanoid saika that catches a thrown ball. In *Robot and Human Communication, 1997. RO-MAN'97. Proceedings., 6th IEEE International Workshop on*, pages 94–99. IEEE, 1997.

A. Orthey and O. Stasse. Towards reactive whole-body motion planning in cluttered environments by precomputing feasible motion spaces. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 274–279, Oct 2013.

P. R. Pagilla and Biao Yu. A stable transition controller for constrained robots. *IEEE/ASME Transactions on Mechatronics*, 6(1):65–74, Mar 2001a. ISSN 1083-4435. doi: 10.1109/3516.914393.

Prabhakar R Pagilla and Biao Yu. Design and experimental evaluation of geometrically constrained robots. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1308–1313. IEEE, 2000.

Prabhakar R Pagilla and Biao Yu. Robotic surface finishing processes: modeling, control, and experiments. *Transactions-American Society of Mechanical Engineers Journal of Dynamic Systems Measurement and Control*, 123(1): 93–102, 2001b.

Alexandros Paraschos, Gerhard Neumann, and Jan Peters. A probabilistic approach to robot trajectory generation. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pages 477–483. IEEE, 2013.

Ga-Ram Park, KangGeon Kim, Chang Hwan Kim, Mun-Ho Jeong, Bum-Jae You, and Syungkwon Ra. Human-like catching motion of humanoid using evolutionary algorithm(EA)-based imitation learning. In *IEEE International Workshop on Robot and Human Interactive Communication*, pages 809–815, 2009.

R. Paul. Manipulator cartesian path control. *IEEE Transactions on Systems,*

Man, and Cybernetics, 9(11):702–711, Nov 1979. ISSN 0018-9472. doi: 10.1109/TSMC.1979.4310109.

Richard Paul. Trajectory control of a computer arm. In *IJCAI*, pages 385–390, 1971.

Richard Paul. Modelling, trajectory calculation and servoing of a computer controlled arm. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1972.

Barak Pearlmutter. Learning state space trajectories in recurrent neural networks: a preliminary report. Technical report, Carnegie Mellon University, 1988.

Barak A Pearlmutter. Learning state space trajectories in recurrent neural networks. *Learning*, 1(2), 2008.

Lieke Peper, Reinoud J Bootsma, Daniel R Mestre, and Frank C Bakker. Catching balls: How to get the hand to the right place at the right time. *Journal of Experimental Psychology: Human Perception and Performance*, 20(3):591, 1994.

Nicolas Perrin and Philipp Schlehuber-Caissier. Fast diffeomorphic matching to learn globally asymptotically stable nonlinear dynamical systems. *Systems Control Letters*, 96:51 – 59, 2016. ISSN 0167-6911.

E. Plaku, K. E. Bekris, B. Y. Chen, A. M. Ladd, and L. E. Kavraki. Sampling-based roadmap of trees for parallel motion planning. *IEEE Transactions on Robotics*, 21(4):597–608, Aug 2005. ISSN 1552-3098. doi: 10.1109/TRO.2005.847599.

M.J.D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis*, volume 275, pages 51–67. Springer Netherlands, 1994.

James A Primbs, Vesna Nevistić, and John C Doyle. Nonlinear optimal control: A control lyapunov function and receding horizon perspective. *Asian Journal of Control*, 1(1):14–24, 1999.

Marc H Raibert and John J Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 102(127): 126–133, 1981.

N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *2009 IEEE International Conference on Robotics and Automation*, pages 489–494, May 2009.

N. Ratliff, M. Toussaint, and S. Schaal. Understanding the geometry of workspace obstacles in motion optimization. In *2015 IEEE International Con-*

ference on Robotics and Automation (ICRA), pages 4202–4209, May 2015.

J. H. Reif. Complexity of the mover's problem and generalizations. In *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, pages 421–427, Oct 1979. doi: 10.1109/SFCS.1979.10.

R. F. Reinhart and J. J. Steil. Neural learning and dynamical selection of redundant solutions for inverse kinematic control. In *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pages 564–569, Oct 2011. doi: 10.1109/Humanoids.2011.6100815.

Jing Ren, K. A. McIsaac, and R. V. Patel. Modified newton's method applied to potential field-based navigation for mobile robots. *IEEE Transactions on Robotics*, 22(2):384–391, April 2006. ISSN 1552-3098.

Marcia Riley and Christopher G. Atkeson. Robot catching: Towards engaging human-humanoid interaction. *Autonomous Robots*, 12(1):119–128, Jan 2002.

E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, Oct 1992. ISSN 1042-296X. doi: 10.1109/70.163777.

L. Roveda, N. Iannacci, F. Vicentini, N. Pedrocchi, F. Braghin, and L. M. Tosatti. Optimal impedance force-tracking control design with impact formulation for interaction tasks. *IEEE Robotics and Automation Letters*, 1(1): 130–136, Jan 2016. ISSN 2377-3766. doi: 10.1109/LRA.2015.2508061.

A. De Santis, A. Albu-Schaffer, C. Ott, B. Siciliano, and G. Hirzinger. The skeleton algorithm for self-collision avoidance of a humanoid manipulator. In *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, pages 1–6, Sept 2007.

Stefan Schaal. Dynamic movement primitives-a framework for motor control in humans and humanoid robotics. In *Adaptive motion of animals and machines*, pages 261–280. Springer, 2006.

Stefan Schaal, Christopher G Atkeson, and Dagmar Sternad. One-handed juggling: A dynamical approach to a rhythmic movement task. *Journal of motor behavior*, 28(2):165–183, 1996.

Stefan Schaal, Shinya Kotosaka, and Dagmar Sternad. Nonlinear dynamical systems as movement primitives. In *IEEE international conference on humanoid robotics*, pages 1–11, 2000.

Gregor Schöner and JA Scott Kelso. Dynamic pattern generation in behavioral and neural systems. *Science*, pages 1513–1520, 1988.

G. Schultz and K. Mombaur. Modeling and optimal control of human-like running. *IEEE/ASME Transactions on Mechatronics*, 15(5):783–792, Oct 2010. ISSN 1083-4435. doi: 10.1109/TMECH.2009.2035112.

T. Senoo, A. Namiki, and M. Ishikawa. Ball control in high-speed batting motion using hybrid trajectory generator. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1762–1767, 2006.

T. Senoo, G. Jinnai, K. Murakami, and M. Ishikawa. Deformation control of a multijoint manipulator based on maxwell and voigt models. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2711–2716, Oct 2016.

T. Senoo, M. Koike, K. Murakami, and M. Ishikawa. Impedance control design based on plastic deformation for a robotic arm. *IEEE Robotics and Automation Letters*, 2(1):209–216, Jan 2017. ISSN 2377-3766.

Homayoun Seraji and Richard Colbaugh. Force tracking in impedance control. *The International Journal of Robotics Research*, 16(1):97–117, 1997. doi: 10.1177/027836499701600107.

Kang Shin and N. McKay. Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, 30(6): 531–541, Jun 1985. ISSN 0018-9286. doi: 10.1109/TAC.1985.1104009.

A. Shukla and A. Billard. Coupled dynamical system based arm-hand grasping model for learning fast adaptation strategies. *Robotics and Autonomous Systems*, 60(3):424–440, 2012a.

Ashwini Shukla and Aude Billard. Augmented-svm: Automatic space partitioning for combining multiple non-linear dynamics. In *Advances in Neural Information Processing Systems*, pages 1016–1024. Curran Associates, Inc., 2012b.

Jeroen BJ Smeets and Eli Brenner. Perception and action are based on the same visual information: distinction between position and velocity. *Journal of Experimental Psychology: Human Perception and Performance*, 21(1):19, 1995.

Christian Smith and Henrik I Christensen. Using cots to construct a high performance robot arm. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4056–4063. IEEE, 2007.

Christian Smith, Yiannis Karayiannidis, Lazaros Nalpantidis, Xavi Gratal, Peng Qi, Dimos V. Dimarogonas, and Danica Kragic. Dual arm manipulation—a survey. *Robotics and Autonomous Systems*, 60(10):1340 – 1353, 2012. ISSN 0921-8890.

Nicolas Sommer, Klas Kronander, and Aude Billard. Learning externally modulated dynamical systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

Jochen J. Steil, Christian Emmerich, Agnes Swadzba, Ricarda Grünberg, Arne Nordmann, and Sebastian Wrede. Kinesthetic teaching using assisted gravity compensation for model-free trajectory generation in confined spaces. In

Florian Röhrbein, Germano Veiga, and Ciro Natale, editors, *Gearing Up and Accelerating Crossfertilization between Academic and Industrial Robotics Research in Europe:*, pages 107–127, Cham, 2014. Springer International Publishing.

R. Suda, K. Kosuge, and H. Kakuya. Object-impedance-based cooperative handling of object by mobile robot helper and human using visual and force information. In *Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings. 2003 IEEE/ASME International Conference on*, volume 1, pages 592–597 vol.1, July 2003. doi: 10.1109/AIM.2003.1225161.

MK Sudareshan and Thomas A Condarcure. Recurrent neural-network training by a learning automaton approach for trajectory learning and control system design. *IEEE Transactions on Neural Networks*, 9(3):354–368, 1998.

H. Sugiura, M. Gienger, H. Janssen, and C. Goerick. Real-time collision avoidance with whole body motion control for humanoid robots. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2053–2058, Oct 2007.

Dong Sun and J. K. Mills. Adaptive synchronized control for coordination of two robot manipulators. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 1, pages 976–981 vol.1, 2002. doi: 10.1109/ROBOT.2002.1013482.

M. Svenstrup, T. Bak, and H. J. Andersen. Trajectory planning for robots in dynamic human environments. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4293–4298, Oct 2010. doi: 10.1109/IROS.2010.5651531.

Tzyh-Jong Tarn, Yunying Wu, Ning Xi, and Alberto Isidori. Force regulation and contact transition control. *IEEE Control Systems*, 16(1):32–40, 1996.

Martin A Tayler and Keith Davids. Catching with both hands: An evaluation of neural cross-talk and coordinative structure models of bimanual coordination. *Journal of motor behavior*, 29(3):254–262, 1997.

R. H. Taylor. Planning and execution of straight line manipulator trajectories. *IBM Journal of Research and Development*, 23(4):424–436, July 1979. ISSN 0018-8646. doi: 10.1147/rd.234.0424.

Emanuel Todorov and Michael I Jordan. Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5(11):1226, 2002.

Masayoshi Tomizuka. Contact transition control of nonlinear mechanical systems subject to a unilateral constraint. *Journal of dynamic systems, measurement, and control*, 119:749, 1997.

A. Tornambe. Modeling and control of impact in mechanical systems: theory and experimental results. *IEEE Transactions on Automatic Control*, 44(2): 294–309, Feb 1999. ISSN 0018-9286. doi: 10.1109/9.746255.

Konstantinos I. Tsianos, Ioan A. Sucan, and Lydia E. Kavraki. Sampling-based robot motion planning: Towards realistic applications. *Computer Science Review*, 1(1):2 – 11, 2007. ISSN 1574-0137. doi: https://doi.org/10.1016/j.cosrev.2007.08.002.

SK Tso and KP Liu. Hidden markov model for intelligent extraction of robot trajectory command from demonstrated trajectories. In *Industrial Technology, 1996.(ICIT'96), Proceedings of The IEEE International Conference on*, pages 294–298. IEEE, 1996.

N. Uchiyama, S. Sano, and K. Ryuman. Control of a robotic manipulator for catching a falling raw egg to achieve human-robot soft physical interaction. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 777–784, Sept 2012. doi: 10.1109/ROMAN.2012.6343846.

Shriram Mahabal Udupa. *Collision detection and avoidance in computer controlled manipulators*. PhD thesis, Phd Thesis, California Institute of Technology, 1977.

J. Umlauft, A. Lederer, and S. Hirche. Learning stable gaussian process state space models. In *2017 American Control Conference (ACC)*, pages 1499–1504, May 2017. doi: 10.23919/ACC.2017.7963165.

N. Vahrenkamp, M. Do, T. Asfour, and R. Dillmann. Integrated grasp and motion planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2883–2888, 2010.

N. Vahrenkamp, T. Asfour, and R. Dillmann. Simultaneous grasp and motion planning: Humanoid robot armar-iii. *IEEE Robotics Automation Magazine*, 19(2):43–57, 2012.

Meiling Wang, Minzhou Luo, Tao Li, and Marco Ceccarelli. A unified dynamic control method for a redundant dual arm robot. *Journal of Bionic Engineering*, 12(3):361 – 371, 2015.

Wei Wang, Robert N.K. Loh, and Edward Y. Gu. Passive compliance versus active compliance in robotbased automated assembly systems. *Industrial Robot: An International Journal*, 25(1):48–57, 1998. doi: 10.1108/01439919810196964.

L. L. Whitcomb, S. Arimoto, T. Naniwa, and F. Ozaki. Adaptive model-based hybrid control of geometrically constrained robot arms. *IEEE Transactions on Robotics and Automation*, 13(1):105–116, Feb 1997. ISSN 1042-296X. doi: 10.1109/70.554351.

Thomas Wimböck and Christian Ott. *Dual-Arm Manipulation*, pages 353–366. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012a.

Thomas Wimböck and Christian Ott. Dual-arm manipulation. In *Towards Service Robots for Everyday Environments*, volume 76 of *Springer Tracks in*

Advanced Robotics, pages 353–366. Springer Berlin Heidelberg, 2012b.

Thomas Wimböck, Christian Ott, Alin Albu-Schäffer, and Gerd Hirzinger. Comparison of object-level grasp controllers for dynamic dexterous manipulation. *The International Journal of Robotics Research*, 31(1):3–23, 2012.

Sebastian Wrede, Christian Emmerich, Ricarda Grünberg, Arne Nordmann, Agnes Swadzba, and Jochen Steil. A user study on kinesthetic teaching of redundant robots in task and configuration space. *Journal of Human-Robot Interaction*, 2(1):56–81, 2013.

Y. Xia and J. Wang. A recurrent neural network for solving linear projection equations. *Neural Networks*, 13(3):337–350, 2000. cited By 95.

Jie Yang, Yangsheng Xu, and C. S. Chen. Human action learning via hidden markov model. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 27(1):34–44, Jan 1997. ISSN 1083-4427. doi: 10.1109/3468.553220.

Kamal Youcef-Toumi and David A Gutz. Impact and force control. In *IEEE International Conference on Robotics and Automation*, pages 410–416. IEEE, 1989.

M. Zhang and M. Buehler. Sensor-based online trajectory generation for smoothly grasping moving objects. In *Proceedings of 1994 9th IEEE International Symposium on Intelligent Control*, pages 141–146, Aug 1994. doi: 10.1109/ISIC.1994.367827.

Y. Zhang, S.S. Ge, and T.H. Lee. A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(5):2126–2132, 2004.

Yunong Zhang. On the lvi-based primal-dual neural network for solving online linear and quadratic programming problems. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 1351–1356 vol. 2, June 2005.

Shao Zhifei and Er Meng Joo. A survey of inverse reinforcement learning techniques. *International Journal of Intelligent Computing and Cybernetics*, 5(3): 293–311, 2012. doi: 10.1108/17563781211255862.

Wen-Hong Zhu. On adaptive synchronization control of coordinated multirobots with flexible/rigid constraints. *IEEE Transactions on Robotics*, 21(3):520–525, June 2005. ISSN 1552-3098.

Matthew Zucker, Nathan Ratliff, Anca Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher Dellin, J. Andrew (Drew) Bagnell, and Siddhartha Srinivasa. Chomp: Covariant hamiltonian optimization for motion planning. *International Journal of Robotics Research*, May 2013.

P. Ögren and N.E. Leonard. A convergent dynamic window approach to obstacle avoidance. *IEEE Transactions on Robotics*, 21(2):188–195, 2005. doi: 10.1109/TRO.2004.838008.

Petr Švestka and Mark H. Overmars. Motion planning for carlike robots using a probabilistic learning approach. *The International Journal of Robotics Research*, 16(2):119–143, 1997. doi: 10.1177/027836499701600201.

Curriculum Vitae

Seyed Sina Mirrazavi Salehian

No.34, Avenue de florissant, 1020 Renens, Switzerland

(+41) 789514497 | sina.mirrazavi@epfl.ch | lasa.epfl.ch/people/member.php?SCIPER=233855 |

sinamr66 | seyed-sina-mirrazavi-salehian-11772856

Education

Ecole polytechnique fédérale de Lausanne — *Lausanne, Switzerland*

PhD in Robotics, Control, and Intelligent Systems — *Mar. 2014 - PRESENT*

- Thesis title: Compliant control of Uni/ Multi- robotic arms with dynamical systems

Amirkabir University of Technology — *Tehran, Iran*

MSc in Mechatronics, Robotics, and Automation Engineering — *Sep. 2010 - Sep-2012*

- Thesis title: Design and analysis of a nonlinear cooperative dual-master, dual-slave teleoperation system

Amirkabir University of Technology — *Tehran, Iran*

BSc in Mechanical Engineering — *Sep. 2006 - Sep-2010*

- Thesis title: Reliability in Micro Electro Mechanical Systems (MEMS)

Professional Experience

Doctoral Researcher at Learning Algorithm and System Laboratory — *EPFL, Lausanne*

Research on motion planning and controlling of uni- or multi- robotic arms. — *Mar. 2014 - PRESENT*

- Research:
  - Development of a real-time controller which endows a multi-arm system with both synchronous and asynchronous behaviors .
  - Development of a real-time (<1 ms) solver for solving the inverse kinematic problem of multiple robotic arms with respect to the kinematic and the self-collision avoidance constraints.
  - Development of a real-time controller for catching flying objects softly .
- Projects:
  - Horizon 2020, Cogimon EU project.
  - The 7th Framework Programme, AlterEgo project.
- Live demos:
  - KUKA innovation award, Team manager, Hannover messa, MANCHU team, April 2017.
  - KUKA innovation award, Team manager, Hannover messa, CATCH team, April 2015.
- Other activities:
  - Development of a real-time control interface for KUKA IIWA.
  - Logitech G402 Proteus Speed Validation.
  - Speed Validation of the blackboards at EPFL.
  - Teaching activities in Robotic Practical, MSc course at EPFL.
  - Supervision of four internships, four master thesis and three semester student projects.

Hirgan energy engineering Co. — *Tehran, Iran*

Mechanical Rotating Equipments Engineer. — *Sep. 2012 - Aug. 2013*

- Being familiar with API 610.

Honors & Awards

April 2017	**Finalist**, KUKA innovation Award	*Hannover, Germany*
Jun 2016	**Received the Best Student Paper Award**, Robotics: Science and Systems Conference	*Ann Arbor, U.S.A*
Jun 2016	**Nominated for the Best Conference Paper Award**, Robotics: Science and Systems Conference	*Ann Arbor, U.S.A*
Jun 2016	**Nominated for the Best Systems Paper Award**, Robotics: Science and Systems Conference	*Ann Arbor, U.S.A*
April 2015	**Finalist**, KUKA innovation Award	*Hannover, Germany*
Sep. 2012	**Ranked 1st**, The master program in the mechanical engineering department	*Tehran, Iran*
Sep. 2010	**Honorary Admission**, The master program in Amirkabir university	*Tehran, Iran*
Sep. 2010	**Ranked 5st**, The bachelor program in the mechanical engineering department	*Tehran, Iran*

Skills

Machine Learning

Expert knowledge with experience of a wide range of algorithms for regression, clustering and classification.

- Dimensionality reduction (PCA, Kernel PCA, LDA, etc).
- Linear and Non-linear regressions (GMR, GP, LWRP, LPV, SVR, etc).
- Linear and Non-linear clustering and classification (GMM, SVM, K-means, etc).
- Big-Data and Learning sparse Models.

Automatic Control and Optimization Theory

Expert with several years of experience primarily with applications in robotics

- Dynamical system based motion planning.
- Linear and Non-linear control algorithms (PID, feedback linearization, backstepping, adaptive, robust, etc).
- Optimal and Model predictive control.
- Data-Driven and Linear Parameter Varying Models.

Computer Skills

- C/C++, Python, Matlab Simulink, Maple, Java.
- Distributed version control systems: bzr, git, Travis.
- Robotic interfaces: ROS, YARP, Gazebo, Orocos, Usarsim.
- Machine learning interfaces: Mlpack(c++), Scikit-learn (Python), ML_toolbox (Matlab).
- CAD softwares: Solidworks, Catia.
- Miscellaneous: LaTeX, UNIX/Linux, Windows, Adobe suite, Microsoft Office suite.

Robotic Platform

- Robotic manipulators: KUKA LWR 4+ and IIWA, ABB YU-Mi, KATANA.
- Humanoids: iCub.
- Haptic devises: Phantom omni and Falcon.
- Grippers and hands: Allegro hand, QB hand, 2-Finger Robotiq hand and Barret hand .
- Vision systems: Motive and Arena from Optitrack.

Language

- Farsi Native or bilingual proficiency.
- Azari Native or bilingual proficiency.
- English Fluent (C1).
- French Elementary proficiency (A2).

Academic Activities

Reviewer experiences

- IEEE Transactions on Robotics.
- IEEE Transactions on Control Systems Technology.
- IEEE Robotics Automation Letters.
- IEEE Transactions on Aerospace and Electronic Systems.
- IEEE/RSJ Int. Conference on Intelligent Robots and Systems.
- IEEE Int. Conference on Robotics and Automation.
- IEEE-RAS Int. Conference on Humanoid Robots.
- IEEE/ASME Int. Conference on Advanced Intelligent Mechatronics.
- Journal of Mechanical Science and Technology.
- ASME's Int. Mechanical Engineering Congress and Exposition.

Selected publications

- Mirrazavi Salehian, S. S., Figueroa, N. and Billard, A. (2016) Coordinated multi-arm motion planning: Reaching for moving objects in the face of uncertainty. In Proceedings of Robotics: Science and Systems XVI , Arbor, Michigan, USA.
- Mirrazavi Salehian, S. S., Khoramshahi, M. and Billard, A. (2016) A Dynamical System Approach for Catching Softly a Flying Object: Theory and Experiment. in IEEE Transactions on Robotics, vol. 32, no. 2, pp. 462-471, April 2016.
- Firouzeh, A., Mirrazavi Salehian, S. S., Billard, A. and Paik, J. (2015) An Underactuated Robotic Arm with Adjustable Stiffness Shape Memory Polymer Joints. Proceedings of ICRA 2015.

Personal Information, Volunteer Experience and Hobbies

Treasurer at IRSA (Iranian Students Associations), May. 2015 - May. 2016 *EPFL , Lausanne*

Member at the student community of the Mechanical Dep., Jan. 2012 - Jan. 2013 *Amirkabir university, Tehran*

Editor of the community's magazine

Treasurer at the climbing community of the Mechanical Dep., Jan. 2011 - Jan. 2012 *Amirkabir university, Tehran*

Personal Information

- 30 Years old, Married, Iranian, permit B

Hobbies

- Volleyball, Biking, Hiking, Basketball, Baking