

---

Electronic Thesis and Dissertation Repository

---

11-17-2022 10:30 AM

## A Framework for Stable Robot-Environment Interaction Based on the Generalized Scattering Transformation

Kanstantsin Pachkouski, *The University of Western Ontario*

Supervisor: Polushin, Ilia G., *The University of Western Ontario*

Joint Supervisor: Patel, Rajni V., *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Engineering Science degree in Electrical and Computer Engineering

© Kanstantsin Pachkouski 2022

Follow this and additional works at: <https://ir.lib.uwo.ca/etd>



Part of the [Biomedical Engineering and Bioengineering Commons](#), [Computer Engineering Commons](#), [Electrical and Computer Engineering Commons](#), [Mechanical Engineering Commons](#), and the [Medicine and Health Sciences Commons](#)

---

### Recommended Citation

Pachkouski, Kanstantsin, "A Framework for Stable Robot-Environment Interaction Based on the Generalized Scattering Transformation" (2022). *Electronic Thesis and Dissertation Repository*. 8970. <https://ir.lib.uwo.ca/etd/8970>

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact [wlsadmin@uwo.ca](mailto:wlsadmin@uwo.ca).

# Abstract

This thesis deals with development and experimental evaluation of control algorithms for stabilization of robot-environment interaction. A framework for stable robot-environment interaction is presented and evaluated on a real physical system. The proposed algorithm fundamentally generalizes the conventional passivity-based approaches to the coupled stability problem. In particular, it allows for stabilization of not necessarily passive robot-environment interaction. The framework is based on the recently developed non-planar conic systems formalism and generalized scattering-based stabilization methods. A comprehensive theoretical background on the scattering transformation techniques, planar and non-planar conic systems is presented. The dynamics of the robot are estimated using data-driven techniques, which allows the equations for the dynamics of a robot to be obtained in an explicit form. The generalized scattering transformation is used in combination with the Lyapunov-based adaptive trajectory tracking control. It is shown that the original interconnected system is not stable due to its non-passive nature; however, the application of the proposed stabilization algorithm allows stability to be achieved, without affecting the robot's trajectory tracking performance in free space.

**Keywords:** robot-environment interaction, machine learning, dynamics estimation, motion control, interaction control, force control, conic systems, coupled stability, stabilization, scattering transformation

## Lay Summary

Since the advent of robotics, considerable effort has been put in the development of appropriate theory, algorithms and their implementation. Nowadays, robotic systems have found applications in various areas including, but not limited to, industrial manufacturing, the services sector, driving and healthcare fields. Further integration of robotics in the economy requires enabling a higher level of physical interaction between a robot and the outside world. In other words, robots should be able to, for example, safely interact with people, other robots and various systems. The robotics research community has put a great amount of effort into investigating methods for interaction control. To date, most commonly applied algorithms for interaction control are limited to passive systems, *i.e.*, systems that do not have internal sources of energy. Simple examples of passive systems are a mass-spring-damper system and a fixed wall. In contrast, both humans and robots come under the concept of non-passive, or active systems, because they have internal sources of energy. Thus, conventional interaction control algorithms typically fail to interact with active environments in a stable manner. In this thesis, a framework for stable robot-environment interaction for passive and non-passive systems is presented and evaluated on a real robot. This work presents a comprehensive overview and theoretical background on control methods and paradigms used to design the framework. All the necessary steps for implementation of the algorithm are described. These include hardware design, a method for estimation of dynamics and cone parameters. It was experimentally verified that the proposed algorithm successfully stabilizes interaction between the robot and the environment. The proposed framework constitutes a fundamental extension of the existing passivity based approaches for the coupled stability problem. The algorithm can be applied for stabilization of interconnections of active systems. For example, possible applications of the framework are bilateral teleoperation with communication delays, robotic surgery on a beating heart, and haptics-based environments for training.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Lay Summary</b>	<b>iii</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Appendices</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Literature Review . . . . .	2
1.1.1 Impedance and Admittance Interaction Control . . . . .	2
1.1.2 Hybrid Position-Force Interaction Control . . . . .	7
1.1.3 Scattering Techniques . . . . .	8
1.2 Motivation . . . . .	10
1.2.1 Passivity . . . . .	10
1.2.2 Interconnections of Passive Systems . . . . .	11
1.3 Thesis Objectives and Contribution . . . . .	15
1.4 Thesis Outline . . . . .	16
<b>2 Conic Systems and Scattering Transformations</b>	<b>18</b>
2.1 Planar Conic Systems . . . . .	19
2.1.1 Planar Conicity . . . . .	19

2.1.2	Conic Systems as Dissipative Systems . . . . .	24
2.2	Non-Planar Conicity and Scattering Transformation . . . . .	28
2.2.1	Non-Planar Conicity . . . . .	29
2.2.2	Graph Separation Stability Condition for Non-Planar Conic Systems . . . . .	35
2.3	Scattering Transformation for Non-Planar Conic Systems . . . . .	37
2.4	Conclusion . . . . .	40
<b>3</b>	<b>Scattering-Based Design for Coupled Stability</b>	<b>41</b>
3.1	Hardware Description . . . . .	42
3.2	Manipulator Modeling and Dynamics Estimation . . . . .	42
3.2.1	Euler-Lagrange Equations . . . . .	43
3.2.2	Manipulator Modeling . . . . .	44
3.2.3	Manipulator Dynamics Estimation . . . . .	47
3.2.4	Linear Regression . . . . .	48
3.2.5	Neural Network Based Dynamics Estimation . . . . .	51
3.3	Environmental Dynamics and Control Algorithm . . . . .	53
3.3.1	Environmental Dynamics . . . . .	54
3.3.2	Adaptive Control Algorithm . . . . .	55
3.4	Dissipativity Analysis of the Robot and the Environment . . . . .	57
3.5	Analysis of the Subsystems' Cones . . . . .	59
3.6	Design of the Scattering Transformation . . . . .	62
3.7	Complete Control Architecture of the System . . . . .	67
3.8	Conclusions . . . . .	68
<b>4</b>	<b>Experimental Results</b>	<b>70</b>
4.1	Velocity and Acceleration Estimation . . . . .	70
4.2	Reference Trajectory . . . . .	72
4.3	Contact Stability Experiments . . . . .	74

4.4	Simulation Results . . . . .	76
4.4.1	Description of the Simulation Environment . . . . .	76
4.4.2	Simulation Results . . . . .	90
4.5	Conclusion . . . . .	91
<b>5</b>	<b>Conclusion</b>	<b>93</b>
5.1	Summary . . . . .	93
5.2	Future Research . . . . .	95
	<b>Bibliography</b>	<b>96</b>
	<b>Curriculum Vitae</b>	<b>108</b>

# List of Figures

1.1	Negative feedback interconnection of two passive systems . . . . .	14
2.1	A conic sector in the plane. Interior of sector is shaded. . . . .	23
2.2	Feedback interconnection of systems $\Sigma_1$ and $\Sigma_2$ . . . . .	35
3.1	Experimental setup: (a) – robot, (b) - force sensor together with the 3D printed mount. . . . .	42
3.2	Mechanical structure of the five-bar linkage manipulator used in the experiments.	43
3.3	Dynamics estimation using linear regression: (a) - prediction of the torque for joint 1, (b) - prediction of torque for joint 2. . . . .	51
3.4	Architecture of a neural network for dynamics estimation. . . . .	52
3.5	Train and validation loss of the neural network for dynamics estimation. . . . .	53
3.6	Dynamics estimation using neural network: (a) - prediction of torque for joint 1, (b) - prediction of torque for joint 2. . . . .	54
3.7	Radius of controlled manipulator’s cone as a function of $\rho > 0$ . . . . .	59
3.8	Scattering-based stabilization of robot-environment interaction. . . . .	63
3.9	Control architecture for robot-environment interaction stabilization through scat- tering transformation . . . . .	67
4.1	Desired task space trajectory of the robot (in blue) and environment position (in red). . . . .	73
4.2	Desired joint space trajectory of the robot: (a) 10.7 s variant of trajectory, (b) 12.8 s variant of trajectory. . . . .	73

4.3	Experimental results: no scattering transformation, trajectory execution time 10.7 s, initial parameters $\hat{\theta}(0) = \hat{\theta}_1^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates $\hat{\theta}$ . . . . .	77
4.4	Experimental results: the scattering transformation is applied, trajectory execution time 10.7 s, initial parameters $\hat{\theta}(0) = \hat{\theta}_1^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates $\hat{\theta}$ . . . . .	78
4.5	Experimental results: no scattering transformation, trajectory execution time 10.7 s, initial parameters $\hat{\theta}(0) = \hat{\theta}_2^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates $\hat{\theta}$ . . . . .	79
4.6	Experimental results: the scattering transformation is applied, trajectory execution time 10.7 s, initial parameters $\hat{\theta}(0) = \hat{\theta}_2^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates $\hat{\theta}$ . . . . .	80
4.7	Experimental results: no scattering transformation, trajectory execution time 10.7 s, initial parameters $\hat{\theta}(0) = \hat{\theta}_3^{init} = \mathbb{O}_{12}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates $\hat{\theta}$ . . . . .	81
4.8	Experimental results: the scattering transformation is applied, trajectory execution time 10.7 s, $\hat{\theta}(0) = \hat{\theta}_3^{init} = \mathbb{O}_{12}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates $\hat{\theta}$ . . . . .	82



4.9	Experimental results: no scattering transformation, trajectory execution time 12.8 s, initial parameters $\hat{\theta}(0) = \hat{\theta}_1^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates $\hat{\theta}$ . . . . .	83
4.10	Experimental results: the scattering transformation is applied, trajectory execution time 12.8 s, initial parameters $\hat{\theta}(0) = \hat{\theta}_1^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates $\hat{\theta}$ . . . . .	84
4.11	Experimental results: no scattering transformation, trajectory execution time 12.8 s, initial parameters $\hat{\theta}(0) = \hat{\theta}_2^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates $\hat{\theta}$ . . . . .	85
4.12	Experimental results: the scattering transformation is applied, trajectory execution time 12.8 s, initial parameters $\hat{\theta}(0) = \hat{\theta}_2^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates $\hat{\theta}$ . . . . .	86
4.13	Experimental results: no scattering transformation, trajectory execution time 12.8 s, initial parameters $\hat{\theta}(0) = \hat{\theta}_3^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates $\hat{\theta}$ . . . . .	87
4.14	Experimental results: the scattering transformation is applied, trajectory execution time 12.8 s, initial parameters $\hat{\theta}(0) = \hat{\theta}_3^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates $\hat{\theta}$ . . . . .	88
4.15	Scheme of the environment used in simulation. Trajectory is shown with the dashed line. The following are the trajectory waypoints: (a) - starting point, (b) - point of collision, (c) - desired end point. . . . .	89

4.16 Distribution of the force along the  $x$  axis of the Nano 43 sensor when the external force is equal to zero. The mean  $\mu_x = -0.0276$  H,  $\mu_y = 0.0085$  H; the standard deviation  $\sigma_x = 0.0043$  H and  $\sigma_y = 0.0059$  H. . . . . 90

4.17 Desired and actual joint positions of the robot in simulation: (a) cutoff frequency of the low-pass filter is 50 Hz, (b) cutoff frequency of the low-pass filter is 100 Hz. . . . . 91

# List of Tables

3.1	The ranges of hyperparameters tuned. Values that minimize the error on validation set are in bold. . . . .	54
-----	--	----

# List of Appendices

# Chapter 1

## Introduction

To date, there were three stages of industrial revolution. The first industrial revolution dates back to thousands of years ago to the late Stone Age. The division of labor marks a significant milestone in the history of humanity, allowing for faster and more efficient production of goods. This, in turn, triggered a chain of new discoveries and innovations in the next millennia. The second industrial revolution occurred in the late 18th century with the introduction of manufacturing machinery, where steam power was harnessed for manufacturing and transportation. The latest industrial revolution took place in 1970s with the introduction of robotic systems in the industrial manufacturing and even beyond, for example, in health care and service sectors. The concept of the fourth industrial revolution – ”Industry 4.0” – was first formed in 2011 at the Hannover Industrial Exhibition in Germany [90]. The aim of this concept is to bring positive change to the economy and society through the use of the intelligent robotic systems, Internet of Things (IoT), Artificial Intelligence (AI), Big Data, and other scientific and engineering developments. The technologies mentioned should be more tightly integrated into different economic sectors and everyday life in order to allow for more efficient production, better healthcare, and in general safer and more comfortable people’s lives. In particular, one of the key factors of the ”Industry 4.0” concept is to enable safe physical interaction of robots with humans and the environment.

The goal of this thesis is to design a framework for stable robot-environment interaction. One of the main challenges associated with robot-environment interactions is to ensure stable contact between the robot and the environment. When a robot encounters an environment, the interaction force is generated between the robot and the environment, thus forming the closed-loop dynamics of the robot-environment system. The problem of stability of robot-environment interaction, also known as contact or coupled stability, is a fundamental problem in robotics. The term “environment” is understood here to mean any physical object(s) other than the controlled robot such as a conveyor belt, other robots, or people.

The rest of this chapter is organized as follows. In Section 1.1, a review of the literature related to stability and control of robot-environment interaction is presented. Section 1.2 provides motivation behind the research. Section 1.3 discusses the goals of the thesis and its contributions. Section 1.4 describes the overall thesis structure.

## **1.1 Literature Review**

In general, there exist several approaches that can be used for control of robot-environment interaction; the most popular approaches include impedance control, admittance control, and hybrid force/position control. In this Section, background materials related to the problem of coupled stability are described. Mainly, the background section covers the developments in impedance, admittance, and hybrid control approaches. In addition, scattering-based techniques are also described.

### **1.1.1 Impedance and Admittance Interaction Control**

The objective of impedance control is to modulate the apparent stiffness, damping, and inertia of the robot. Admittance control, on the other hand, can be described as position-based impedance control [61]. Specifically, admittance control achieves the effect of compliant control performance by changing the reference trajectory [70] in response to the force ap-

plied, whereas the idea behind impedance control is to modulate force in response to a change in trajectory. The desired control performance in impedance and admittance control is guaranteed through imposing a proper impedance model, which relates the external force to the manipulator's motion. Thus, selection of the impedance/admittance model parameters appears particularly important. This is achieved by using a dynamical model of the robot and the contact force feedback. In early research works on impedance control, a desired impedance model was prescribed to the robot, while the main challenge was to precisely estimate the dynamics of the robot and to handle the uncertainties of the model [16, 59, 13, 91, 54]. Some works are based on learning the impedance model, or implementing an adaptive impedance model. In some cases, a dynamical model of the environment is also incorporated into the impedance control scheme [12], which enables the impedance behaviour to be more agile. Moreover, there has been a significant research effort related to implementing control schemes with variable impedance [81, 82, 10, 36, 37]. In these works it was shown that, compared to controllers with fixed impedance, variable admittance controllers result in better stability and performance in robot-environment interaction tasks where the environment has variable stiffness properties or where the robot interacts with a human. The main idea of variable impedance approaches is to switch parameter values between predefined constants. Although these approaches may provide better control performance, the impedance parameters (stiffness, mass, and damping) are obtained using heuristic methods and cannot be easily used in other applications. To overcome this problem, iterative learning approaches have been developed to find impedance parameters in cases where the environmental dynamics are unknown. The idea of the iterative learning approach is to repeat the same task with the goal of iteratively improving the control performance [3, 4]. Indeed, in order to learn how to interact with previously unknown objects or environments, humans typically take several, possibly unsuccessful, attempts [21, 100]. In [15], associative search network learning is applied to a wall-following task. In [101], an impedance learning approach based on an internal model is developed and implemented for a high-speed insertion application. In [80], neural networks are used to regulate the impedance parameter of

the end-effector, *i.e.*, the neural networks for position and velocity are used to control a robot in free space. The other neural network is used to control the robot during a contact with an environment. Thus, this method regulates stiffness, damping, inertia and virtual trajectory of the end-effector. There has also been some research effort in investigating adaptive methods for obtaining the impedance parameters. This approach does not require the robot to repeat the same operation. However, impedance adaptation is a challenging problem, because such adaptation may require some environmental variables to be invariant, which is not the case with a dynamically changing environment. There has been extensive research done in impedance adaptation; however, this approach is less popular as compared to impedance learning. In [83], a trajectory tracking control method is proposed for multi-joint robots by adapting the stiffness through the use of resonance. In [76], switching between different values of the impedance parameters is implemented in order to dissipate the energy of the system faster.

In [23], the researchers proposed tuning the admittance damping parameter based on the measurements performed by a force/torque (F/T) sensor mounted at the end-effector of a 3-DOF Cartesian robot. The inertia parameter in this approach remains constant. In their work, forces and the rate of change of the forces are used to predict human intentions. In [46], variable admittance control laws are proposed that adjust the admittance parameters based on the magnitude of acceleration. This approach was further improved in [45], where the authors added an algorithm to anticipate and avoid kinematic limitations. Compared to the previous control schemes, this approach has faster execution time and lower errors in robot-environment interaction tasks. However, the authors used three or four translational degrees-of-freedom (DOF) systems in their experiments, which lack the coupling complexity of a 6-DOF robot. In [25], the researchers combined Cartesian impedance modulation and redundancy resolution to obtain better performance in human-robot physical interaction. They concluded that controllers with variable impedance demonstrate better performance compared to those with the fixed impedance, because in the case of manual guidance the robot is able to adapt its behaviour dynamically depending on the task and the human intention, which makes it more comfortable



for the user to control the robot. In addition, variable impedance controllers allow to reach a more favorable trade-off between accuracy and execution time as compared to controllers with fixed impedance. Labrecque and Gosselin [44] presented a control scheme where variable impedance allows a smooth transition between two interaction modes, unilateral and bilateral. Their experimental setup includes a 7-DOF robot with two F/T sensors. Experiments showed that this approach allows an operator to interact with a robot in an efficient and intuitive manner. This approach improves performance and stability in physical Human-Robot Interaction (HRI); however, the implementation and validation are explored on a collaborative robot and rely on its advantages such as inherent low inertia and joint torque sensing. In [22], an online approach is proposed that calculates virtual constraining repulsive forces based on different performance indices to avoid singularities and achieve low efforts in human–robot cooperation tasks using a 7-DOF robot.

Generally, optimization is an integral part of impedance learning and adaptation algorithms, because these algorithms control both force and position, and the control design typically involves a trade-off between the two objectives. A reward function or a cost function can be defined which evaluates the interaction performance, and the parameters of control scheme subsequently can be chosen to maximize the reward function or minimize the cost function. For example, in [38], a Linear Quadratic Regulator (LQR) is used to find the optimal parameters for a control algorithm in the situation where the environmental dynamics are known beforehand. In [62], the environmental dynamics are also assumed to be known, however, the parameters of the control algorithm are updated online, in contrast with [38] where the parameters are fixed after optimizing the LQR parameters. The drawback of these two control algorithms is that the system’s dynamics have to be known *a priori*. To address the problem of unknown system dynamics in optimal control, techniques such as Reinforcement Learning or Adaptive Dynamic Programming (ADP) have been extensively researched [9, 51, 92, 95, 93, 96, 94]. The main idea of Adaptive Dynamic Programming is to design a control scheme that would mimic the way biological systems interact with their surrounding environment. The control

system in ADP is defined as an agent or an actor which modifies its behaviour based on the previous outcomes of interaction with the environment. The agent or actor are either punished or rewarded for a control action based on the cost function [51, 92]. Out of all ADP control schemes, the most recognized are heuristic dynamic programming, Q-Learning, globalized and dual-heuristic programming. The main feature of these control approaches is that only partial information about the system's dynamics is required for the design of the optimal controller. There are also works where ADP is used for impedance adaptation of a robotic manipulator. In [57], the output feedback adaptive dynamic programming method is used to determine the desired impedance parameters. In [11], variable impedance control is designed by using path-integrals ( $PI^2$ ) in the Reinforcement Learning algorithm. The Path-Integrals algorithm is a sampling-based learning algorithm which is derived from first principles of stochastic optimal control. This technique allows for learning the joint-space variable impedance skills. That is, by specifying the task through the cost function, this algorithm tunes the stiffness for high performance in various tasks, *e.g.*, via-point trajectory following and flipping a light switch. The disadvantage of this method is that the joint-space impedance learned has limited policy transferability. This technique was further improved in [71], where the state-dependent stiffness model is learned together with trajectories. In [60], the authors investigated how the choice of action space influences the performance of the RL algorithm in manipulation tasks. It was found that the variable impedance control in end-effector space performs best in contact-rich and constrained tasks. However, the performance of the RL algorithm highly depends on the design of the reward function. The main idea of Inverse Reinforcement Learning approaches is to infer the reward function from expert demonstrations, whereas in traditional Learning from Demonstration (LfD) methods, the goal is to mimic expert actions. In RL methods, the variable impedance policies are obtained by maximizing the reward function using forward RL. One commonly used framework for IRL is the maximum entropy IRL [105]. This work assumes that expert trajectories follow a Boltzmann distribution model and updates the cost function by maximum likelihood learning. This framework was improved in [50] by using

the local Laplace approximation of the cost function, which allowed inverse optimal control to be performed in high dimensional and continuous domains. However, for the cost function updates, these methods require a dynamical model of the system, which may be difficult to obtain, particularly for the more complex robotic manipulators.

Recently, a sample-based IRL approach was proposed [28], where a nonlinear cost function is learned for high-dimensional continuous systems. In this approach, the actor alternates between optimizing the cost function and optimizing the variable impedance policy, which in turn generates an optimal trajectory. This IRL approach does not require a dynamic model of the system. Later the authors combined this approach with Generative Adversarial Networks and introduced the GAN-GCL algorithm [27]. The performance of this algorithm was improved in [30] where an Adversarial Inverse Reinforcement Learning (AIRL) algorithm extended the GAN-GCL to single state-action pairs, which helped to achieve better results in simulation. In [104], variable impedance gain action space is introduced, that allows more general representation of the expert policy to be found compared to using force as action. This improves the reward function transfer performance in a new task setting.

### **1.1.2 Hybrid Position-Force Interaction Control**

The other type of interaction control algorithms is the hybrid position/force control. The idea behind the hybrid control strategy is to control either a position or a force in complementary subspaces of the task space, through the use of proper selection matrices [69]. Lozano and Brogliato [58] presented an adaptive controller for redundant manipulators, which is based on the decomposition of the robot Jacobian and the environmental stiffness matrices. This approach does not require measurements of joint accelerations and force derivatives. Yoshikawa and Sudou [35] further improved the control scheme presented in [69]. The authors proposed a dynamic hybrid position/force control architecture, which takes into consideration the manipulator dynamics and the end-effector constraints specified by the given task. In [43], the authors proposed a new robust adaptive control scheme for simultaneous force/motion control of con-

strained rigid robots. The algorithm is based on a sliding-mode technique which guarantees robustness with respect to parameter variations in both manipulator and motors dynamics. The adaptive force/position controller proposed in [63] is based on a strict-feedback backstepping technique. The authors in [20] presented an adaptive position/force controller for constrained environments. This control architecture does not require the measurements of velocity. In [99], the authors used the force/torque and vision sensors in their implementation of the hybrid position/force controller. This control scheme allows a manipulator equipped with an uncalibrated camera and force sensor to move along a path on an unknown surface with acceptable position/force errors. In [72], the authors presented an adaptive force control algorithm with low-level position/velocity controllers for robotic manipulators. This algorithm achieves stable contact with surfaces with unknown linear compliance. In [14], a motion and force tracking controller is proposed for robots with uncertain kinematics and dynamics. In [41], a method is proposed that reduces impulsive contact force between the manipulator and its environment. In [31], the authors use Neural Networks to compensate for unmodeled dynamics of the robot using hybrid force/position controllers.

### 1.1.3 Scattering Techniques

The scattering transformation, also called the wave transformation or the Cayley transformation, has been known in robotics for a long time. However, initially this approach originated in the field of electrical networks, where it was used in transmission lines and distributed networks with delays [97]. In the field of robotics and control, the scattering transformation was first introduced in the work of Anderson [1], where it was used for establishing the relationship between the small-gain theorems and the notion of passivity. Later in the work of Anderson and Spong [2], the approach based on the scattering transformation was used to stabilize force-reflecting teleoperators in the presence of communication delays. The scattering based approach presented in [2] and that in the parallel developments of Niemeyer and Slotine [64], appear to be the backbone of most popular methods for stabilization of bilateral teleoperators.

That is, scattering-based techniques are commonly used to tackle the instabilities caused by communication delays in the force reflecting teleoperators [65, 77]. The stability is achieved due to the fact that conventional scattering operator transforms a passive system into a system with  $L_2$ -gain less than or equal to one, as shown in [2, Theorem 3.1]. More specifically, if we represent a force reflecting teleoperator as an interconnection of passive subsystems, the scattering operator applied to the both sides of the communication channel would transform these passive systems into systems with  $L_2$ -gain less than or equal to one. The stability of the overall interconnected system, *i.e.*, the bilateral teleoperator, then follows from the small gain theorem.

Extension of conventional scattering transformation to the case of not necessarily passive systems was pursued in [34] and [68]. The proposed generalization of scattering techniques is based on the fact that the conventional scattering operator performs a rotation of the input-output pair by the angle  $\pi/4$ . That is, the approach described in [68] performs a rotation by the desired angle which may not necessarily be  $\pi/4$ . In addition, the described extension allows a desired gain to be prescribed to the system being transformed. Thus, these two developments provide more control over the transformation to be designed, which in turn makes it possible to formulate more general stability conditions for the interconnected system. For example, the generalized scattering transformation described in [68] is used to satisfy the graph separation stability condition, which defines conditions for a more general stability in comparison with the small gain theorem. The developments presented in [68] rely heavily on the conic system formalism originally described in the work of Zames [103]. That is, a dynamical system is represented as a conic sector on the plane parameterized by two variables. This notion is fairly general and includes finite  $L_2$ -gain stable, passive and non-passive systems. Extensions to nonlinear conic sectors were pursued in [73] and [78], which allows for a more versatile representation of a system.

Further generalizations of the scattering techniques are based on the non-planar conic system formalism [85, 86, 87]. In the case of the coupled stability problem, the use of the non-

planar conic systems framework and generalized scattering transformations may potentially lead to fundamental extensions of the existing coupled stability criteria and methods for stabilization of robot-environment interaction. Direct application of the methods developed in [85]– [87] to the coupled stability problem, however, is not preferable, particularly because straightforward designs based on the methods from the above cited works would interfere with a robot’s tracking performance in free space. The design framework for the coupled stability problem which is compatible with an arbitrary tracking control algorithm and does not affect the trajectory tracking performance in free space was developed in [84]. Specifically, this work formulates a problem of control design for coupled stability as the problem of design of a scattering transformation that stabilizes the robot-environment interaction while satisfying structure constraints that preclude its interference with the tracking control algorithm during the free space motion. A procedure was presented for constrained scattering-based design as well as a detailed control design example. In this example a manipulator controlled by a trajectory tracking algorithm experiences nonpassive contact with an environment which results in coupled instability. The application of the developed scattering-based methods stabilizes the robot-environment interaction. The work presented in [84] can be regarded as the most advanced and applicable scattering-based approach for the stabilization of a coupled system. The goal of this thesis is to adapt and test the scattering-based framework proposed in [84] on a real, physical robot-environment interaction system.

## **1.2 Motivation**

### **1.2.1 Passivity**

Most of the interaction control algorithms described above are based on an assumption that the closed-loop system is passive. Their results either show limited performance or no performance at all with non-passive systems. However, there are many examples of inherently non-passive environmental dynamics that include a user’s dynamics in a robotic rehabilitation systems [5],

robotic surgery on a beating heart [102], terrain with slippage in mobile robotics [52, 53], digitally implemented environments for haptics applications [55], *etc.* Another example of active environment is a second manipulator executing a prescribed trajectory.

It is also worth noting that the closed-loop system composed of the interconnection of the robot and its environment might not be passive, even if it appears to be so. This problem is often overlooked in the robotics community, as in most cases researchers simply assume the interconnection is passive, without going into detail. However, in reality an interconnection of two passive systems might be non-passive in a strict sense. This will be discussed in more detail in the next section. In addition, even for passive environments, a more detailed description of the environmental behaviour can frequently be obtained which forms a (possibly small) subset of a general passive behaviour. In many situations, control schemes based only on passivity may be overly conservative and carry unneeded constraints on the system. For example, the passivity requirement imposed on closed-loop robot dynamics contradicts with the manipulator's tracking performance [84].

Therefore, the need to design control architectures for interaction with non-passive systems arises. Algorithms for coupled stability that go beyond the passivity framework are presented in [17, 18]. In fact, these works implement loop transformations that, for linear time-invariant systems, expand the passivity and the small-gain criteria to more general cases of graph separation stability conditions.

### 1.2.2 Interconnections of Passive Systems

In this section, properties of interconnections of passive systems are discussed. In particular, we elaborate that the interconnection of two passive systems, the robot and its environment in our case, can be non-passive. Consider an affine nonlinear control system of the following form

$$\Sigma : \begin{cases} \dot{x} = f(x) + g(x)u, \\ y = h(x) \end{cases} \quad (1.1)$$

where  $x \in X \subset \mathbb{R}^n$ ,  $u \in \mathbb{R}^m$ ,  $y \in Y \subset \mathbb{R}^m$  are the state, input and output of the system, respectively;  $f: X \rightarrow \mathbb{R}^n$ ,  $h: X \rightarrow \mathbb{R}^m$ , and  $g: X \rightarrow \mathbb{R}^{n \times m}$  are functions of the state. Let the state space  $X \subset \mathbb{R}^n$  be an open connected set. Then the map  $f: X \rightarrow \mathbb{R}^n$  is the vector field, defined on  $X$ . The vector field is said to be smooth if it is smooth (differentiable) as a map  $f$ . If  $V: X \rightarrow \mathbb{R}$  is a differentiable function, then the Lie derivative of  $V(x)$  along the flow of vector field  $f$  is defined as follows

$$L_g V(x) = \frac{\partial V}{\partial x} f(x). \quad (1.2)$$

The value of  $L_g V(x)$  is equal to the rate of change of  $V(x)$  along the trajectories of the differential equation  $\dot{x} = f(x)$ . The higher order Lie derivatives can be computed using the following equation:

$$L_g^k V(x) = L_g(L_g^{k-1} V(x)). \quad (1.3)$$

For simplicity let us say that the storage function  $V \in C^k$ , where  $k = 0, 1, 2, \dots$ , if  $V$  is  $k$  times differentiable. Let us define the system (1.1) when  $u \equiv 0$  as

$$\Sigma_0: \begin{cases} \dot{x} = f_0(x), \\ y = h_0(x) \end{cases} \quad (1.4)$$

Let us also define the derivative of  $f_0$  with respect to  $u_i$  as

$$g_i^0(x) = \frac{\partial f_0(x)}{\partial u_i}, 1 \leq i \leq m, \quad (1.5)$$

and consider the following distribution

$$\mathcal{D} = \text{span} \left\{ \text{ad}_{f_0}^k g_i^0 : 0 \leq k \leq n-1, 1 \leq i \leq m \right\}.$$



Let us define the following sets

$$\begin{aligned}\mathcal{S} &:= \left\{ x \in X : L_{f_0}^k L_{\tau} V(x) = 0 \forall \tau \in \mathcal{D}, \text{ and } \forall 0 \leq k \leq r-1 \right\}, \\ \Omega &:= \left\{ x \in X : L_f^k V(x) = 0, k = 1, \dots, r \right\}\end{aligned}$$

where  $r \geq 1$  is the order of smoothness of the storage function  $V$ . In [56], it was found that for passive systems of type (1.1) with a positive storage function  $V \in C^r$ , the condition  $\Omega \cap \mathcal{S} = \emptyset$  implies that the system (1.1) is zero-state observable. The following result is then valid:

**Lemma 1.2.1 (Condition for 0-GAS) [67]** *Let the system (1.1) be passive with positive storage function  $V \in C^r$  and also be zero-state observable, i.e.,  $\Omega \cap \mathcal{S} = \emptyset$ . Let us also define the following control law*

$$u = -\phi(y), \tag{1.6}$$

where  $\phi : \mathbb{R}^m \rightarrow \mathbb{R}^m$  is the smooth function such that  $y^T \phi(y) > 0$  for all  $y \neq 0$ . Then the control law (1.6) ensures that the zero-state of the system is globally asymptotically stable.

For the affine systems of type (1.1), it is possible to find a state feedback such that the system becomes stable. For example, let the system (1.1) be stable with zero input and Lyapunov function of the form  $V \in C^r, r \geq 1$ . Then the frequency theorem (Yakubovich-Kalman Lemma) [7] implies that if we choose the output as  $y = L_g V(x)$ , then the system is passive. Hence, using the lemma 1.2.1 we get the following result [67].

**Theorem 1.2.2 (Conditions for 0-GAS [67])** *Let the system (1.1) be zero-input stable with Lyapunov function  $V \in C^r, r \geq 1$  and also be zero-state observable, i.e.,  $\Omega \cap \mathcal{S} = \emptyset$ . Then the control law*

$$u = -\gamma L_g V(x) \tag{1.7}$$

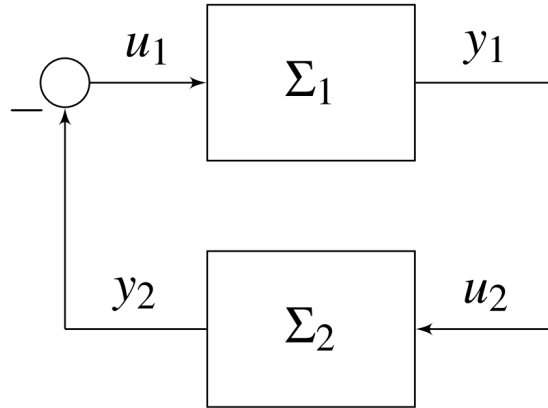


Figure 1.1: Negative feedback interconnection of two passive systems

ensures zero-state global asymptotic stability with any  $\gamma > 0$ .

The above theorem generalizes the results obtained in [39, 40, 47]. Next, consider two interconnected systems as shown on Figure 1.1. The following result is valid:

**Theorem 1.2.3 (Passivity of interconnected systems [67])** *Let  $\Sigma_1$  and  $\Sigma_2$  be the affine non-linear control systems, such that  $u_2 = y_1$ ,  $u_1 = -y_2$ . Suppose that both systems  $\Sigma_1$ ,  $\Sigma_2$  are passive, and the corresponding storage functions  $V_1(x_1)$  and  $V_2(x_2)$  are continuously differentiable and positive definite. We see that  $V(x_1, x_2) = V_1(x_1) + V_2(x_2)$  is positive definite as a function of  $x_1, x_2$ , and  $\dot{V}_1(x_1) \leq y_1^T u_1$ ,  $\dot{V}_2(x_2) \leq y_2^T u_2$ . This implies the following:*

$$\begin{aligned} \dot{V}(x_1, x_2) &= \dot{V}_1(x_1) + \dot{V}_2(x_2) \leq \\ &\leq y_1^T u_1 + y_2^T u_2 = -y_1^T y_2 + y_1^T y_2 = 0. \end{aligned}$$

Therefore the feedback system  $\Sigma = \Sigma_1 + \Sigma_2$  is Lyapunov stable and passive.

The robot-environment interaction system can be represented as two interconnected passive systems. In fact, the environment has force  $f_e$  as an input and velocity  $\dot{x}_e$  as an output. The robot as a physical system has velocity  $\dot{p}$  at the input and end-effector force  $f_r$  at the output. This force can also be defined as an external torque  $\tau_e$ . Thus, whenever the robot's end-effector

is in contact with its environment, the velocities of robot and environment are equal  $\dot{p} = \dot{x}_e$ . The force applied at the end-effector is opposite to the force of the environment  $f_r = -f_e$ .

Considering this setup, it is common in the robotics community to assume that this interaction system is passive. Typically, researchers assume it is passive without going into details. However, it is not always correct. Considering the methodology described above, namely, Theorems 1.2.2 and 1.2.3, we conclude the system is non-passive.

In our setup, the robot is controlled using the passivity-based adaptive control (2.3). In [66], it is proven that this algorithm is passive with respect to input  $r$  and output  $u$ . The vector  $r$  represents the combined position and velocity error, as in equation (2.5). Let us assume that the output of the environment  $\dot{x}_e$  is the new input to the control algorithm (2.3), *i.e.*, we change the input of the control algorithm from  $r$  to  $\dot{x}_e$ . According to Theorem (1), necessary condition for the system to be passive is that the input has the specific form (1.7). Therefore, in our case, if we change the input  $r$  to input  $\dot{x}_e$ , the robot becomes non-passive. Theorem 1.2.3 requires both interconnected systems to be passive; hence, the joint system  $\Sigma$  is also non-passive in that case. The non-passivity of the joint system leads to instabilities that were observed in experiments described further in this thesis.

### 1.3 Thesis Objectives and Contribution

Summarizing the background presented in the previous section, the following should be noted. First of all, impedance, admittance and hybrid control algorithms are generally not applicable to the case of interconnection of non-passive systems. That is, in a strict sense, these algorithms are not stable when applied to non-passive systems. The recent developments based on scattering techniques therefore deserve attention; however, there is no published work to date that evaluates the performance of the most recent and advanced scattering techniques for stabilization on a real physical robot-environment interaction system.

This thesis deals with development and experimental evaluation of control algorithms for

stabilization of robot-environment interaction based on the conic systems formalism and scattering transformation techniques. The main goal of this work is to adapt and evaluate the generalized scattering transformation developed in [84] on a real robot-environment system. The experiments were conducted on a 5-bar linkage robot interacting with soft tissue, that represents the environment. Since the scattering transformation algorithm requires knowledge of the dynamic model of a system, it becomes necessary to implement algorithms to estimate these dynamics. The dynamics of the robot were estimated using two data driven Machine Learning techniques: linear regression and neural network based approach. Due to the fact that the overall system is non-passive, instability occurs when the robot encounters its environment; however, when the scattering-based algorithm is applied, the instability vanishes, while the task-space trajectory in free space is preserved. Thus, it was verified in this work that a scattering-based technique works not only in simulation but also on a real coupled system. Further, this thesis presents a comprehensive theoretical background and a review on conic systems and scattering transformation techniques, generalizing the currently available knowledge on this topic.

## 1.4 Thesis Outline

This thesis is organized as follows: Chapter 2 contains a thorough theoretical foundation required for the implementation of the scattering-based stabilization algorithms. In particular, Chapter 2 presents the theoretical background on planar and non-planar conic systems, defines conditions for stability of coupled systems and describes scattering transformation techniques for stabilization. Chapter 3 describes all the necessary steps to implement a scattering-based stabilization framework on a real system. Chapter 4 presents the results of experiments with scattering transformations on a real coupled system. The chapter provides a description of a set-up of the interconnected system and presents evaluation results of the scattering-based stabilization algorithm. Chapter 5 presents a summary of the work performed and possible

directions for future work.

## **Chapter 2**

# **Conic Systems and Scattering Transformations**

The goal of this chapter is to present the theoretical background related to different versions of the conicity notion and the scattering transformations. This theoretical foundation forms a basis for development and implementation of the framework for stabilization of robot-environment interaction which is described subsequently in Chapter 3. In order to analyze stability and implement scattering transformation on an interconnected system, the interconnected dynamic systems need to be represented as conic systems.

The theoretical background related to conventional (planar) notion of conicity is presented in Section 2.1. Further, Section 2.2 describes a more general class of dynamical systems – the so-called non-planar conic systems. The same section also introduces methods for calculation of the dynamic cones' parameters and for stability analysis of interconnections of conic systems. Finally, Section 2.3 describes a method for stabilization of interconnected systems using scattering transformation techniques.

## 2.1 Planar Conic Systems

To describe the scattering transformation approach and how it can be applied to stabilize interconnected systems, first the notion of a conic system needs to be defined. In this section, the notion of a conic system is defined following the original work of Zames [103]. This section starts with the theoretical background necessary for the introduction of conic systems. Subsequently, several properties of the representation of such systems are given.

### 2.1.1 Planar Conicity

Systems, broadly speaking, can behave in one of the two opposing ways: it can either be stable or unstable. Generally, the stability of a system can be analyzed using either Lyapunov's methods, or an alternative approach that evaluates the relationship between the input and the output. Below, the latter approach is chosen for the evaluation of a system's stability properties. Using this approach, a system is defined as a function that maps one function of a time, called the input, to the other function of time, called the output. In the majority of cases, the output of a given system might be required to track some function of the input. In order for a system to be stable, it must exhibit the following two properties:

1. The system must be non-explosive, *i.e.*, bounded inputs must result in bounded outputs.
2. Outputs must not be sensitive to small changes in inputs.

Here both stable and unstable systems will be analyzed; therefore, the space where input and output functions are defined need to include both bounded and unbounded functions. In other words, the space also needs to include functions that grow without bound as time approaches infinity. For example, this space has to include logarithmic functions  $\log_2(t)$  and exponential functions  $e^t$ . Such functions are not contained in the spaces commonly used in functional analysis, for example, in Lebesgue linear spaces  $L_p$ . Therefore, to meet the requirement defined above, a normed linear space  $X$ , that contains bounded functions, can be extended

by including unbounded functions to form a special space  $X_e$ . That is,  $X_e$  will be the enlargement or extension of the normed linear space  $X$ . Each finite-time truncation of a function from  $X_e$  will belong to  $X$ . Thus, a function  $x \in X_e$  truncated on  $[0, t]$  will have a finite norm and lie in  $X$ . However, this norm may grow without bound as time approaches infinity.

In the strict sense, spaces  $X$  and  $X_e$ , functions  $x$  and their truncations  $x_t$  can be defined as described below.

**Truncated functions.** Let  $V$  be a linear space,  $T$  be an subinterval of the reals, for example,  $(\infty, t_0]$ . Let  $x$  be a function that maps points from  $T$  to  $V$ , *i.e.*,  $x : T \rightarrow V$ . Let  $t$  be any point in  $T$ . Then the function  $x$  truncated at time  $t_0$ , denoted by  $x_{t_0} : T \rightarrow V$ , can be defined as follows

$$x_t(t) = \begin{cases} x(t), & \text{for } t < t_0 \\ 0, & \text{for } t \geq t_0. \end{cases}$$

**Space  $X$ .**  $X$  is a normed linear space consisting of functions  $x : T \rightarrow V$  which satisfies the following assumptions:

1. If  $x \in X$  then  $x_t \in X$  for all  $t \in T$ .
2.  $\|x_{t_0}\|$  is a nondescending function of  $t_0 \in \{t | a < t < b\} \subset T$ , where  $a$  and  $b$  are any finite numbers in  $T$ .
3. If  $\lim_{t \rightarrow \infty} \|x_t\| < +\infty$ , then  $x \in X$  and  $\lim_{t \rightarrow \infty} \|x_t\| = \|x\|$ .

**Space  $X_e$ .**  $X_e$  is an enlargement of space  $X$ , that is, it consists of  $X$  plus all functions with unbounded norm at infinity, *i.e.*,  $X_e = X \cup \{x \mid \|x\| \rightarrow \infty \text{ as } t \rightarrow \infty, \text{ while } \|x_t\| < \infty \text{ for all } t < \infty\}$ . Thus, all finite time truncations of any  $x \in X_e$  will also belong to  $X$ . An extended norm of a function, denoted as  $\|x\|_e$  is defined as follows:  $\|x\|_e = \|x\|$  for  $x \in X$ , and  $\|x\|_e = \infty$  for  $x \notin X$ .

The model of a system will be defined as a relation between points in  $X_e$ . That is, such a relation maps some functions from a space  $X_e$ , called input space, into a set of functions



functions in another space  $X_e$ , called output space. Due to the fact that different initial states of a system may result in different outputs for the same input, the system model should do precisely that – map some inputs into possibly many outputs. In a strict sense, the model of a system can be described as follows:

**System Model.** A model of a physical system can be represented as a relation  $H$  on  $X_e$ , which, in turn, can be represented as a subset of the product space  $X_e \times X_e$ . Pairs  $(x, y)$  in product space  $X_e \times X_e$  constitute a mapping of input functions to the output functions. Output  $y$  will be said to be  $H$ -related to input  $x$ ;  $y$  will also be said to be an image of  $x$  under relation  $H$ . It is worth noting, that given the notations described above, systems that "explode" only on infinite times will be considered here. The reason for this is because outputs of the system are in  $X_e$ , *i.e.*, the output functions are bounded on a finite time interval. Further, a domain  $Do(H)$  and codomain  $Ra(H)$  of relation  $H$  on  $X_e$  can be defined as the following sets:

- $Do(H) = \{ x \mid x \in X_e, \text{ and there can be found } y \in X_e \text{ such that a pair } (x, y) \in H \text{ exists} \}$
- $Ra(H) = \{ y \mid y \in X_e, \text{ and there can be found } x \in X_e \text{ such that a pair } (x, y) \in H \text{ exists} \}$

In order to simplify many derivations, it is best to consider a type of relations  $H$  that map zero element to zero. If this condition is not met for some mapping  $H$ , the outset in this case can be shifted so that the condition is satisfied. Thus, consider the following definition.

**Class  $\mathcal{R}$ .**  $\mathcal{R}$  is the class of those relations  $H$  on  $X_e$  having the property that the zero element, denoted  $o$ , lies in  $Do(H)$ , and  $Ho = o$ . Consider the following assumptions about  $H$ :

1. If  $H, K \in \mathcal{R}$ , and  $c$  is a real constant, product  $cH$ , sum  $(H + K)$ , composition product  $(KH)$  are the relations in  $\mathcal{R}$ .
2. The inverse of  $H \in \mathcal{R}$ , denoted by  $H^{-1}$ , belongs to  $\mathcal{R}$ .

Stability property can be defined as follows: a system is stable if it produces bounded outputs for any bounded inputs. In addition, the system should not be sensitive to small variations of input. Consider the following definition of the input-output stability.

**Input-Output Stability.** A system  $H$  is input-output stable if it satisfies the following two conditions:

1. Relation  $H$  is bounded. That is, for any bounded subset  $U \subset X_e$ , the image  $Hu$  for all  $u \in U$  is also bounded.
2.  $H$  is continuous. That is, given any input function  $x_0 \in X$ , and any arbitrarily small  $\varepsilon > 0$ , there exists  $\delta(\varepsilon) > 0$  such that, for all  $x \in \{x \mid \|x - x_0\| < \varepsilon, x \in X\}$ , norms of all images  $\|Hx - Hx_0\| < \delta(\varepsilon)$ .

A co-domain  $Ra(H)$  of some relations  $H$  in class  $\mathcal{R}$  may have, or be inscribed in, a specific geometric form. In other words, the relation  $H$  maps its domain to some subset  $Y \subset X_e$  that forms a specific geometric structure. As mentioned in [103], the vast majority of physical systems fall into the *conic* system category. In the strict sense, the system is said to be conic if it exhibits the following properties.

**Conicity.** A mapping  $H \subset \mathcal{R}$  is interior conic, if there exist constants  $r > 0$ ,  $c \in \mathbb{R}$  such that the following inequality holds:

$$\|(Hx)_t - cx_t\| \leq r \|x_t\| \quad (2.1)$$

for all  $x \in Do(H)$  and  $t \in T$ . If the inequality sign is reversed the system is said to be exterior conic. The constant  $r$  corresponds to the radius of the cone, whereas the constant  $c$  corresponds to the center parameter.

Considering this equation in  $X_e$  space, it means that systems  $H$  that satisfy the inequality should map the inputs  $x$  to the outputs  $(Hx)_t$  in a specific and strictly defined way. More precisely, outputs  $(Hx)_t$  should always fall into a sphere with following properties:

- center of the sphere is located along the vector  $x_t$ , and its distance to  $x_t$  is proportional to the norm  $\|x_t\|$ .
- radius of the sphere is proportional to the norm of truncated input  $\|x_t\|$ .

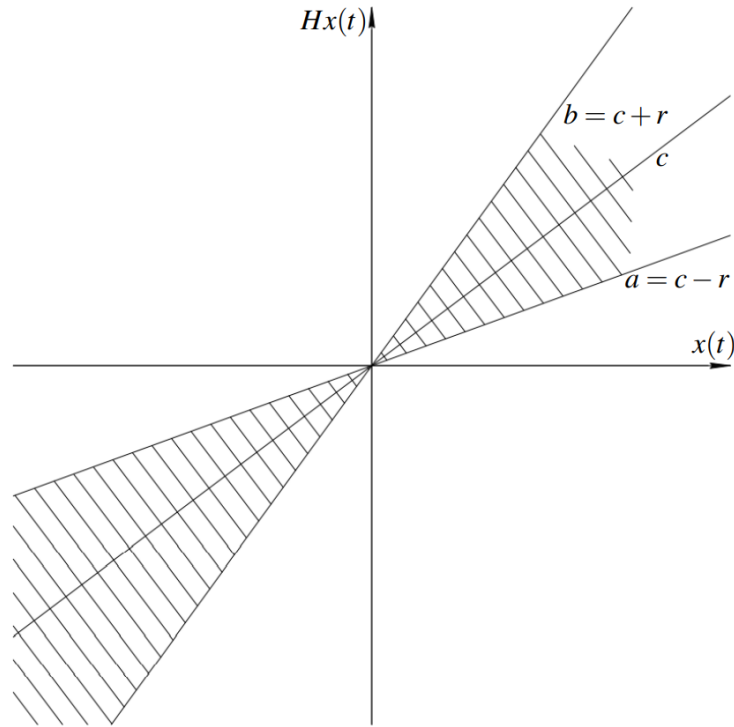


Figure 2.1: A conic sector in the plane. Interior of sector is shaded.

Thus, if we fix the direction of the vector  $x_t$ , the spheres  $(Hx)_t$ , that depend on  $x_t$  could always be inscribed in a cone. A cone's axis would then be located along the direction of  $x_t$ . The radius of the cone would be determined by the constants  $c$  and  $r$ .

There is also the other way of considering a relation  $H$ . This mapping can be examined in the product space  $X_e \times X_e$ . That is, a set of ordered pairs  $(x_t, (Hx)_t)$  for all  $x_t \in Do(H)$  and  $t \in T$  will form a cone if the relation  $H$  is conic.

However, the most convenient way of representing a conic system is by visualizing a relation  $H$  on a 2D plane. This could be done in the following way. Suppose  $H$  is a relation on Lebesgue space with Euclidean norm  $L_{2e}$ . Further, let  $Hx(t)$  be the value assumed by the function  $Hx$  at time  $t \in T$ . In other words,  $Hx(t)$  is a function of  $x(t)$ , say  $Hx(t) = N(x(t))$ . Then the function  $N(\cdot)$  can be represented as a graph on a 2D plane. Depending on whether a system is interior or exterior conic, this graph will be located either inside or outside the conic sector. The slope of the center of a conic sector is equal to constant  $c$ . The boundaries of a cone

are defined by the lines with slopes  $c - r$  and  $c + r$ , as shown in Figure 2.1. More generally, the necessary and sufficient condition for a system to be, for example, interior conic is that function  $Hx(t)$  maps all inputs  $x \in Do(H)$  for all times  $t \in T$  to a conic sector.

Inequality (2.1) can be also expressed in the form  $\|(Hx)_t - cx_t\| \leq r \|x_t\|$ . If we express norms as inner products, then, after factoring we get the following:

$$\langle (Hx)_t - ax_t, (Hx)_t - bx_t \rangle \leq 0 \quad (2.2)$$

where  $a = c - r$  and  $b = c + r$ . If inequality (1.1) holds for all  $x \in Do(H)$  and  $t \in T$  then system  $H$  is interior conic. A system is exterior conic if the inequality (2.2) holds with the inequality sign reversed.

## 2.1.2 Conic Systems as Dissipative Systems

Conic systems can be viewed as dissipative systems with a specific supply rate parameterized by the radius and the center of a given cone [68]. This can be done as follows: Suppose we have a system of the form

$$\Sigma : \begin{cases} \dot{x} = f(x, \eta) \\ y = h(x, \eta) \end{cases} \quad (2.3)$$

where  $x \in \mathbb{R}^n$  is a state of the system,  $\eta, y \in \mathbb{R}^m$  are the input and output,  $f(\cdot, \cdot), h(\cdot, \cdot)$  are locally Lipschitz continuous maps of corresponding dimensions. The input is a function of time  $\eta : t \rightarrow \mathbb{R}^m$  where  $t \in T$ . The control inputs  $\eta(t)$  belong to  $\mathcal{U}_{x_0}^{[t_0, t_1]} \subset X_e$  which denotes a set of Lebesgue measurable locally essentially bounded functions. The solutions  $x(t)$  of a system (2.3) given any inputs  $\eta(t)$  from the set  $\mathcal{U}_{x_0}^{[t_0, t_1]}$  are well-defined for all  $t \in T$ . A system (2.3) is said to be dissipative if there exists a storage function  $V : \mathbb{R}^n \rightarrow \mathbb{R}_+$  such that the

inequality

$$V(x(t_1)) - V(x(t_0)) \leq \int_{t_0}^{t_1} \omega(y(\tau), \eta(\tau)) d\tau \quad (2.4)$$

holds along the trajectories of system (2.3) for all inputs  $\eta \in \mathcal{W}_{x_0}^{[t_0, t_1]}$ , all  $t_0, t_1 \in T$  such that  $t_1 \geq t_0$ , and any initial state  $x(t_0)$ . For ease of presentation, storage function  $V(x)$  is assumed to be radially unbounded. That is,  $V(x) \rightarrow \infty$  as  $\|x\| \rightarrow \infty$ , *i.e.*, storage function grows without bound as  $x$  goes away from the origin. Depending on the choice of supply rate function, the type of dissipative system can be further specified. Two well-know types of dissipative systems are passive and  $L_2$ -gain stable with gain less than or equal to  $\gamma > 0$ . Specifically, a system is said to be passive if it is dissipative with supply rate  $\omega(y, \eta) = y^T \eta$ . A dissipative system is finite  $L_2$ -gain stable if  $\omega(y, \eta) = (\gamma \|\eta\|^2 - \|y\|^2) / \gamma / 2$ . Conic systems, in turn, constitute a more general notion that also includes dissipative systems with other types of supply rates. The supply rate of a conic system can be defined in the following two ways.

**Definition (supply rate for conic systems).** Following inequality (2.2), a supply rate of an interior conic system can be defined as

$$\omega(y, \eta) = (b\eta - y)^T (y - a\eta) \quad (2.5)$$

where  $a, b \in \mathbb{R} \cup \{\pm\infty\}, a \leq b$ . A conic system is said to be exterior conic if  $\omega(y, \eta) = -(b\eta - y)^T (y - a\eta)$ .

The representation of supply rate  $\omega(\cdot)$  as in equation (2.5) may become ambiguous when infinite values of  $a$  or  $b$  may be required to describe certain types of systems. For example, if a conic system is passive,  $a = 0$  and  $b = +\infty$ . The following definition introduces a more convenient representation of supply rate function for conic systems.

**Definition (another form of supply rate for conic systems).** A system of the form (2.3) is

said to be interior conic if it is dissipative with supply rate

$$\omega(y, \eta) = \begin{bmatrix} \eta^T & y^T \end{bmatrix} W(\phi_c, \phi_r) \begin{bmatrix} \eta \\ y \end{bmatrix} \quad (2.6)$$

where  $\phi_c \in \mathbb{R}$  is a center of cone,  $\phi_r \in [0, \pi/2]$  is an angle defining radius of a cone. Matrix  $W(\cdot)$  is defined as

$$W(\phi_c, \phi_r) = \quad (2.7)$$

$$\frac{1}{2 \sin 2\phi_r} \cdot \begin{bmatrix} (\cos 2\phi_c - \cos 2\phi_r)\mathbb{I} & \sin 2\phi_c \mathbb{I} \\ \sin 2\phi_c \mathbb{I} & -(\cos 2\phi_c - \cos 2\phi_r)\mathbb{I} \end{bmatrix} \quad (2.8)$$

where  $\mathbb{I}$  is a unit matrix of corresponding dimensions. A system (2.3) is said to be exterior conic with respect to a cone with radius  $\phi_r$  and center  $\phi_c$  if it is dissipative with supply rate  $\omega(y, \eta) = -[\eta^T, y^T]W(\phi_c, \phi_r)[\eta^T, y^T]^T$ , where  $W(\phi_c, \phi_r)$  is defined as in (2.6).

Throughout the rest of this Chapter, a shorthand notation  $\Sigma \in \text{Int}(\phi_c, \phi_r)$  will be used to represent an interior conic system with center  $\phi_c$  and radius  $\phi_r$ . Similarly,  $\Sigma \in \text{Ext}(\phi_c, \phi_r)$  will denote exterior conic system. A number of useful observations can be made with regards to conic systems described above.

**Remark 1.** Suppose the parameters of a conic system is defined by parameters  $a$  and  $b$ , where  $a, b \in \mathbb{R} \cup \{\pm\infty\}, a \leq b$ . Then the center of the cone  $\phi_c \in \mathbb{R}$  and the radius  $\phi_r \in [0, \pi/2]$  can be found using the equation below

$$\phi_c = \frac{\tan^{-1} a + \tan^{-1} b}{2}, \quad \phi_r = \frac{\tan^{-1} b - \tan^{-1} a}{2} \quad (2.9)$$

**Remark 2.** An interior conic system with a certain center and radius is exterior with respect to a different center and radius. Similarly, an exterior conic system is interior with respect to different center and radius. That is, a system  $\Sigma \in \text{Int}(\phi_c, \phi_r)$  with  $\phi_r \in [0, \pi/2]$  is equivalent to

a system  $\Sigma \in \text{Ext}(\phi'_c, \phi'_r)$  where  $\phi'_c = \phi_c \pm \pi/2$  and  $\phi'_r = \pi/2 - \phi_r$ . This can be shown directly by substituting  $\phi'_c, \phi'_r$  for  $\phi_c, \phi_r$  to equation (2.6). Given that, it is reasonable to consider only one type of system, because if a statement is valid for an exterior conic system it will be also valid for an interior one, and vice versa. •

**Remark 3.** From the supply rate equation (2.6), it can be seen that a system  $\Sigma \in \text{Int}(\phi_{c1}, \phi_r)$  is equivalent to  $\Sigma \in \text{Int}(\phi_{c2}, \phi_r)$  whenever  $\phi_{c1} - \phi_{c2} = k\pi$  where  $k$  is an integer ( $k \in \mathbb{Z}$ ). Therefore, it would be more practical to consider conic center  $\phi_c$  to be an element of equivalence class in a quotient set  $\mathbb{R} \setminus \Pi$ . The equivalence relation  $\Pi$  can be defined according to the formula  $\Pi = \{\phi_1 \sim \phi_2 \text{ iff } \phi_1 - \phi_2 = k\pi, k \in \mathbb{Z}\}$ . Thus, any interval of the form  $(a, a + \pi]$  will have exactly one element from each of the equivalence classes in the quotient set  $\mathbb{R} \setminus \Pi$ ; such an element can be denoted as  $\phi_{(a, a + \pi]}$ . •

**Remark 4.** Suppose a system (2.3) is also a conic system  $\Sigma \in \text{Int}(\phi_c, \phi_r)$ . The inverse system  $\Sigma^{-1}$  can be defined as a system with flipped input and output signals, where  $\eta$  is considered an output and  $y$  as an input. Considering the equation

$$\begin{bmatrix} y \\ \eta \end{bmatrix}^T W(\phi_c, \phi_r) \begin{bmatrix} y \\ \eta \end{bmatrix} = \begin{bmatrix} \eta \\ y \end{bmatrix}^T W(\pi/2 - \phi_c, \phi_r) \begin{bmatrix} \eta \\ y \end{bmatrix} \quad (2.10)$$

that can be directly verified from (2.6), it can be concluded that the inverse system  $\Sigma^{-1} \in \text{Int}(\phi_c, \phi_r)$ . •

**Remark 5.** Both passive and finite  $L_2$ -gain stable systems can be defined in terms of conic relations parametrized with  $\phi_c$  and  $\phi_r$ . In particular, any passive system is also interior conic  $\Sigma = \text{Int}(\phi_c, \phi_r)$  with  $\phi_c = \pi/4$  and  $\phi_r = \pi/4$ . Similarly, any finite  $L_2$ -gain stable system with gain less than or equal to  $\gamma \geq 0$  is also interior conic with respect to a cone center  $\phi_c = 0$  and radius  $\phi_r = \tan^{-1} \gamma$ . •

## 2.2 Non-Planar Conicity and Scattering Transformation

In the previous section, the notion of planar conic systems was defined. As shown for example in [68], methods based on the scattering transformation for planar conic systems can be used to stabilize interconnections of systems. Despite the fact that many physical systems fall into a class of planar conic systems, this representation of systems has certain limitations. First, any conic system is parameterized with only two scalar values: conic center  $\phi_c$  and radius  $\phi_r$ . Such systems came to be called planar conic systems because of the fact that their dynamics can be presented as a conic sector on a plane. The notion of planar conic systems is fairly general; that is, it includes different types of passive systems, finite  $L_2$ -gain systems, *etc.*, as special cases. This description also lacks flexibility, which in turn limits its range of applications. Second, apart from finite  $L_2$ -gain stable systems, the dimension of inputs has to be equal to the dimension of outputs in this type of conic systems, which in turn further limits the applications of this method. Lastly, another substantial limitation of planar conicity is that the feedback interconnection of two planar conic systems is, generally speaking, not a planar conic system. Description of an overall system in that case can be a non-trivial task. Thus, this fact makes it difficult to analyze complex interconnections using the notion of planar conicity.

In this section, an extension of conicity notion to non-planar case [87] is defined, which removes all the limitations described above. This approach generalizes planar conic systems to the case when a center of a cone has a dimension greater than one. The set of supply rates of non-planar conic systems coincides with that of  $(Q, S, R)$ -dissipative systems. In particular, for a given quadratic supply rate, parameters defining a cone of a system can be calculated using procedure described later in this section. In addition, for a given interconnection of two non-planar systems, a graph separation condition for finite  $L_2$  gain stability is defined. This condition is derived in terms of the relation between the radii of subsystems' cones and the maximal singular value of the product of the projection operators onto central subspaces. Subsequently, a new generalized scattering transformation is described that allows for rendering the dynamics of a non-planar conic system into a cone with prescribed parameters. This, in turn, allows for



stabilization of feedback interconnections by modifying parameters of the subsystems' cones. Scattering transformations derived for non-planar conic systems can, in particular, be applied to the problems of stable robot-environment interaction or bilateral teleoperation.

This section is organized as follows. First, the notion of non-planar systems is defined, and the procedure for calculation of cone parameters is described. Next, graph separation condition for stability of feedback interconnection of non-planar conic systems is defined. Later, the generalized scattering transformation that renders a non-planar conic system into a cone with prescribed parameters is presented. Lastly, a procedure for stabilization of interconnected systems is described.

### 2.2.1 Non-Planar Conicity

Consider a nonlinear system of the form

$$\Sigma : \begin{cases} \dot{x} = f(x, \eta) \\ y = h(x, \eta) \end{cases} \quad (2.11)$$

where  $x \in \mathbb{R}^n$  is the state of the system,  $\eta \in \mathbb{R}^m$  is input, and  $y \in \mathbb{R}^p$  is the output of the system. The functions  $f(\cdot)$  and  $h(\cdot)$  are locally Lipschitz continuous in their arguments. A system (2.11) is said to be dissipative with respect to supply rate  $\omega : \mathbb{R}^p \times \mathbb{R}^m \rightarrow \mathbb{R}$  if there exists a storage function  $V : \mathbb{R}^n \rightarrow \mathbb{R}_+$  such that the inequality

$$V(x(t_1)) - V(x(t_0)) \leq \int_{t_0}^{t_1} \omega(y(\tau), \eta(\tau)) d\tau \quad (2.12)$$

holds along the trajectories of the system (2.11) for any  $t_1 \geq t_0$ , any initial state  $x(t_0)$ , and arbitrary admissible control input  $\eta(t)$  where  $t \in [t_0, t_1]$ . Throughout this thesis, it is assumed that all storage functions are radially unbounded, *i.e.*,  $V(x) \rightarrow \infty$  as  $|x| \rightarrow \infty$ . There is a notable difference between system (2.11) and a system used in derivations of planar conic systems (2.3). The difference is that in inequality (2.3) input and output signals have to have the same dimen-

sions, whereas inequality (2.11) allows input and output signals to have different dimensions.

Now, let us derive supply rate for non-planar conic systems. In the development below, the parameter  $\lambda$  can be chosen to be equal to  $\lambda = \sqrt{(a^2 + 1)(b^2 + 1)}$ , where the parameters  $a$  and  $b$  define slopes of the conic sector (2.2). A matrix  $W(\phi_c, \phi_r)$  of the quadratic supply rate used in planar conic system developments (2.6) can also be written in the form

$$W(\phi_c, \phi_r) = \lambda \cdot [l_c l_c^T - \cos^2 \phi_r \mathbb{I}_2] \otimes \mathbb{I}_m \quad (2.13)$$

where  $l_c = [\cos \phi_c \sin \phi_c]^T$  is the unit vector that lies on the subspace defining a center of cone. The above representation of matrix  $W(\cdot, \cdot)$  allows for an extension to a non-planar case. This can be done in the following way. To simplify the derivations, suppose that  $\lambda = 1$  and  $m = 1$ . If we substitute expression (2.11) to (2.6), supply rate can be written in the following form

$$\omega(y, \eta) = \begin{bmatrix} \eta^T & y^T \end{bmatrix} l_c l_c^T \begin{bmatrix} \eta \\ y \end{bmatrix} - \cos^2 \phi_r \cdot \left\| \begin{bmatrix} \eta \\ y \end{bmatrix} \right\|^2. \quad (2.14)$$

Since  $l_c$  is the unit vector lying on the central subspace, the dot product of  $\begin{bmatrix} \eta^T & y^T \end{bmatrix}^T$  and  $l_c$  corresponds to the length of projection of input-output pair onto the central subspace. Based on this observation, the supply rate (2.14) can be rewritten in the following form

$$\omega(y, \eta) = \begin{bmatrix} \eta^T & y^T \end{bmatrix} \left[ \Pi_c^T \Pi_c - \cos^2 \phi_r \mathbb{I}_2 \right] \begin{bmatrix} \eta^T & y^T \end{bmatrix}^T \quad (2.15)$$

where

$$\Pi_c = \begin{bmatrix} \cos^2 \phi_c & \sin \phi_c \cos \phi_c \\ \sin \phi_c \cos \phi_c & \sin^2 \phi_c \end{bmatrix} \quad (2.16)$$

is the matrix that projects vectors onto the center of the cone. Since the matrix  $\Pi_c$  is a projection matrix it is symmetric ( $\Pi_c^T = \Pi_c$ ) and indempotent ( $\Pi_c^2 = \Pi_c$ ); therefore,  $\Pi_c^T = \Pi_c$  and supply

rate can be further simplified as follows

$$\omega(y, \eta) = \begin{bmatrix} \eta^T & y^T \end{bmatrix} \left[ \Pi_c - \cos^2 \phi_r \mathbb{I}_2 \right] \begin{bmatrix} \eta^T & y^T \end{bmatrix}^T. \quad (2.17)$$

The above equation allows for the generalization of planar conic systems supply rate (2.6) in the two following directions. First of all, the dimensions of the input and output of a conic system represented by (2.17) may not be equal. Secondly, a center of a conic system may have a dimension higher than one. These two improvements allow to design methods that have wider ranger of applications compared to methods based on planar conicity. Formally, generalization of interior conicity to a non-planar case, when dimensions of the input and output are not equal  $m \neq p$ , is presented in the definition below.

**Definition (Supply rate of non-planar conic system).** Suppose the input of the system (2.11)  $\eta \in \mathbb{R}^m$  and output  $y \in \mathbb{R}^p$ . Then input-output vector  $\begin{bmatrix} \eta^T & y^T \end{bmatrix}^T \in \mathbb{R}^{m+p}$ . Consider a subspace of input-output space  $\Omega \subset \mathbb{R}^{m+p}$ ,  $\dim \Omega = l \in \{0, \dots, m+p\}$ , and radius of a cone  $\phi_r \in [0, \pi/2)$ . A system  $\Sigma$  of the form (2.11) is said to be interior conic with respect to a center  $\Omega$  and radius  $\phi_r$ , in other words,  $\Sigma \in \text{Int}(\Omega, \phi_r)$ , if it is dissipative with supply rate

$$\omega(y, \eta) = \begin{bmatrix} \eta^T & y^T \end{bmatrix} W(\Omega, \phi_r) \begin{bmatrix} \eta^T & y^T \end{bmatrix}^T, \quad (2.18)$$

where matrix  $W(\Omega, \phi_r)$  is defined as

$$W(\Omega, \phi_r) = \Pi_\Omega - \cos^2 \phi_r \mathbb{I}_{m+p}, \quad (2.19)$$

where  $\Pi_\Omega$  is the projection matrix onto central subspace  $\Omega$ .

The general notion defined above in Definition 2.2.1 represents the generalization of planar conic systems (2.6). That is, in the case of Definition 2.2.1, planar conic system is the special case of non-planar conic system. Thus, if  $m = p$  and the central subspace is of the form

$y - \tan \phi_c \cdot \eta = 0$  where  $\phi_c \in (-\pi/2, \pi/2)$ , the supply rate (2.18) would represent a planar conic system.

**Remark 6 (non-planar conicity vs. planar conicity).** Non-planar systems remove several limitations and have higher flexibility compared to planar representation. Even if the dimensions of input and output of the system coincide, *i.e.*,  $m = p$ , non-planar representation is more advantageous compared to planar. This can be shown in the following way: Suppose the dimensions of input and output of system  $\Sigma$  are equal ( $m = p$ ), and center of a cone  $\phi_r \in (-\pi/2, \pi/2]$ . Then, such a system can be represented using a notion of planar conicity described in Definition 2.1.2. At the same time system  $\Sigma$  can be represented as a non-planar cone with supply rate given in Definition 2.2.1. The central subspace of system  $\Sigma$  can be defined in the following way

$$\Omega = \text{span} \left\{ \left[ \begin{array}{cc} \cos \phi_c & \sin \phi_c \end{array} \right]^T \otimes \mathbb{I}_m \right\}. \quad (2.20)$$

Given that  $m = p$  the dimension of input-output vector equals to  $2m$ , or, in other words,  $\left[ \begin{array}{cc} \eta^T & y^T \end{array} \right]^T \in \mathbb{R}^{2m}$ . For any  $\phi_r \in (-\pi/2, \pi/2]$ , dimension of central subspace  $\dim \Omega = m$ . Cone center  $\Omega$  is a subspace of input output space; therefore,  $\Omega \subset \mathbb{R}^{2m}$ . Since  $\Sigma$  can be any subspace of  $\mathbb{R}^{2m}$ , we can consider  $\Sigma$  to be belonging to a set of all possible  $m$ -dimensional subspaces of  $2m$  dimensional linear space. Such a set of all possible subspaces can be represented as a Grassmanian manifold  $\text{Gr}(m, n)$ . The dimension of the Grassmanian manifold can be found using the following equation  $\dim \{\text{Gr}(m, n)\} = m(n - m)$ . In our case, the set of all possible  $m$ -dimensional subspaces of  $2m$ -dimensional space forms  $m^2$ -dimensional manifold. However, if we represent a system using a notion of planar conicity, a center of cone is defined using a single scalar value  $\phi_c \in (-\pi/2, \pi/2]$ . Therefore, even in the case where  $m = p$ ,  $m > 1$ , *i.e.*, when a system can be represented as a planar cone, it is still better to follow the notion of a non-planar conicity. Fundamentally, this representation is more flexible and allows for a more precise description of a system's dynamics.

The other advantage of non-planar conicity lies in its relationship with a well-know notion of  $(Q, S, R)$ -dissipativity, studied in [33]. In particular, central subspace  $\Omega$  and radius  $\phi_r$  of interior conic system  $\Sigma \in \text{Int}(\Omega, \phi_r)$  can be found, knowing  $[QSR]$  matrix. The following paragraphs elaborate on this idea.

A system of the form (2.11) is said to be  $(Q, S, R)$ -dissipative if it is dissipative with supply rate

$$\begin{aligned} \omega(y, \eta) &= y^T Q y + 2y^T S \eta + \eta^T R \eta \\ &= \begin{bmatrix} \eta^T & y^T \end{bmatrix} \begin{bmatrix} QSR \end{bmatrix} \begin{bmatrix} \eta \\ y \end{bmatrix} \end{aligned} \quad (2.21)$$

where

$$\begin{bmatrix} QSR \end{bmatrix} = \begin{bmatrix} R & S^T \\ S & Q \end{bmatrix} \in \mathbb{R}^{(m+p) \times (m+p)}. \quad (2.22)$$

where matrices  $Q = Q^T \in \mathbb{R}^{p \times p}$ ,  $R = R^T \in \mathbb{R}^{m \times m}$ , and  $S \in \mathbb{R}^{p \times m}$ . All interior conic systems in the sense of Definition 3 are also  $(Q, S, R)$ -dissipative. Similarly, if a system is  $(Q, S, R)$ -dissipative then it is also non-planar conic.

Matrix  $\begin{bmatrix} QSR \end{bmatrix}$  is real symmetric; thus, its eigenvalues  $\mu_1, \dots, \mu_{p+m}$  are all real. Consider the following

$$G^T \cdot \begin{bmatrix} QSR \end{bmatrix} \cdot G = \text{diag}[\mu_1, \dots, \mu_{p+m}], \quad (2.23)$$

where  $\mu_1, \dots, \mu_{p+m}$  are the eigenvalues of  $\begin{bmatrix} QSR \end{bmatrix}$  written in an arbitrary prescribed order; in addition, considering the nature of the matrix  $\begin{bmatrix} QSR \end{bmatrix}$ , all its eigenvalues are real. The matrix  $G$  is a real orthogonal matrix, composed in such a way that its  $i$ th column is an eigenvector of  $\begin{bmatrix} QSR \end{bmatrix}$  that corresponds to  $i$ th eigenvalue  $\mu_i, i = 1, \dots, m + p$ . Now, let  $\lambda(QSR) =$

$\{\mu_i, i = 1, \dots, m+p\}$  denote a set of eigenvalues of  $\begin{bmatrix} QSR \end{bmatrix}$ ,  $\lambda^-(QSR) \subset \lambda(QSR)$  the set of strictly negative ( $< 0$ ) eigenvalues of  $\begin{bmatrix} QSR \end{bmatrix}$  and  $\lambda^+(QSR) = \lambda(QSR) \setminus \lambda^-(QSR)$  the set of nonnegative ( $\geq 0$ ) eigenvalues of  $\begin{bmatrix} QSR \end{bmatrix}$ . Next, let us introduce the following notations

$$l = \text{card} \{ \lambda^+(QSR) \}, l \in \{0, \dots, m+p\}, \quad (2.24)$$

$$\mu^- = \min \{ |\mu_i| : \mu_i \in \lambda^-(QSR) \}, \quad (2.25)$$

$$\mu^+ = \max \{ |\mu_i| : \mu_i \in \lambda^+(QSR) \}, \quad (2.26)$$

where  $l$  is the number of nonnegative eigenvalues of  $\begin{bmatrix} QSR \end{bmatrix}$ . The value of  $\mu^-$  is well-defined if  $\lambda^-(QSR) \neq 0$ , or, in other words, if  $l < m+p$ . Similarly, the value of  $\mu^+$  is well-defined if  $\lambda^+(QSR) \neq 0$ , i.e., if  $l > 0$ . The following statement is valid.

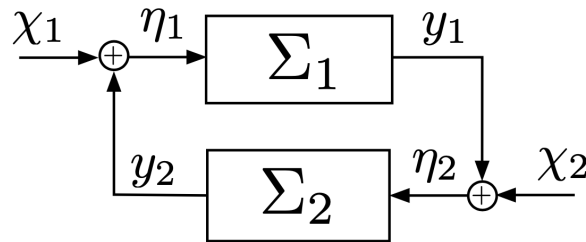
**Lemma 2.2.1 (Parameters of a non-planar cone [87]).** *Suppose a system (2.11) is  $(Q, S, R)$ -dissipative. Then it is non-planar interior conic in the sense of Definition 3  $\Sigma = \text{Int}(\Omega, \phi_r)$ . System  $\Sigma$  has a center  $\Omega \subset \mathbb{R}^{m+p}$ ,  $\dim \Omega = l$ , and radius  $\phi_r \in [0, \pi/2)$ . In particular, conic center  $\Omega$  can be found using the following expression*

$$\Omega = \text{span} \{ g_1^+, \dots, g_l^+ \} \quad (2.27)$$

which is essentially a subspace spanned by eigenvectors  $g_1^+, \dots, g_l^+$  of matrix  $\begin{bmatrix} QSR \end{bmatrix}$  that correspond to its nonnegative eigenvalues  $\mu_i \in \lambda^+(QSR)$ . If  $0 < l < m+p$ , conic radius can be found the following way

$$\phi_r = \tan^{-1} \left( \sqrt{\mu^+/\mu^-} \right), \quad (2.28)$$

If condition  $0 < l < m+p$  is not met, i.e., if  $l = 0$  or  $l = m+p$ , radius  $\phi_r$  can be chosen arbitrarily from the range  $(0, \pi/2)$ .

Figure 2.2: Feedback interconnection of systems  $\Sigma_1$  and  $\Sigma_2$ .

### 2.2.2 Graph Separation Stability Condition for Non-Planar Conic Systems

In this section, a graph separation stability condition for the interconnection of two non-planar conic systems will be formulated. In this work, we will address finite gain  $L_2$ -stability. A system of the form (2.11) is said to be finite gain  $L_2$ -stable if it is dissipative with supply rate  $\omega(y, \eta) = \gamma^2 |\eta|^2 + |y|^2$ , where  $\gamma \geq 0$  is the  $L_2$ -gain, see [89]. Finite gain  $L_2$ -stability of a feedback interconnection of two non-planar conic subsystems shown in Figure 2.2 can be guaranteed by a graph separation condition which was originally developed in [85, 87]. This stability condition is based on the dynamic cone parameters of the subsystems. These parameters, *i.e.*, the central subspace and radius, can be found using quadratic supply rate of the subsystem as shown in Lemma 6.

To formulate the graph separation stability condition, it is convenient to use the notion similar to the one of the inverse graph [79]. Thus, a conic system with certain center  $\Omega$  and radius  $\phi_r$  is called inverse interior conic, if the same system with swapped inputs and outputs is interior conic  $\text{Int}(\Omega, \phi_r)$  with the same parameters center and radius. In other words, the conic parameters of inverse conic system stay the same even if we have  $y$  on input and  $\eta$  on output. The following paragraphs elaborate on this idea.

Suppose we are given a central subspace  $\Omega \subset \mathbb{R}^{m+p}$ ,  $\Omega = \text{span}\{\omega_1, \dots, \omega_m\}$ ,  $\dim \Omega = m$ ,

*i.e.*, the vectors  $\omega_1, \dots, \omega_m$  form a basis in  $\Omega$ . Consider the following space

$$\bar{\Omega} = \text{span} \{P_{(m,p)}\omega_1, \dots, P_{(m,p)}\omega_m\}, \quad (2.29)$$

where  $P_{(m,p)} \in \mathbb{R}^{m+p}$  is a permutation matrix defined as follows:

$$P_{(m,p)} = \begin{bmatrix} \mathbb{O} & \mathbb{I}_p \\ \mathbb{I}_m & \mathbb{O} \end{bmatrix}. \quad (2.30)$$

Thus, a system of the form (2.11) is considered to be inverse interior conic with center  $\Omega$  and radius  $\phi_r$  if and only if it is interior conic with respect to center  $\bar{\Omega}$  and radius  $\phi_r$ . Multiplication of the input-output vector by  $P_{(m,p)}$  only swaps inputs and outputs, *i.e.*,  $P_{(m,p)} \cdot [\eta^T y^T]^T = [y^T \eta^T]^T$ . In addition, projection matrix  $\Pi_\Omega$  is related to  $\Pi_{\bar{\Omega}}$  according to the formula  $\Pi_{\bar{\Omega}} = P_{(m,p)}\Pi_\Omega P_{(m,p)}^T$ .

Consider two nonlinear systems of the form

$$\Sigma_i : \begin{cases} \dot{x}_i = f_i(x_i, \eta_i), \\ y_i = h_i(x_i, \eta_i), \end{cases} \quad i \in \{1, 2\} \quad (2.31)$$

where  $y_2, \eta_1 \in \mathbb{R}^m$  and  $y_1, \eta_2 \in \mathbb{R}^p$ . Suppose systems  $\Sigma_1$  and  $\Sigma_2$  are interconnected in the following way

$$\eta_1 = y_2 + \chi_1, \quad \eta_2 = y_1 + \chi_2 \quad (2.32)$$

where  $\chi_1 \in \mathbb{R}^m, \chi_2 \in \mathbb{R}^p$  are disturbance inputs. The architecture of the overall system is presented on Figure 2.2. This system has input  $[\chi_1^T, \chi_2^T]^T \in \mathbb{R}^{m+p}$  and output  $[y_1^T, y_2^T]^T \in \mathbb{R}^{m+p}$ . Consider the following theorem.

**Theorem 2.2.2 (Graph Separation Condition [78, 85]).** *Suppose two subsystems of the form (2.31) are interconnected according to equations (2.32). If both systems are interior*



conic  $\Sigma_1 \in \text{Int}(\Omega_1, \phi_{r1})$ ,  $\Sigma_2 \in \text{Int}(\Omega_2, \phi_{r2})$ , where  $\overline{\Omega_1} \cap \Omega_2 = \{0\}$ ,  $\dim \Omega_1 = m$ ,  $\dim \Omega_2 = p$ , and if the following condition is satisfied

$$\sigma \left( \Pi_{\overline{\Omega_1}}, \Pi_{\Omega_2} \right) < \cos(\phi_{r1} + \phi_{r2}) \quad (2.33)$$

then the interconnected system (2.31), (2.32) is finite gain  $L_2$ -stable.

The proof for Theorem 2.2.2 can be found in Sections 4 and 5 of conference paper [85]. It should be noted that the condition for stability defined above assumes that dimension of central subspace should be equal to the dimension of system's input. In other words,  $\dim \Omega_1 = \dim \eta_1 = m$  and  $\dim \Omega_2 = \dim \eta_2 = p$ . This requirement is apparently necessary to exclude meaningless and/or overly conservative cases. More precisely, let us consider the case when  $\dim \Omega_1 < \dim \eta_1$  or  $\dim \Omega_2 < \dim \eta_2$ . This assumption imposes restrictions on values of input signals  $\eta_1(t)$ ,  $\eta_2(t)$ . In the case where  $\dim \Omega_1 + \dim \Omega_2 > m + p$ , the graph separation is impossible. However, these issues will be studied in detail in our future research.

## 2.3 Scattering Transformation for Non-Planar Conic Systems

A scattering transformation enables input-output characteristics of subsystems to be transformed in such a way that the graph separation condition is satisfied. More precisely, scattering transform enables the desired center and radius of the non-planar cone of the system to be prescribed. This operator essentially performs planar rotation and scaling of the input-output vector. Thus, it transforms a passive system into a system with gain less than or equal to one. The scattering transformation presented in [86, 87] allows for rendering of the input-output characteristics of a non-planar conic system into an arbitrary prescribed cone with equal dimension of the central subspace. Specifically, let us consider a system  $\Sigma$  of the form (2.11). Suppose this system is non-planar interior conic  $\Sigma \in \text{Int}(\Omega, \phi_r)$  where center  $\Omega \subset \mathbb{R}^{m+p}$ ,  $\dim \Omega = m$  and radius  $\phi_r \in (0, \pi/2)$ . The desired central subspace and radius are given as  $\Omega_d \subset \mathbb{R}^{m+p}$ ,  $\dim \Omega_d = m$

and  $\phi_{rd} \in (0, \pi/2)$ , correspondingly. Thus, a scattering operator should transform the input-output signal in the following way

$$\begin{bmatrix} u^T & v^T \end{bmatrix}^T = \mathbb{S}(\Omega, \Omega_d, \phi_r, \phi_{rd}) \begin{bmatrix} \eta^T & y^T \end{bmatrix}^T. \quad (2.34)$$

where  $\begin{bmatrix} u^T & v^T \end{bmatrix}^T$  is the new input-output vector of system  $u \in \mathbb{R}^m, v \in \mathbb{R}^p$ . This transformation should render input-output characteristics into a cone with center  $\Omega_d$  and radius  $\phi_{rd}$ . That is, scattering transforms an interior conic system  $\Sigma \in \text{Int}(\Omega, \phi_r)$  to a system  $\Sigma_{(u,v)} \in \text{Int}(\Omega_d, \phi_{rd})$ .

An operator with the above described properties can be constructed using the following procedure. First, suppose that vectors  $g_1, g_2, \dots, g_m$  form orthonormal basis in central space  $\Omega$ . Now, suppose that there is a set of vectors  $\{g_{m+1}, \dots, g_{m+p}\} \in \Omega^\perp$  that together with basis  $\{g_1, g_2, \dots, g_m\} \in \Omega$  forms orthonormal basis in the input-output space  $\mathbb{R}^{m+p}$ . Let us denote the basis of input-output space using the following matrix

$$G = \begin{bmatrix} g_1 & \dots & g_m & g_{m+1} & \dots & g_{m+p} \end{bmatrix}. \quad (2.35)$$

Similarly, a matrix  $G_d$  can be constructed such that its first  $m$  columns form an orthonormal basis in  $\Omega_d$ , while the whole set of its columns form an orthonormal basis in  $\mathbb{R}^{m+p}$ . Next, consider the following transformation

$$\mathbb{S}(\Omega, \Omega_d, \phi_r, \phi_{rd}) = G_d \cdot \Gamma(\phi_r, \phi_{rd}) \cdot G^T \quad (2.36)$$

where

$$\begin{aligned} \Gamma(\phi_r, \phi_{rd}) &= \begin{pmatrix} \cos \phi_{rd} \\ \cos \phi_r \end{pmatrix}^\alpha \cdot \begin{pmatrix} \sin \phi_{rd} \\ \sin \phi_r \end{pmatrix}^{-\beta} \\ &\times \begin{bmatrix} \left(\frac{\tan \phi_{rd}}{\tan \phi_r}\right)^\alpha \mathbb{I}_m & \mathbb{O}_{mp} \\ \mathbb{O}_{pm} & \left(\frac{\tan \phi_{rd}}{\tan \phi_r}\right)^\beta \mathbb{I}_p \end{bmatrix}, \end{aligned} \quad (2.37)$$

and  $\alpha = -p/(m+p)$ ,  $\beta = m/(m+p)$ . Consider the following Lemma.

**Lemma 2.3.1 (Scattering transformation for non-planar conic systems [86])** *Suppose that system  $\Sigma$  of type (2.11) is interior conic  $\Sigma \in \text{Int}(\Omega, \phi_r)$ , where  $\Omega \subset \mathbb{R}^{m+p}$ ,  $\dim \Omega = m$ , and  $\phi_r \in (0, \pi/2)$ . Then the transformed system (2.11), (2.34), (2.36), (2.37) with new input-output variables  $(u, v)$  satisfies  $\Sigma_{(u,v)} \in \text{Int}(\Omega_d, \phi_{rd})$ .*

One special case of Lemma 7 is of particular interest for the problem of stabilization of robot-environment interaction. Suppose subsystems  $\Sigma_i$ ,  $i = 1, 2$  are non-planar conic. To guarantee stability of the feedback interconnection of  $(\Sigma_1, \Sigma_2)$ , one can implement a scattering transformation for one of the subsystem which renders its input-output characteristics into a desired dynamic cone. If the parameters of the desired cone are chosen in a way that guarantees the fulfilment of the graph separation stability condition (Theorem 3), then the interconnection is guaranteed to be finite gain  $L_2$ -stable. Design methods that use scattering transformation of the form (2.36), (2.37) to guarantee stability of the interconnected system can be found in [87].

In regards to the scattering transformation defined using equations (2.34), (2.36), (2.37), it is worth noting the following. Matrices  $G$  and  $G_d$  define central subspaces  $\Omega$  and  $\Omega_d$ . More precisely, columns of matrices  $G$ ,  $G_d$  form orthonormal basis of central subspaces. Since there are no further restrictions on  $G$  and  $G_d$ , this orthonormal basis can be chosen arbitrarily. Therefore, technically, there is an infinite number of bases that can define a given subspace. That is, there is an infinite number of possible transformations that renders a system into a non-planar cone with prescribed parameters. The choice of matrices  $G$  and  $G_d$  may depend on the specific application.

Thus, to stabilize the interconnected system, the following steps should be taken. First, the dynamic cone parameters, *i.e.*, subspace  $\Omega$  and radius  $\phi_r$ , should be calculated for both of the subsystems. This could be done using subsystems' supply rates as shown in Lemma 6. Next, the desired dynamic cone parameters for the one of the systems should be found. This can be done following the graph separation stability condition defined in Theorem 3. Further, a scattering transformation should be calculated (2.34), (2.36), (2.37) and inserted into the

communication channel to transform one of the subsystems parameters into the desired ones. A more detailed design example that uses scattering transformation (2.34), (2.36), (2.37) for the stabilization of two interconnected non-planar conic systems is presented in Chapter 3.

## 2.4 Conclusion

In this Chapter 2, the theoretical background on conic systems and scattering transformation techniques was presented. First, planar conic systems in the form introduced by Zames [103] were described. That is, the notion of planar conicity enables dynamic systems to be represented in the form of a cone defined with two scalar parameters – center  $\phi_c$  and radius  $\phi_r$ . Next, Section 2.2 built on these developments and described a more general representation of systems - so called non-planar conic systems that were introduced in [85, 86]. In addition, this section presented a comparison of the notions of planar vs. non-planar conicity. Further, in Lemma 2.2.1 a method to calculate cone parameters from the supply rate of the system was described. Next, Theorem 2.2.2 describes a graph separation condition for stability. Finally, this chapter introduced a method for design of scattering transformation for stabilization of coupled systems. The core of this method is that the scattering transformation effectively change parameters of one of the subsystem's cones such that the graph separation condition is satisfied. Overall, this chapter presented a theoretical foundation which forms the basis for the implementation of the scattering-based framework for stabilization of robot-environment interaction which is described in the next chapter.

# Chapter 3

## Scattering-Based Design for Coupled Stability

The goal of this chapter is to present the necessary steps for implementation of the scattering based stabilization framework on a real physical system. The chapter begins with Section 3.1 which contains description of the experimental set-up. In Section 3.2, dynamical modeling and estimation of the robotic device used in the experiments are presented. Estimation of the dynamics is performed using two data driven methods, specifically, parameter estimation using Linear Regression, and dynamics estimation using Neural Networks. The environmental dynamics and the adaptive trajectory tracking control algorithm are described in Section 3.3. Analysis of dissipativity properties of the robot and the environment and parameters of the corresponding dynamic cones are derived in Sections 3.4 and 3.5, respectively. The scattering transformation design for stability of robot environment interaction is presented in Section 3.6. The control architecture is summarized in Section 3.7. Section 3.8 contains some concluding remarks.

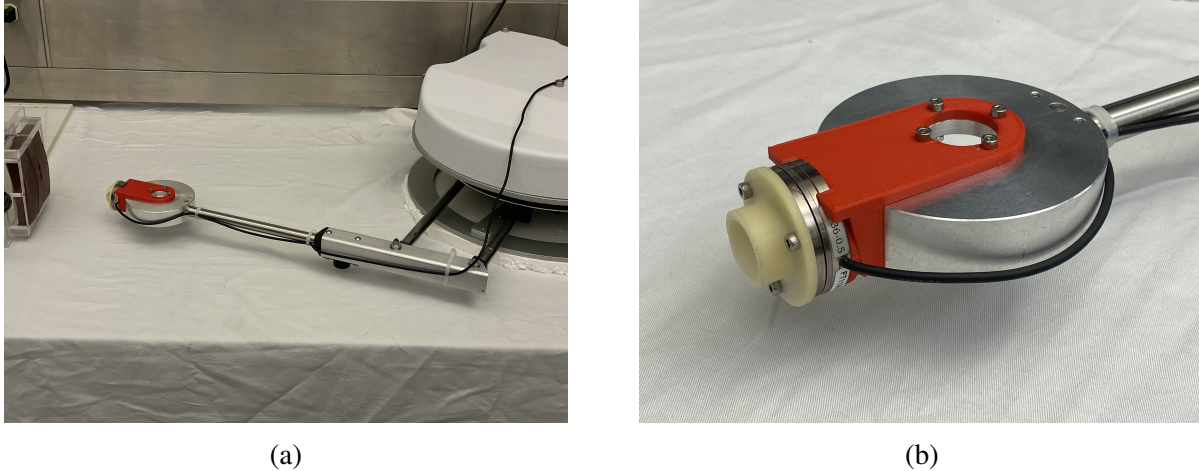


Figure 3.1: Experimental setup: (a) – robot, (b) - force sensor together with the 3D printed mount.

### 3.1 Hardware Description

In our experiments, a 2-DoF cable driven robot manufactured by Quanser was used. The experimental setup is shown in Figure 3.1, while the mechanical structure of the manipulator is presented in Figure 3.2. The robot is driven by DC motors that are controlled by specifying the motors' currents. The design of the built-in motor controller as well as the back EMF of the motors results in substantial kinetic friction. In addition, the robot's joints have relatively high static friction. The rotation angles of the motors are measured by discrete encoders. One full rotation of the robot's joint corresponds to 80,000 counts of the motor's encoder. The encoder values can be read at a maximum frequency of  $2kHz$ . An ATI Nano43 sensor is used to measure the force at the end-effector. The sensor is shown in Figure 3.1(b). The reading frequency of this sensor can be as high as  $10kHz$ , and the resolution is  $1/128N$ .

### 3.2 Manipulator Modeling and Dynamics Estimation

In this section we derive dynamics equations that describe the time evolution of the joints' torque of the robot. Further, we describe and evaluate two data-driven approaches for dealing with unknown robot parameters and unmodeled dynamics.

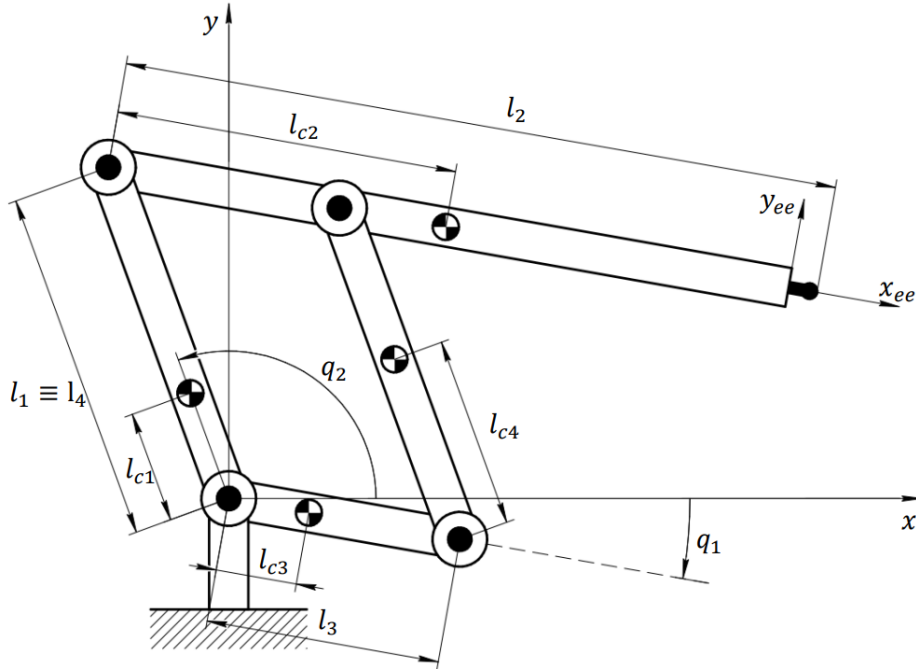


Figure 3.2: Mechanical structure of the five-bar linkage manipulator used in the experiments.

### 3.2.1 Euler-Lagrange Equations

Generally, the dynamics of a rigid-link robot can be described using the Euler-Lagrange equations [29]. This set of differential equations describe the system's dynamics subject to holonomic constraints where the constraint forces satisfy the principle of virtual work. In the general form, the Euler-Lagrange equations can be written as follows

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_k} - \frac{\partial \mathcal{L}}{\partial q_k} = \tau_k, \quad k = 1, \dots, n, \quad (3.1)$$

where  $\mathcal{L}$  is the Lagrangian function of the system defined as follows,

$$\mathcal{L} = K - P, \quad (3.2)$$

where  $K$  and  $P$  are the kinetic and the potential energies of the system, respectively. Since there is no potential energy in our system, the Lagrangian function would simply be equal to  $K$ . Thus, starting from kinetic energy of the system and taking the corresponding derivatives,

one can arrive at the resulting equations as follows:

$$H_p(q)\ddot{\mathbf{x}} + C_p(q, \dot{q})\dot{\mathbf{x}} + F_p(\dot{q}) = f_r + f_e, \quad (3.3)$$

$$f_r = J_p^{-T} \tau, \quad (3.4)$$

where  $q, \dot{q} \in \mathbb{R}^n$  are robot's position and velocity vectors represented in joint space coordinates,  $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}} \in \mathbb{R}^m$  are position, velocity and acceleration, respectively, of the robot's end-effector represented in the task space coordinates,  $H_p(q), C_p(q, \dot{q}), F_p(\dot{q})$  are matrices of inertia, Coriolis and centrifugal forces, and vector of damping forces represented in the task space coordinates,  $f_e$  denotes the force applied at the end-effector,  $f_r$  is the task space control input,  $\tau \in \mathbb{R}^n$  is the vector of joint torques,  $J_p(q)$  is the Jacobian. In the cases when the Jacobian is non square  $J_p \in \mathbb{R}^{m \times n}$ ,  $n \neq m$ , it cannot be inverted directly. Instead, we can construct the right pseudoinverse of  $J_p$  using its singular value decomposition, for details see [32]. Description of the robot's dynamics in the task space significantly simplifies the analysis of their interaction with environment. Similarly to the task space dynamics (3.3), the joint space dynamics have the form

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(\dot{q}) = \tau \quad (3.5)$$

where  $H(q), C(q, \dot{q}), F(\dot{q})$  are matrices of inertia, Coriolis and centrifugal forces, and vector of damping forces, respectively, represented in the joint space coordinates.

### 3.2.2 Manipulator Modeling

The goal of this subsection is to determine a mathematical model of the manipulator used in the experiments (and described in Section 3.1) in the form of Euler-Lagrange equations in joint space coordinates (3.5). Specifically, given the mechanical structure of the robot used in the experiments, we need to derive expressions for functions  $H(\cdot), C(\cdot), F(\cdot)$  that enter the



dynamics equation (3.5). This derivation can be done as follows. First, as can be seen in Figure 3.2, even though the robot has four links, there are in fact only two degrees of freedom. Thus, the links of the manipulator form a closed loop chain. As a first step, let us write down the equations that define the coordinates of the centers of mass of the various links as a function of joint position

$$\begin{aligned}
 \begin{bmatrix} x_{c1} \\ y_{c1} \end{bmatrix} &= \begin{bmatrix} l_{c1} \cos q_2 \\ l_{c1} \sin q_2 \end{bmatrix}, \\
 \begin{bmatrix} x_{c2} \\ y_{c2} \end{bmatrix} &= \begin{bmatrix} l_1 \cos q_2 + l_{c2} \cos q_1 \\ l_1 \sin q_2 + l_{c2} \sin q_1 \end{bmatrix}, \\
 \begin{bmatrix} x_{c3} \\ y_{c3} \end{bmatrix} &= \begin{bmatrix} l_{c3} \cos q_1 \\ l_{c3} \sin q_1 \end{bmatrix}, \\
 \begin{bmatrix} x_{c4} \\ y_{c4} \end{bmatrix} &= \begin{bmatrix} l_3 \cos q_1 + l_{c4} \cos q_2 \\ l_3 \sin q_1 + l_{c4} \sin q_2 \end{bmatrix},
 \end{aligned} \tag{3.6}$$

where  $m_i, l_i$  are link masses and lengths, correspondingly. Next, given the equations above, we can derive the equations defining the task space velocities of the centers of mass of the various links as functions of  $\dot{q}_1$  and  $\dot{q}_2$ . Thus, after taking the corresponding derivatives we get the

following

$$\begin{aligned}
v_{c1} &= J_{v_{c1}} \dot{q} = \begin{bmatrix} 0 & l_{c1} \sin q_2 \\ 0 & -l_{c1} \cos q_2 \end{bmatrix} \dot{q}, \\
v_{c2} &= J_{v_{c2}} \dot{q} = \begin{bmatrix} l_{c2} \sin q_1 & l_1 \sin q_2 \\ -l_{c2} \cos q_1 & -l_1 \cos q_2 \end{bmatrix} \dot{q}, \\
v_{c3} &= J_{v_{c3}} \dot{q} = \begin{bmatrix} l_{c3} \sin q_1 & 0 \\ -l_{c3} \cos q_1 & 0 \end{bmatrix} \dot{q}, \\
v_{c4} &= J_{v_{c4}} \dot{q} = \begin{bmatrix} l_3 \sin q_1 & l_{c4} \sin q_2 \\ -l_3 \cos q_1 & -l_{c4} \cos q_2 \end{bmatrix} \dot{q},
\end{aligned} \tag{3.7}$$

where  $J_{v_{ci}}$ ,  $i \in \{1, \dots, 4\}$  are the velocity Jacobians. The angular velocities of the four links are given by

$$\begin{aligned}
\omega_2 &= \omega_3 = \dot{q}_1 k, \\
\omega_1 &= \omega_4 = \dot{q}_2 k.
\end{aligned} \tag{3.8}$$

Thus, the inertia matrix  $H(q)$  is given by

$$H(q) = \sum_{i=1}^4 m_i J_{v_{ci}}^T J_{v_{ci}} + \begin{bmatrix} I_2 + I_3 & 0 \\ 0 & I_1 + I_4 \end{bmatrix} \tag{3.9}$$

where  $I_i$  are the inertia tensors. If we now substitute from the equation (3.7) into (3.9), after some calculations, we get the equations defining the elements of the inertia matrix  $H(q)$

$$\begin{aligned}
d_{11} &= m_2 l_{c2}^2 + m_3 l_{c3}^2 + m_4 l_3^2 + I_2 + I_3 \\
d_{12} = d_{21} &= (m_2 l_1 l_{c2} + m_4 l_3 l_{c4}) \cos(q_1 - q_2) \\
d_{22} &= m_1 l_{c1}^2 + m_2 l_1^2 + m_4 l_{c4}^2 + I_1 + I_4
\end{aligned} \tag{3.10}$$

where  $m_i$ ,  $l_i$ ,  $I_i$  are link masses, lengths and inertia tensors, correspondingly. The elements of the Coriolis and centrifugal forces matrix  $C(q, \dot{q})$  are defined as follows:

$$c_{kj} = \sum_{i=1}^n c_{ijk}(q) \dot{q}_i \tag{3.11}$$

where

$$\begin{aligned}
c_{111} &= \frac{1}{2} \frac{\partial d_{11}}{\partial q_1} = 0, & c_{121} = c_{211} &= \frac{1}{2} \frac{\partial d_{11}}{\partial q_2} = 0, & c_{122} = c_{212} &= \frac{1}{2} \frac{\partial d_{22}}{\partial q_1} = 0, \\
c_{221} &= \frac{\partial d_{12}}{\partial q_2} - \frac{1}{2} \frac{\partial d_{22}}{\partial q_1} = -(m_2 l_1 l_{c2} + m_4 l_3 l_{c4}) \sin(q_1 - q_2) \cdot \dot{q}_2, \\
c_{112} &= \frac{\partial d_{12}}{\partial q_2} - \frac{1}{2} \frac{\partial d_{22}}{\partial q_1} = (m_2 l_1 l_{c2} + m_4 l_3 l_{c4}) \sin(q_1 - q_2) \cdot \dot{q}_1.
\end{aligned} \tag{3.12}$$

The matrix of friction torques has the following form:

$$F(\dot{q}) = \begin{bmatrix} k_1 \dot{q}_1 \\ k_2 \dot{q}_2 \end{bmatrix} + \begin{bmatrix} b_{11} \text{sign}(\dot{q}_1) \\ b_{21} \text{sign}(\dot{q}_2) \end{bmatrix} + \begin{bmatrix} b_{12} \text{sign}(\dot{q}_1 - \dot{q}_2) \\ b_{22} \text{sign}(\dot{q}_1 - \dot{q}_2) \end{bmatrix}, \tag{3.13}$$

where  $k_i$  and  $b_i$  are the friction coefficients. The first term defines the kinetic friction of motors. The second and third terms define the static frictions in motors and four joints of the robot.

### 3.2.3 Manipulator Dynamics Estimation

As the robot's manufacturer provides neither a dynamical model nor any parameters or specifications, the robot's dynamics and/or parameters need to be estimated. In such a case, one could acquire the dynamics of the robot using a data-driven approach. Data-driven approaches allow

to estimate dynamic equations by interpolating a function based on a set of sample points. Given the joint space dynamics (3.5), a sample point  $k$  can be represented as

$$k \in Q \times \dot{Q} \times \ddot{Q} \times \mathcal{T}, \quad (3.14)$$

where  $Q$ ,  $\dot{Q}$ ,  $\ddot{Q}$ , and  $\mathcal{T}$  are sets of possible instantaneous values of robot positions, velocities, accelerations, and torques, respectively. Thus, the task is now to find the function  $\mathcal{D}$  that maps instantaneous configuration and its first two derivatives to instantaneous joint torques  $\mathcal{D}: (q, \dot{q}, \ddot{q}) \rightarrow \tau$ , where  $(q, \dot{q}, \ddot{q}) \in \mathbb{R}^6$  and  $\tau \in \mathbb{R}^2$ .

Two data-driven approaches for dynamics estimation were implemented and evaluated in this work, based on the linear regression and the neural networks, respectively. Both of these approaches rely on recordings of the robot's movement and supplied torque. The data gathering was done as follows. First, one thousand points were randomly selected from the robot task space. Next, using the inverse kinematics, the corresponding joint space configurations were found. Furthermore, the robot traversed through all of these points under a PD control algorithm. The commanded torque, configurations, and estimations of velocity and accelerations (3.14) were recorded.

### 3.2.4 Linear Regression

Based on dynamical model (3.5), (3.10) - (3.13), robot's parameters can be estimated using the Linear Regression (LR) method. The estimation method using LR was inspired by the work reported in [98]. That is, given a dataset of points  $\{y_i, x_{i1}, \dots, x_{ip}\}_{i=1}^n$ , the LR model assumes that the relation between dependent variables  $y_i$  and vectors of regressors  $x_i$  is linear. The linear regression model is defined as

$$y_i = \theta_1 x_{i1} + \dots + \theta_p x_{ip} + \varepsilon_i, \quad i = 1, \dots, n, \quad (3.15)$$

where  $\theta_i$  are regression coefficients,  $\varepsilon_i$  are unobserved random variables,  $n$  is the number of sample points, and  $p$  is the number of regression coefficients. Equation (3.15) can also be written in the matrix form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \boldsymbol{\varepsilon}, \quad (3.16)$$

where

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{pmatrix} = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ x_{21} & \dots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{np} \end{pmatrix}, \boldsymbol{\theta} = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_p \end{pmatrix}, \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}, \quad (3.17)$$

where  $\mathbf{y} \in \mathbb{R}^n$  is the dependent variable, matrix  $\mathbf{X} \in \mathbb{R}^{n \times p}$  is the set of regressors,  $\boldsymbol{\theta} \in \mathbb{R}^p$  is the vector of regression coefficients, and  $\boldsymbol{\varepsilon} \in \mathbb{R}^n$  is the vector of disturbances. In order to estimate the vector of parameters  $\boldsymbol{\theta}$ , the ordinary least squares (OLS) method is used. This algorithm minimizes the sum of squared residuals, and leads to a closed-form expression for the estimated value  $\hat{\boldsymbol{\theta}}$  of the unknown parameter vector  $\boldsymbol{\theta}$ :

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = \left( \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^T \right)^{-1} \left( \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i y_i \right). \quad (3.18)$$

Thus, if we substitute the sample points  $k$  (3.14) into the dynamic equations (3.5), (3.10) - (3.13), the vector of joint torques  $\boldsymbol{\tau}$  would linearly regress on the unknown parameters.

$$\boldsymbol{\tau} = \mathbf{X}\boldsymbol{\theta}, \quad (3.19)$$

where

$$\mathbf{X}^T = \begin{bmatrix} \ddot{q}_2 \cos(q_1 - q_2) + \sin(q_1 - q_2) \cdot \dot{q}_2^2 & \ddot{q}_2 \cos(q_1 - q_2) + \sin(q_1 - q_2) \cdot \dot{q}_1^2 \\ \ddot{q}_1 & 0 \\ \dot{q}_1 & 0 \\ \text{sign}(\dot{q}_1) & 0 \\ \text{sign}(\dot{q}_1 - \dot{q}_2) & 0 \\ 0 & \ddot{q}_2 \\ 0 & \dot{q}_2 \\ 0 & \text{sign}(\dot{q}_2) \\ 0 & \text{sign}(\dot{q}_1 - \dot{q}_2) \end{bmatrix}, \quad (3.20)$$

and

$$\boldsymbol{\theta} := \begin{bmatrix} m_2 l_1 l_{c2} + m_4 l_3 l_{c4} \\ m_2 l_{c2}^2 + m_3 l_{c3}^2 + m_4 l_3^2 + I_2 + I_3 \\ k_1 \\ b_{11} \\ b_{12} \\ m_1 l_{c1}^2 + m_2 l_1^2 + m_4 l_{c4}^2 + I_1 + I_4 \\ k_2 \\ b_{21} \\ b_{22} \end{bmatrix}. \quad (3.21)$$

Thus, after applying ordinary least squares method (3.18), the vector of unknown parameters was obtained as follows

$$\boldsymbol{\theta}^T = \begin{bmatrix} 0.042 & 0.109 & 0.409 & 0.185 & 0.123 & 0.084 & 0.388 & 0.230 & 0.071 \end{bmatrix}. \quad (3.22)$$

The resulting estimates of the robot's parameters can be used as an initial guess for the param-

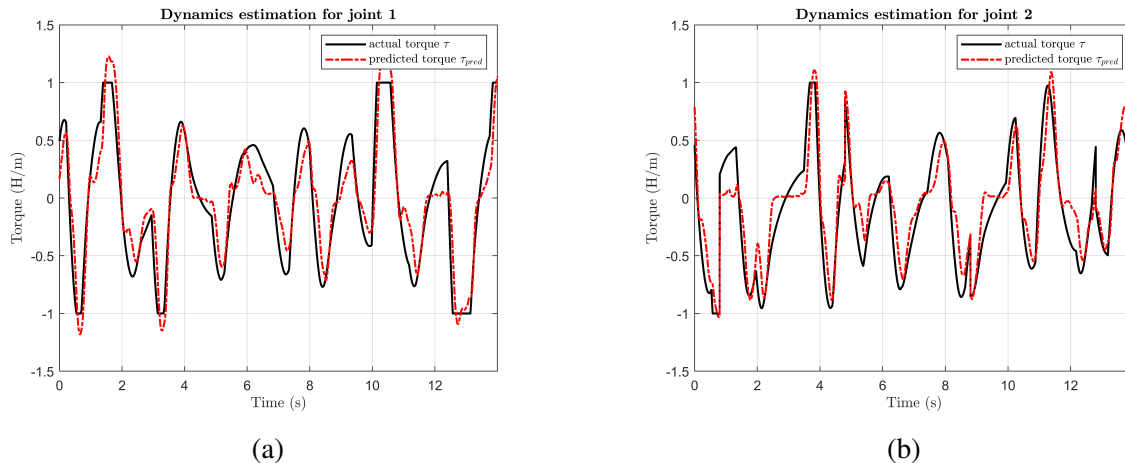


Figure 3.3: Dynamics estimation using linear regression: (a) - prediction of the torque for joint 1, (b) - prediction of torque for joint 2.

eter vector  $\hat{\Theta}_{init}$  used below in the trajectory control algorithm (3.28).

The result of fitting the linear regression model into the recorded data is shown in Figure 3.3. Thus, the linear regression method can be used to find the unknown parameters of the manipulator. The mean absolute error on the test set is less than 3% of the torque range.

### 3.2.5 Neural Network Based Dynamics Estimation

Analyzing the results of evaluation of the linear regression method, it can be noted that the robot's responses are not predicted perfectly. This suggests that the dynamic equations (3.5), (3.10) - (3.13), do not capture the dynamics in full and need some additional terms. Generally, it is quite hard to identify these additional terms because they may be highly nonlinear. The second approach estimates  $H(q)$ ,  $C(q, \dot{q})$  and  $F(q)$  using neural networks and does not require equations describing these matrices. This dynamics estimation method was inspired by [26]. In contrast with the linear regression approach, this technique estimates the functions rather than the constant parameters.

The neural network architecture that was used in experiments is presented in the Figure 3.4. This architecture allows functional expressions for matrices  $H(q)$ ,  $C(q, \dot{q})$  and  $F(q)$  to be found that further can be used in the control algorithm or dynamic cone estimation. The

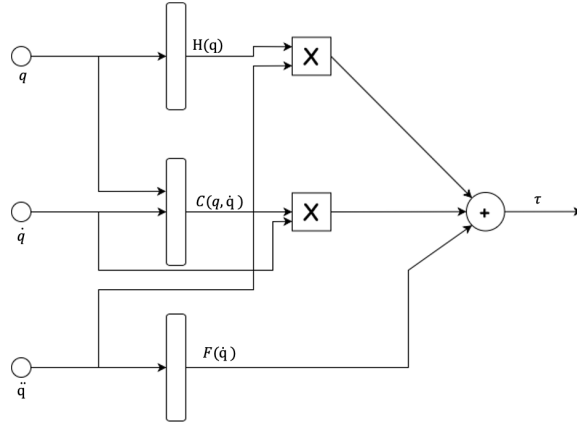


Figure 3.4: Architecture of a neural network for dynamics estimation.

inputs to the neural network in Figure 3.4 are the manipulator configurations, velocities and acceleration, the output is the joint torque. The part of neural network that estimates matrix function  $H(q)$  has the configuration  $q \in \mathbb{R}^2$  and matrix  $H(q) \in \mathbb{R}^{2 \times 2}$  on output. In the same manner, the  $C(q, \dot{q})$  part of neural network has vectors  $q \in \mathbb{R}^2$  and  $\dot{q} \in \mathbb{R}^2$  on input and  $C(q, \dot{q})$  on output. The same logic applies to  $F(\dot{q})$  part of the network. Further, the outputs of the above described subnetworks together with the inputs  $(q, \dot{q}, \ddot{q})$  are used to calculate the torque as in the Euler-Lagrange equation of dynamics 3.5.

The dataset has 20,000 sample points, where each point is comprised of input points  $(q, \dot{q}, \ddot{q}) \in \mathbb{R}^6$  and output points  $\tau \in \mathbb{R}^2$ . The dataset was randomly split in the ratio 80 : 10 : 10 to correspond to training, validation, and testing sets. The validation set is used to find hyperparameters of the network, such as the number of neurons, layers, activation function type, number of training epochs, batch size and type of loss function. The ranges of hyperparameters tuned are presented in the first row of Table 3.1. The hyperparameters were selected using the grid search approach [48]. The selected parameters are showed in Table 3.1 in bold. The lower and upper bounds of the hyperparameters were selected manually, based on the performance of the network. Thus, the network was trained on the selected hyperparameters. The training and validation losses are shown in Figure 3.5. The loss decreases rapidly within the first 20 epochs. After that the loss decreases slowly reaching a plateau after the 250th epoch.



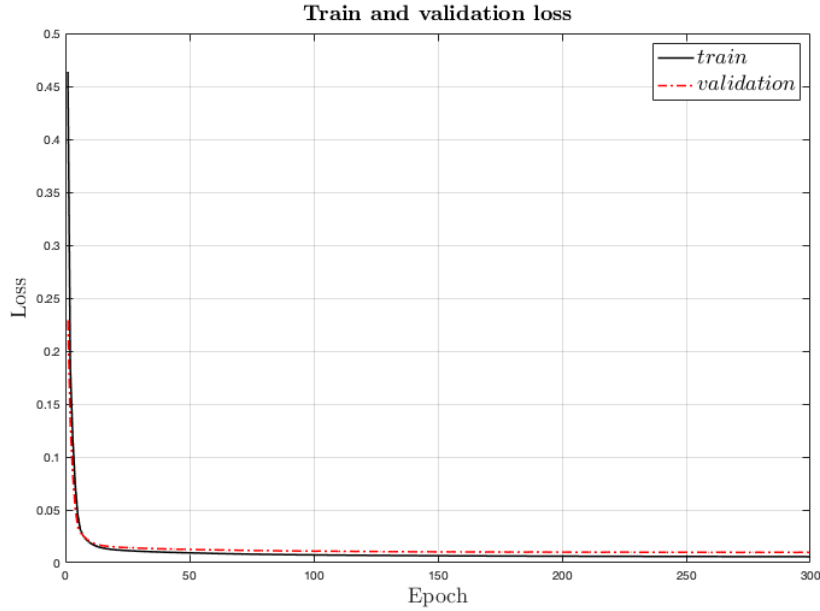


Figure 3.5: Train and validation loss of the neural network for dynamics estimation.

Next, after training the whole network, *i.e.*, finding a function that maps configuration and its derivatives to instantaneous joint torques  $\mathcal{D} : (q, \dot{q}, \ddot{q}) \rightarrow \tau$ , the network was split in three parts, that represent matrices  $H(q)$ ,  $C(q, \dot{q})$  and  $F(q)$  (see Figure 3.4). That is, the corresponding parts of the trained network were saved and used later as the separate neural networks. As with the linear regression method, the mean squared error on the testing set was also below 3% of the torque range.

Thus, both of the dynamic estimation approaches have similar mean absolute error on a test set. The linear regression approach is used in further developments and experiments, since its behaviour is more predictable and well defined compared to that of the neural network approach.

### 3.3 Environmental Dynamics and Control Algorithm

In this section, the mathematical model of the environment is presented, and adaptive control algorithm is described which is used in our experiments in conjunction with the scattering-

hyperparameter	number of neurons	layers	activation function type	number of epochs	batch size	loss function type
<b>range</b>	10, 50, <b>100</b> , 500, 1000	<b>1</b> , 2, 5, 10	<i>linear</i> , <i>ReLU</i> , <b><i>LeakyReLU</i></b> , <i>tanh</i> , <i>sigmoid</i>	50, 100, 200, <b>300</b> , 500	5,000, 10,000, 15,000, <b>20,000</b>	<b><i>MSE</i></b> , <b><i>MAE</i></b> , <b><i>CosineSimilarity</i></b> , <b><i>MSLE</i></b>

Table 3.1: The ranges of hyperparameters tuned. Values that minimize the error on validation set are in bold.

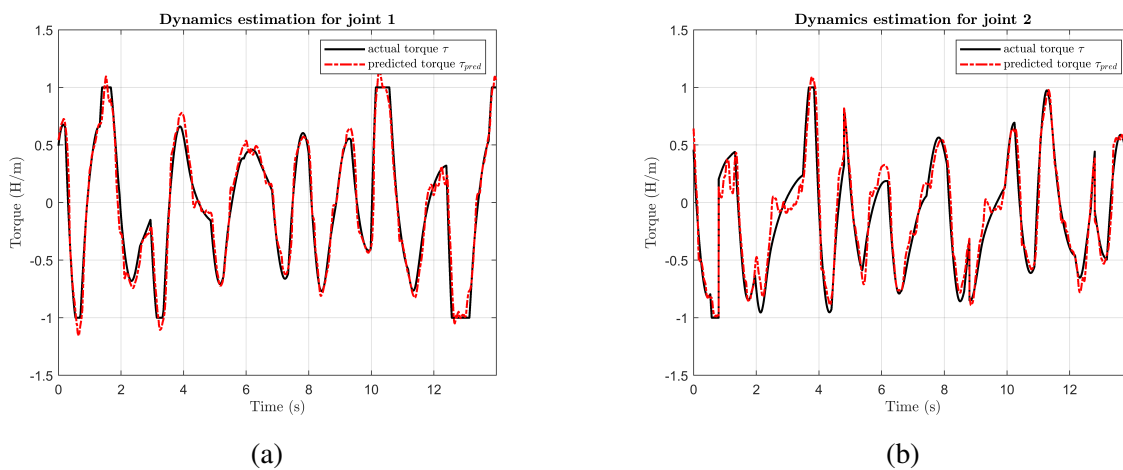


Figure 3.6: Dynamics estimation using neural network: (a) - prediction of torque for joint 1, (b) - prediction of torque for joint 2.

based stabilization method.

### 3.3.1 Environmental Dynamics

The dynamics of the environment can be described by the following equation

$$H_e \ddot{\mathbf{x}}_e + \frac{\partial P_e(\mathbf{x}_e)}{\partial \dot{\mathbf{x}}_e} + D_e \dot{\mathbf{x}}_e + f_e = 0, \quad (3.23)$$

where  $H_e \in R^{2 \times 2}$  and  $D_e \in R^{2 \times 2}$  are matrices representing inertia and damping characteristics of the environment, respectively, and  $P_e(\cdot)$  is the potential energy of the environment.

### 3.3.2 Adaptive Control Algorithm

One of the possible ways to enable trajectory tracking control of the robot is to employ the inverse dynamics model [75]. Typically such a model has the same structure as the actual dynamics model of manipulator. However, it is not practically feasible to find equations that would model the true physical system exactly. In particular, parametric uncertainty and unmodeled non-linearities usually present. The passivity- or Lyapunov-based adaptive control [74] allows for online update of the parameter estimates for the inverse dynamics control law:

$$\tau = \hat{H}(q)a + \hat{C}(q, \dot{q})v + \hat{F}(\dot{q}) - Kr \quad (3.24)$$

where  $\hat{H}(q)$ ,  $\hat{C}(q, \dot{q})$  and  $\hat{F}(\dot{q})$  are estimates of inertia, Coriolis/centrifugal, and friction matrices. The signals  $v$ ,  $a$ , and  $r$  are defined as follows:

$$v := \dot{q}^d - \Lambda \tilde{q} \quad (3.25)$$

$$a := \dot{v} = \ddot{q}^d - \Lambda \dot{\tilde{q}} \quad (3.26)$$

$$r := \dot{q} - v = \dot{\tilde{q}} + \Lambda \tilde{q} \quad (3.27)$$

where  $K = K^T > 0$  and  $\Lambda = \Lambda^T > 0$  are gain matrices, and  $\tilde{q} = q - q^d$  is the error between the desired and current configuration of manipulator. If the robot dynamics are parametrized using the linear regressor approach, the control law (3.24) becomes:

$$\tau = Y(q, \dot{q}, a, v)\hat{\theta} - Kr, \quad (3.28)$$

where  $Y(q, \dot{q}, a, v) \in \mathbb{R}^{m \times p}$  is the linear regressor,  $\hat{\theta} \in \mathbb{R}^p$  is a vector of the estimated robot parameters  $\theta$  given by (3.21). In the case of robot manipulator described in Section 3.2,  $m = 2$ ,  $p = 9$ , and regressor  $Y(q, \dot{q}, a, v) \in \mathbb{R}^{2 \times 9}$  can be obtained from (3.20) by matching the corresponding signals in (3.5) and (3.24). Specifically, the regressor  $Y(q, \dot{q}, a, v)$  has the following

form:

$$Y^T(q, \dot{q}, a, v) = \begin{bmatrix} a_2 \cos(q_1 - q_2) + \sin(q_1 - q_2) \cdot \dot{q}_2 \cdot v_2 & a_1 \cos(q_1 - q_2) + \sin(q_1 - q_2) \cdot \dot{q}_1^2 \cdot v_1 \\ a_1 & 0 \\ \dot{q}_1 & 0 \\ \text{sign}(\dot{q}_1) & 0 \\ \text{sign}(\dot{q}_1 - \dot{q}_2) & 0 \\ 0 & a_2 \\ 0 & \dot{q}_2 \\ 0 & \text{sign}(\dot{q}_2) \\ 0 & \text{sign}(\dot{q}_1 - \dot{q}_2) \end{bmatrix}, \quad (3.29)$$

The vector of the parameter estimates  $\hat{\theta} \in \mathbb{R}^9$  can be updated using different methods of adaptive control, such as least squares or gradient update laws. The latter one has the following form:

$$\dot{\hat{\theta}} = \Gamma^{-1} \cdot Y^T(q, \dot{q}, a, v) \cdot r, \quad (3.30)$$

where  $\Gamma = \Gamma^T > 0$  is a matrix (usually diagonal) of coefficients which control the update speed of the parameter vector  $\hat{\theta}$ . The lower the coefficients of which  $\Gamma$  comprises, the faster the update speed of the parameters of the robot. However, if the update speed is too fast, in practice this may result in controlled manipulator to become unstable. Thus, the parameters on the diagonal of matrix  $\Gamma$  were chosen to guarantee the fastest update speed while preserving the stability of the controlled robot.

The combination of the equations of the robot's dynamics (3.3), (3.4), and control laws (3.24)–(3.30), yields the closed loop system in the following form:

$$H(q)a + C(q, \dot{q})v + F(\dot{q}) + Kr - \tau_e = Y(q, \dot{q}, a, v)(\hat{\theta} - \theta) \quad (3.31)$$

### 3.4 Dissipativity Analysis of the Robot and the Environment

In order to design a scattering transformation that would stabilize robot-environment interaction, we need to find cones of each of the subsystems. That is, we need to find central subspaces  $\Omega$  and radii  $\phi_r$  of the robot and the environment cones. This can be done by analyzing the dissipativity properties of the respective subsystems. The closed-loop dynamics of the controlled manipulator can be found by substituting the control law (3.24) in the dynamics equation (3.3). As a result, we have the following

$$\dot{\tilde{\mathbf{x}}} = -\Lambda\tilde{\mathbf{x}} + r \quad (3.32)$$

$$\dot{r} = H_p^{-1}(q) [-C_p(q, \dot{q})r - Kr + f_e + f_r]. \quad (3.33)$$

Consider a storage function candidate for the manipulator

$$V_r = \frac{1}{2}r^T H_p(q)r + \frac{\rho}{2}\tilde{\mathbf{x}}^T \tilde{\mathbf{x}}, \quad (3.34)$$

where  $\rho > 0$  is a positive parameter. The time derivative of  $V_r$  along the trajectory is

$$\begin{aligned} \dot{V}_r &= -r^T Kr - r^T \left( \frac{1}{2}\dot{H}_p(q) - C_p(q, \dot{q}) \right) r + r^T \mathbf{v}_f \\ &\quad - \rho\tilde{\mathbf{x}}^T \Lambda\tilde{\mathbf{x}} + \rho\tilde{\mathbf{x}}^T r = -r^T Kr + r^T \mathbf{v}_f - \rho\tilde{\mathbf{x}}^T \Lambda\tilde{\mathbf{x}} + \rho\tilde{\mathbf{x}}^T r \\ &= \begin{bmatrix} \mathbf{v}_f \\ \tilde{\mathbf{x}} \\ r \end{bmatrix}^T \begin{bmatrix} \mathbb{O} & \mathbb{O} & \frac{1}{2}\mathbb{I} \\ \mathbb{O} & -\rho\Lambda & \frac{1}{2}\rho\mathbb{I} \\ \frac{1}{2}\mathbb{I} & \frac{1}{2}\rho\mathbb{I} & -K \end{bmatrix} \begin{bmatrix} \mathbf{v}_f \\ \tilde{\mathbf{x}} \\ r \end{bmatrix} \end{aligned} \quad (3.35)$$

where  $\mathbf{v}_f := f_e + f_r$  and  $\tilde{\mathbf{x}} := \mathbf{x} - \mathbf{x}_d$ . This implies that the manipulator (3.3) controlled with the control law (3.24) is  $(Q, S, R)$ -dissipative with respect to input  $\mathbf{v}_f$  and output  $(\tilde{\mathbf{x}}^T, r^T)^T$ . To simplify further analysis, let us transform the output vector to the form  $(\tilde{\mathbf{x}}^T, \hat{\mathbf{x}}^T)^T$ . This can be

done using the following transformation

$$\begin{bmatrix} \tilde{\mathbf{x}} \\ \dot{\tilde{\mathbf{x}}} \end{bmatrix} = \mathbf{T}_\lambda^{-1} \begin{bmatrix} \tilde{\mathbf{x}} \\ r \end{bmatrix}, \text{ where } \mathbf{T}_\lambda = \begin{bmatrix} \mathbb{I} & \mathbb{O} \\ \Lambda & \mathbb{I} \end{bmatrix} \in \mathbb{R}^{2m \times 2m}. \quad (3.36)$$

Substituting transformation (3.36) in equation (3.35) yields

$$W_r = \begin{bmatrix} \mathbb{I} & \mathbb{O} \\ \mathbb{O} & T_\Lambda^T \end{bmatrix} \begin{bmatrix} \mathbb{O} & \mathbb{O} & \frac{1}{2}\mathbb{I} \\ \mathbb{O} & -\rho\Lambda & \frac{1}{2}\rho\mathbb{I} \\ \frac{1}{2}\mathbb{I} & \frac{1}{2}\rho\mathbb{I} & -K \end{bmatrix} \begin{bmatrix} \mathbb{I} & \mathbb{O} \\ \mathbb{O} & T_\Lambda \end{bmatrix} \quad (3.37)$$

where  $W_r$  is the  $[QSR]$ -matrix of the robot (3.3), (3.24).

Next, let us analyse dissipativity of environment (3.31). The storage function candidate can have the following form

$$V_e = \frac{1}{2} \dot{\mathbf{x}}_e^T H_e \dot{\mathbf{x}}_e + P(\mathbf{x}_e) \quad (3.38)$$

The time derivative of  $V_e$  along the trajectory of the environment when the robot and the environment are in contact is defined as

$$\dot{V}_e = \dot{\mathbf{x}}_e^T D_e \dot{\mathbf{x}}_e - \dot{\mathbf{x}}_e^T f_e$$

It can be bounded from above using the following inequality

$$\dot{V}_e \leq \begin{bmatrix} f_e \\ \mathbf{x}_e \\ \dot{\mathbf{x}}_e \end{bmatrix}^T \mathbf{W}_e \begin{bmatrix} f_e \\ \mathbf{x}_e \\ \dot{\mathbf{x}}_e \end{bmatrix} = \quad (3.39)$$

$$\begin{bmatrix} f_e \\ \mathbf{x}_e \\ \dot{\mathbf{x}}_e \end{bmatrix} \begin{bmatrix} \mathbb{O} & \mathbb{O} & -\frac{1}{2}\mathbb{I} \\ \mathbb{O} & \mathbb{O} & \mathbb{O} \\ -\frac{1}{2}\mathbb{I} & \mathbb{O} & \mathbb{O} \end{bmatrix} \begin{bmatrix} f_e \\ \mathbf{x}_e \\ \dot{\mathbf{x}}_e \end{bmatrix} \quad (3.40)$$

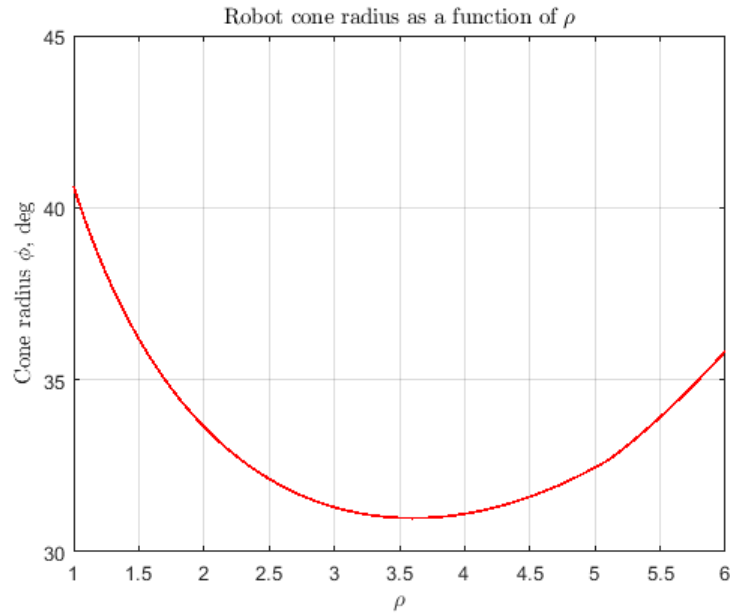


Figure 3.7: Radius of controlled manipulator's cone as a function of  $\rho > 0$ .

where  $\mathbf{W}_e$  is the matrix of quadratic supply rate of the environment.

### 3.5 Analysis of the Subsystems' Cones

The next step is to find central subspaces  $\Omega$  and the radii  $\phi_r$  of the cones defining the controlled manipulator and the environment. The values of  $\Omega$  and  $\phi_r$  can be found using the method described in Lemma 2.2.1. That is, to find the parameters of subsystems' cones only the supply rate matrices of the robot  $W_r$  and the environment  $W_e$  are used. In other words, it is not necessary to have the full knowledge of the dynamics of the interconnected subsystems. This is especially useful in the case of the environment, because it might not always be feasible to obtain equations that would precisely describe its dynamics. As can be seen from equation (3.37), the quadratic supply rate matrix of the robot depends on matrices  $\Lambda$  and  $K$  of the control law (3.24). These matrices can be chosen so that the robot has a desired trajectory tracking performance in free space. In the experiments conducted for this work, the matrices

$\Lambda$  and  $K$  were chosen as follows:

$$\Lambda = \begin{bmatrix} 1.05 & 0 \\ 0 & 2.25 \end{bmatrix} \frac{1}{s}, K = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.55 \end{bmatrix} \frac{s \cdot H}{m^2} \quad (3.41)$$

In addition to the matrices  $\Lambda$  and  $K$ , there is also a parameter  $\rho > 0$  in (3.37) that needs to be defined. The value of the coefficient  $\rho$  is chosen to minimize the radius of the robot's cone. The relation between  $\phi_r$  and  $\rho$  is shown on Figure 3.7. Thus,  $\rho = 3.6$  is the point of the minimum of the function that corresponds to  $\phi_r \approx 30.98^\circ$ . The center of the robot's cone forms a 2D subspace, which is defined as

$$\Omega_r = \text{span} \left\{ \begin{bmatrix} -0.0000 \\ 0.9303 \\ -0.0000 \\ 0.1984 \\ 0.0000 \\ 0.3085 \end{bmatrix}, \begin{bmatrix} -0.9416 \\ 0.0000 \\ -0.1714 \\ 0.0000 \\ -0.2898 \\ -0.0000 \end{bmatrix} \right\}. \quad (3.42)$$

Next, let us find the parameters of the environmental cone. The set of non-negative eigenvalues  $\lambda^+$  of  $[QSR]$  matrix of the environment is

$$\lambda^+(W_e) = \left\{ 0, 0, \frac{1}{2}, \frac{1}{2} \right\} \quad (3.43)$$

and the set of negative eigenvalues has the following form:

$$\lambda^-(W_e) = \left\{ -\frac{1}{2}, -\frac{1}{2} \right\}. \quad (3.44)$$



The parameters  $\mu^+$  and  $\mu^-$  are defined as follows:

$$\mu^- = \min \{ |\mu_i| : \mu_i \in \lambda^-(W_e) \} = \frac{1}{2}, \quad (3.45)$$

$$\mu^+ = \max \{ |\mu_i| : \mu_i \in \lambda^+(W_e) \} = \frac{1}{2}. \quad (3.46)$$

Therefore, according to Lemma 2.2.1, the radius of the environmental cone is defined as

$$\phi_e = \tan^{-1} \left( \sqrt{\frac{\mu^+}{\mu^-}} \right) \approx 0.785 \quad (3.47)$$

The center of the environmental cone is a 4D subspace, spanned by the following vectors

$$\Omega_e = \text{span} \left\{ \begin{array}{c} \begin{bmatrix} 0 \\ 0.7071 \\ 0 \\ 0 \\ 0 \\ -0.7071 \end{bmatrix}, \begin{bmatrix} 0.7071 \\ 0 \\ 0 \\ 0 \\ -0.7071 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \end{array} \right\}, \quad (3.48)$$

Once the parameters of the subsystems' cones are known, we should check if the graph separation stability condition is satisfied. First, the projection matrices  $\Pi_{\Omega_r}$  and  $\Pi_{\Omega_e}$  can be found from the known subspaces  $\Omega_r$  and  $\Omega_e$  in the following way. First, let  $\hat{\Omega}_r$  be a matrix whose columns span the central subspace  $\Omega_r$ . Then the robot's projection matrix  $\Pi_{\Omega_r} = \hat{\Omega}_r \hat{\Omega}_r^T$ . The projection matrix of the environment can be found in similar way. As calculations show,  $\sigma_{max} = \left( \Pi_{\hat{\Omega}_r}, \Pi_{\Omega_e} \right) \approx 0.372$  and  $\cos(\phi_r + \phi_e) \approx 0.243$ . This does not satisfy the graph separation stability condition (2.33), *i.e.*,  $\sigma \left( \Pi_{\hat{\Omega}_r}, \Pi_{\Omega_e} \right) \not\leq \cos(\phi_r + \phi_e)$ . This theoretical observation is in accordance with the experimental results where the robot-environment system experiences instability. However, the system can be stabilized with the scattering transformation techniques described in the following section.

### 3.6 Design of the Scattering Transformation

The robot-environment system (3.3), (3.24), (3.23) can be stabilized by using scattering techniques. In this section, we describe how to apply the scattering transformation described earlier in this work to the specific interconnected system of the controlled manipulator and environment. That is, the developed scattering transformation should stabilize the coupled system while preserving the free space task trajectory of the robot. As before, the general idea is to transform the cone of one of the subsystems, which in our case will be the robot, so that the graph separation stability condition is satisfied. First, suppose that the end-effector of the robot follows a desired trajectory  $\Psi_d = [\mathbf{x}_d^T(t), \dot{\mathbf{x}}_d^T(t)]$ . Then the actual trajectory of the robot can be defined as  $\Psi = [\mathbf{x}^T(t), \dot{\mathbf{x}}^T(t)]$ , where  $\mathbf{x}$  and  $\mathbf{x}_d$  represent the actual and desired positions,  $\dot{\mathbf{x}}$  and  $\dot{\mathbf{x}}_d$  represent the actual and desired velocities. When the robot's end-effector is in contact with environment the force  $f_e$  is generated, thus forming a closed-loop robot-environment system. To guarantee the stability of the overall system, we can apply a scattering transformation on the robot subsystem in the following way. Suppose that the controlled manipulator has input  $f_e$ , and tracking error  $v = \Psi - \Psi_d$ . Then we can design an operator  $\mathbb{S}$  that would transform a signal in the following way

$$\begin{bmatrix} f_e \\ v \end{bmatrix} = \mathbb{S} \begin{bmatrix} \mathbf{v}_f \\ \mathcal{E} \end{bmatrix} \quad (3.49)$$

where  $\mathbf{v}_f$  is a new force input and  $\mathcal{E}$  is a new tracking error. Since there is a physical interaction with energy exchange between the robot and its environment, the scattering transformation (3.49) cannot be applied directly. Instead, we can implement the scattering transformation indirectly through the introduction of reference signals  $f_r = \mathbf{v}_f - f_e$  and  $\tilde{\Psi}_r = v - \mathcal{E}$  as shown on Figure 3.8. Next, let us put constraints on the scattering transformation  $\mathbb{S}$  so that the trajectory tracking performance of the robot remains the same when there is no contact with the environment. That is, in this case the signal going through the scattering transformation block should

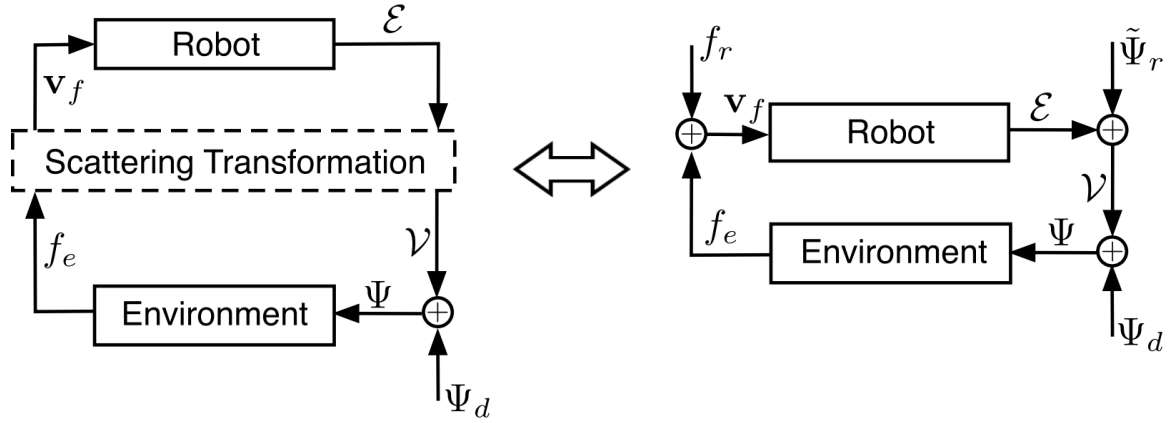


Figure 3.8: Scattering-based stabilization of robot-environment interaction.

remain unchanged. Thus, when there is no external force  $f_e$ , the transformed force  $\mathbf{v}_f$  should be equal to zero, and the new tracking error  $\mathcal{E}$  should be equal to the actual tracking error  $v$ . Thus, the inverse of the scattering operator should have the following form

$$\mathbb{S}^{-1} = \begin{bmatrix} \mathbb{S}_1 & \mathbb{O} \\ \mathbb{S}_2 & \mathbb{I} \end{bmatrix} \quad (3.50)$$

where  $\mathbb{S}_1, \mathbb{S}_2$  are arbitrary and  $\mathbb{S}_1$  is nonsingular. To get the original scattering transformation (3.49), we can use the inverse operator (3.50) and have the following:

$$\mathbb{S} = \begin{bmatrix} \mathbb{S}_1^{-1} & \mathbb{O} \\ -\mathbb{S}_2\mathbb{S}_1^{-1} & \mathbb{I} \end{bmatrix}. \quad (3.51)$$

Thus, the scattering transformation (3.49), (3.51) guarantees that the trajectory performance of the robot in free space remains unchanged. However, since signals  $f_e$  and  $v$  are readily available, it is easier to use the inverse scattering operator as follows

$$\begin{bmatrix} \mathbf{v}_f \\ \mathcal{E} \end{bmatrix} = \mathbb{S}^{-1} \begin{bmatrix} f_e \\ v \end{bmatrix}. \quad (3.52)$$

Next, we need to find matrices  $\mathbb{S}_1$  and  $\mathbb{S}_2$  that would guarantee the stability of the coupled system. First, consider signals  $\mathcal{E}$  and  $\tilde{\Psi}_r$  in Figure 3.8 that are defined as follows

$$\mathcal{E} = \begin{bmatrix} \mathbf{x} - \mathbf{x}_r \\ \dot{\mathbf{x}} - \dot{\mathbf{x}}_r \end{bmatrix}, \tilde{\Psi}_r = \begin{bmatrix} \mathbf{x}_r - \mathbf{x}_d \\ \dot{\mathbf{x}}_r - \dot{\mathbf{x}}_d \end{bmatrix} \quad (3.53)$$

Therefore, considering the transformation (3.52) scattering should establish a relationship between the following signals

$$\begin{bmatrix} f_e + f_r \\ \mathbf{x} - \mathbf{x}_r \\ \dot{\mathbf{x}} - \dot{\mathbf{x}}_r \end{bmatrix} = \mathbb{S}^{-1} \begin{bmatrix} f_e \\ \mathbf{x} - \mathbf{x}_d \\ \dot{\mathbf{x}} - \dot{\mathbf{x}}_d \end{bmatrix}. \quad (3.54)$$

Equation (3.54) is equivalent to the following transformation

$$\begin{bmatrix} f_r \\ \mathbf{x}_d - \mathbf{x}_r \\ \dot{\mathbf{x}}_d - \dot{\mathbf{x}}_r \end{bmatrix} = [\mathbb{S}^{-1} - \mathbb{I}] \begin{bmatrix} f_e \\ \mathbf{x} - \mathbf{x}_d \\ \dot{\mathbf{x}} - \dot{\mathbf{x}}_d \end{bmatrix}. \quad (3.55)$$

Thus, considering the structure of the operator  $\mathbb{S}^{-1}$  given by (3.50), the scattering transformation should have the following form

$$\mathbb{S}^{-1} = \begin{bmatrix} \mathbb{S}_1 & \mathbb{O} & \mathbb{O} \\ \mathbb{S}_{21} & \mathbb{I} & \mathbb{O} \\ \mathbb{S}_{22} & \mathbb{O} & \mathbb{I} \end{bmatrix} \in \mathbb{R}^{6 \times 6} \quad (3.56)$$

Substituting the scattering transformation (3.56) in (3.55), the reference signals can be derived as follows

$$f_r = [\mathbb{S}_1 - \mathbb{I}_3] f_e, \quad (3.57)$$

$$\mathbf{x}_r = \mathbf{x}_d - \mathbb{S}_{21}f_e, \quad (3.58)$$

$$\dot{\mathbf{x}}_r = \dot{\mathbf{x}}_d - \mathbb{S}_{22}f_e, \quad (3.59)$$

The scattering transformation (3.56) is a function of a parameter vector  $a = \begin{bmatrix} a_1 \dots a_6 \end{bmatrix}^T \in \mathbb{R}^6$ , *i.e.*,  $\mathbb{S} = \mathbb{S}(a)$ . The functional that is minimized has the following form:

$$F_\Delta = [a - a_0]^T \cdot \Delta \cdot [a - a_0], \quad (3.60)$$

where  $\mathbf{a}_0 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T \in \mathbb{R}^6$ , and  $\Delta$  is a diagonal matrix with  $\delta_i > 0, i = 1, \dots, 6$ , such that  $\text{tr}\Delta = \sum \delta_i = 1$ . In other words, a scalar  $F_\Delta$  can be calculated in the following way:

$$F_\Delta = \text{tr}(W_1(\mathbb{S}_1 - \mathbb{I}_3)^2 + W_2S_{21}^2 + W_3S_{22}^2) \quad (3.61)$$

where the matrices  $W_i$  are the weight matrices. In the experiments  $W_i = \mathbb{I}/6$  for  $i = 1, \dots, 3$ . Therefore, in order to find matrix  $\mathbb{S}^{-1}$  the point of minimum of function (3.61) needs to be found. The point of minimum  $\mathbf{a}$  can be used to comprise the matrix  $\mathbb{S}^{-1}$ . In other words:

$$\mathbf{a}^* = \arg \min_{\mathbf{a} \in \mathbb{R}^6} F_\Delta(\mathbf{a}) \quad (3.62)$$

This optimization problem is subject to the constraints that are defined in the remaining part of this section.

The matrix  $W$  of the robot has the following form:

$$W = G_{rob} \begin{bmatrix} s^2 \mathbb{I}_2 & \mathbb{O} \\ \mathbb{O} & -c^2 \mathbb{I}_4 \end{bmatrix} G_{rob}^T \quad (3.63)$$

where  $G_{rob}$  is the matrix comprised of the basis vectors of the robot's cone,  $s = \sin \phi_{rob}$  and

$$c = \cos \phi_{rob}.$$

The desired  $[QSR]_d$  matrix has the form:

$$[QSR]_d = \mathbb{S}_0^T \cdot W \cdot \mathbb{S}_0 \quad (3.64)$$

The matrix  $M$  and scalar  $\sigma$  are defined in the following way:

$$M = P_d \cdot P_e \quad (3.65)$$

$$\sigma = \sqrt{\max v} \quad (3.66)$$

where  $P_d$  is the desired projector, *i.e.*, projector of  $[QSR]_d$ ,  $P_e$  is the projector of the environment, vector  $v$  is the vector of eigenvalues of matrix  $M \cdot M^T$ . Thus, the constraint on the optimization problem (3.62) is defined as:

$$C(\mathbf{a}) = \begin{bmatrix} \sigma - \cos(\phi_e + \phi_d + \delta_d) \\ -(\mathbf{a}_c^T)^2 + 10^{-3} \end{bmatrix} < 0 \quad (3.67)$$

where  $\mathbf{a}_c = [a_1, a_2]$ . Therefore, by solving equation (3.62) with the constraints (3.67), the matrix  $\mathbb{S}^{-1}$  can be found.

Finally, let us demonstrate the actual values of the scattering transformation found using the above described pipeline. The desired gap  $\delta_d$  was chosen to be equal to  $4^\circ$  in the experiments. Thus, matrix  $\mathbb{S}^{-1}$  has the following form:

$$\mathbb{S}^{-1} = \begin{bmatrix} \mathbb{S}_1 & \mathbb{O} & \mathbb{O} \\ \mathbb{S}_{21} & \mathbb{I} & \mathbb{O} \\ \mathbb{S}_{22} & \mathbb{S}_3 & \mathbb{O} \end{bmatrix} \quad (3.68)$$

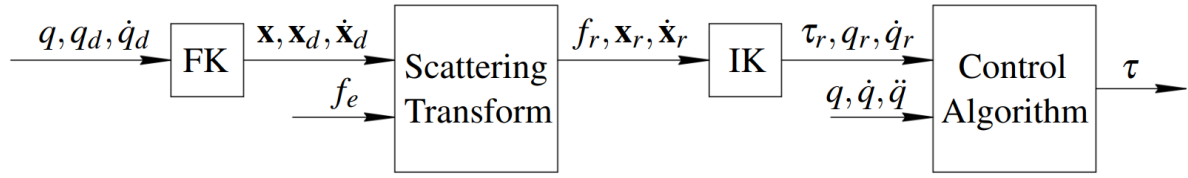


Figure 3.9: Control architecture for robot-environment interaction stabilization through scattering transformation

where

$$\begin{aligned} \mathbb{S}_1 &\approx \begin{bmatrix} 0.846 & 0 \\ 0 & 0.844 \end{bmatrix}, \mathbb{S}_{21} \approx \begin{bmatrix} 0.082 & 0 \\ 0 & 0.070 \end{bmatrix}, \\ \mathbb{S}_{22} &\approx \begin{bmatrix} -0.349 & 0 \\ 0 & -0.344 \end{bmatrix} \end{aligned} \quad (3.69)$$

### 3.7 Complete Control Architecture of the System

Thus  $f_r$ ,  $\mathbf{x}_r$  and  $\dot{\mathbf{x}}_r$  serve as the new end-effector force, desired position and velocity for the passivity-based adaptive control algorithm. Since the scattering transforms signal in task space, we need to convert joint positions and forces to and from task space. The joint variables are converted to task space using forward kinematic rules. The velocity of the end-effector is determined as

$$\mathbf{x}_d = J(q_d)\dot{q}_d \quad (3.70)$$

where  $J(q_d)$  is the manipulator's Jacobian matrix,  $\mathbf{x}_d$  is the desired end-effector position. The robot's Jacobian is defined as:

$$J(q) = \begin{bmatrix} -l_2 \sin q_1 & -l_1 \sin q_2 \\ l_2 \cos q_1 & l_1 \cos q_2 \end{bmatrix}. \quad (3.71)$$

The readings from the force sensor attached to the end-effector are transformed to the base frame using the following equation:

$$f_e = Rf_{ee} = \begin{bmatrix} \cos(q_1 + q_2) & -\sin(q_1 + q_2) \\ \sin(q_1 + q_2) & \sin(q_1 + q_2) \end{bmatrix} f_{ee} \quad (3.72)$$

where  $R$  is the rotation frame that represents the rotation of the end-effector frame with respect to the inertial frame. Thus, having calculated the environment force  $f_e$ , the desired position  $\mathbf{x}_d$  and the velocity  $\dot{\mathbf{x}}_d$ , using equations (3.70)–(3.72), we apply the scattering calculation (3.57)–(3.59). Further, we need to transform the reference signals from task space back to joint space. This can be done as follows:

$$\tau_r = J^T(q) \cdot f_r \quad (3.73)$$

$$\dot{q}_r = J^T(q) \cdot \dot{\mathbf{x}}_r \quad (3.74)$$

The reference joint position  $q_r$  is determined using the inverse kinematics of the robot. The reference torque  $\tau_r$  is used as an additional term in the control law (3.28):

$$\tau = Y(q, \dot{q}, a, v) \hat{\theta} - Kr - \tau_r, \quad (3.75)$$

In this way, the reference signal calculated using the scattering transformation is fed to the control algorithm (3.28). The complete control architecture with the forward and inverse kinematics blocks, the scattering transformation and the passivity-based adaptive control algorithm is presented on Figure 3.9.

## 3.8 Conclusions

This chapter described all the necessary steps for implementation of the scattering-based stabilization algorithm on a real physical system. The chapter started with the description of the



hardware used in the experiments. Next, the dynamic equations of the robot were derived. The two data-driven approaches were employed to estimate the unknown robot parameters and counter unmodeled dynamics. Both dynamics estimation methods, linear regression and neural network-based estimation showed similar performance. Thus, parameter estimation method based on Linear regression was selected to be used in further experiments since it is more predictable and well-defined compared to the neural network-based approach. The resulting dynamics equations of the robot are necessary for the dynamic cone estimation and implementation of the trajectory control algorithm. Next, the dynamics of the environment are described, which is also necessary for the estimation of the dynamic cone of the environment and further design of the scattering transformation. By analyzing the dissipativity properties and dynamic cones alignment of the robot and the environment systems, it was found that the closed-loop system is unstable, which is in a complete accordance with the experimental observations. Further, the scattering transformation design for stability of robot-environment interaction was presented.

# Chapter 4

## Experimental Results

This chapter presents the implementation of the scattering transformation on a real physical robot and the corresponding experimental results. The chapter starts with the description of how velocities and accelerations are estimated based on the position measurements. In Section 4.2, the reference trajectories used in the experiments are described. Experimental results are reported in Section 4.3. Furthermore, in order to gain better understanding of the behavior of a coupled system, the models of the robot and the environment used in the experiments are implemented in simulations. Section 4.4 describes the simulation environment and the steps needed to bring the behaviour of a simulated system closer to that of a real one. Finally, Section 4.5 contains concluding remarks.

### 4.1 Velocity and Acceleration Estimation

The robot used in experiments does not provide joint velocity and acceleration measurements, these signals have to be reconstructed by taking the derivatives from the joint position. A well-known problem with the signal differentiation is its sensitivity to high frequency measurement noises. Practical differentiation is a trade-off between exact differentiation and robustness with respect to noise since it is typically impossible to discern between the noise and the actual signal with any degree of reliability.

Differentiators are commonly based on the assumption that the measurement noise has low magnitude and high frequency. The goal is to filter out this noise [42]. Thus, due to the filtering involved, conventional high-frequency [6] and sliding-mode [24, 88] differentiators do not provide exact derivatives. The exact derivatives can be obtained using the robust exact finite-time-convergent differentiator [49], provided that the highest order derivative is bounded by a known constant. This differentiator has the best possible asymptotics in the presence of infinitesimal Lebesgue-measurable sampling noises. It has found its application in numerous fields [8, 19]. The following is the design of the high-order sliding mode observer [49] for estimation of the first and the second derivatives of the input signals.

Let the input signal  $\phi(t)$  be a function defined on  $[0, +\infty)$ , and consisting of a bounded Lebesgue-measurable noise with unknown characteristics and unknown basic signal. Let us assume that the  $k$ -th derivative of the basic signal has known Lipschitz constant  $L > 0$ . Then the real-time robust estimations of the derivatives  $\phi_0^{(i)}(t)$ ,  $i = 0, \dots, k$ , for  $k = 4$ , which coincide with the exact derivatives in the absence of noise, are computed as follows [49]:

$$\begin{bmatrix} \dot{z}_0 \\ \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} -\lambda_3 \cdot L^{1/4} \cdot |z_0 - \phi|^{3/4} \cdot \text{sign}\{z_0 - \phi\} + z_1 \\ -\lambda_2 \cdot L^{1/3} \cdot |z_1 - v_0|^{2/3} \cdot \text{sign}\{z_1 - v_0\} + z_2 \\ -\lambda_1 \cdot L^{1/2} \cdot |z_2 - v_1|^{1/2} \cdot \text{sign}\{z_2 - v_1\} + z_3 \\ -\lambda_0 \cdot L \cdot \text{sign}\{z_3 - v_2\} \end{bmatrix} \quad (4.1)$$

where  $\lambda_0 = 0.5, \lambda_1 = 0.5, \lambda_2 = 2, \lambda_3 = 3$ , and  $L > 0$  is a sufficiently large constant gain. Theoretically,  $L$  should be chosen such that

$$L \geq \max_t |\phi^{(4)}(t)|. \quad (4.2)$$

In practice,  $L > 0$  should be chosen sufficiently large. However, the experiments showed that the higher the gain  $L$ , the more noisy the estimation becomes. Thus,  $L$  should be chosen such that there is no noise in the estimation, while being bounded from below by inequality

(4.2). In our experiments  $L = 5$ , which satisfies inequality (4.2). The velocity and acceleration estimations are obtained from (4.1) as follows:

$$\begin{aligned}\hat{\phi} &\approx z_1, \\ \hat{\dot{\phi}} &\approx z_2.\end{aligned}\tag{4.3}$$

Despite the fact that we only need to estimate the first and second derivatives of the measured position, we still need our observer to be of fourth order, *i.e.*, internally it should also estimate the third derivative. The reason for this is because if our estimator is only of the third order, the differential equation that corresponds to estimation of acceleration is not continuous. Therefore, the solution of this equation would have cusp points, *i.e.*, infinite curvature. However, the additional, fourth order provides smooth estimations of acceleration.

The control scheme (3.24) used in the experiments only requires estimation of the first derivative of the robot's position, while for the desired trajectory both the first and second derivatives have to be estimated. The signals in both cases are estimated using the fourth order sliding mode observer described above.

## 4.2 Reference Trajectory

In this section, the reference (desired) trajectory used in the experiments below is described. The desired trajectory of the robot's end-effector in task space is shown in Figure 4.1. This trajectory consists of two parts. The first part of the trajectory represents a path which starts at the point  $p_s = (0, 0.5)m$ , and comprises two circles with radius  $r_{tr} = 0.16m$  and center at the point  $p_c = (0.16, 0.50)m$  in the counter-clockwise direction. The purpose of this part of the trajectory is to give the control algorithm time to update and stabilize parameters that enter the vector  $\hat{\theta}$ . The second part of the trajectory represents a straight line normal to the surface of the environment. The last point of the trajectory is  $0.02m$  inside the environment. The desired trajectory  $\mathbf{x}_d$  together with its first derivative  $\dot{\mathbf{x}}_d$  and second derivative  $\ddot{\mathbf{x}}_d$  are obtained via a

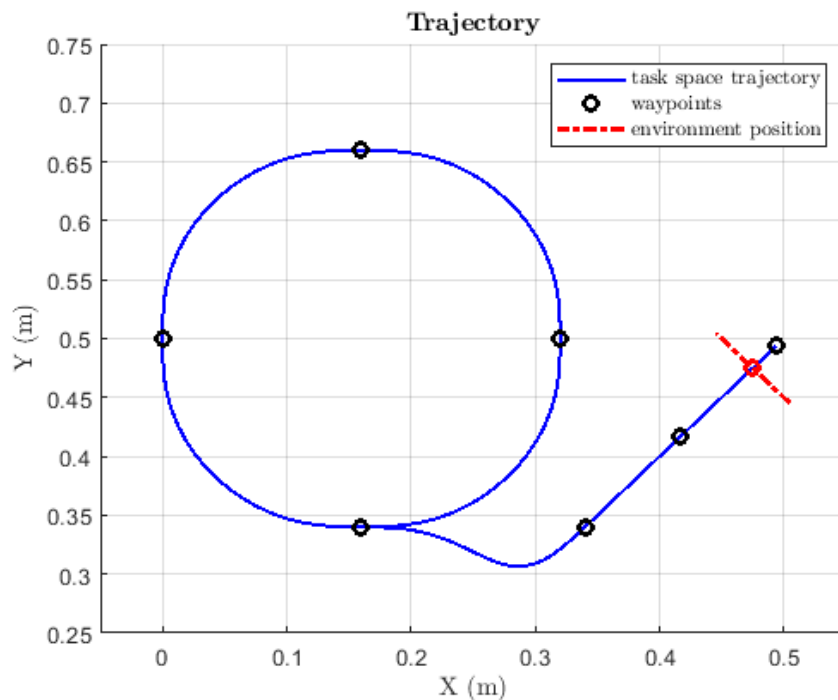


Figure 4.1: Desired task space trajectory of the robot (in blue) and environment position (in red).

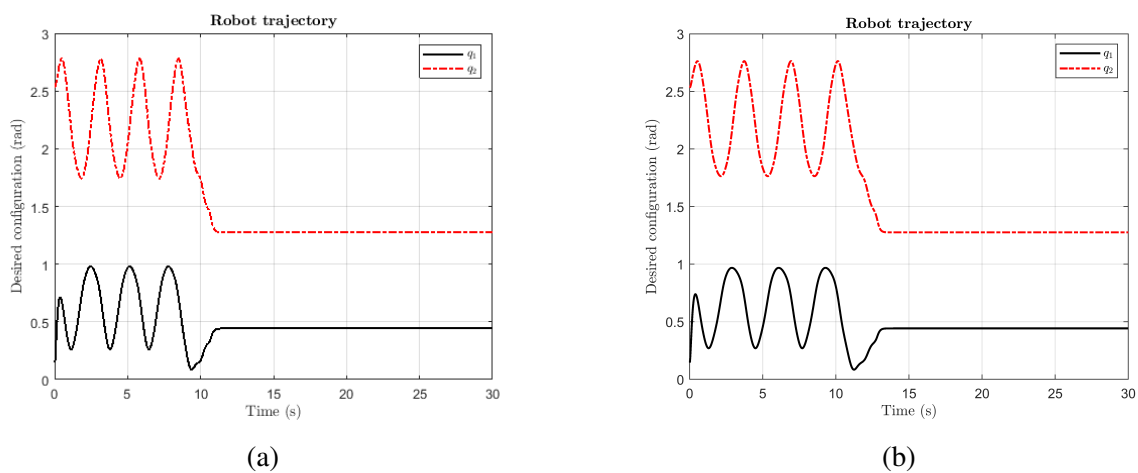


Figure 4.2: Desired joint space trajectory of the robot: (a) 10.7 s variant of trajectory, (b) 12.8 s variant of trajectory.

high-order sliding mode observer described above in Section 4.1.

Two versions of this trajectory were examined in this work, which differ in their desired velocity. The first version of the trajectory corresponds to the case where the above mentioned path that has to be completed in 10.7 s. The second version represents a trajectory that has to be completed in 12.8 s. Trajectories are designed in such a way that the robot's end-effector moves along the trajectory with the uniform speed. Plots of both versions of the desired trajectory in joint space are shown in Figure 4.2. The coefficients of trajectory control algorithm were chosen such that the robot has a desired trajectory performance in free space; for exact values see (3.41). At the point of collision, a piece of soft silicon with a width of  $\approx 20\text{mm}$  is rigidly attached to the table, as shown in Figure 3.1a. To execute the trajectory described above, both of the robot's motors have to be used. In addition, their contributions are different considering the structure of the robot. To reach the starting point of the trajectory from the home position, the PD control scheme was used.

### 4.3 Contact Stability Experiments

The passivity-based (Lyapunov-based) adaptive control algorithm (3.28), (3.30)) which is used in the contact stability experiments has a number of parameters that need to be tuned. These parameters include matrices  $K > 0$ ,  $L > 0$ , and  $\Gamma > 0$ . The matrices of the coefficients  $K > 0$ ,  $L > 0$  of the motion control algorithm were chosen such that the robot has a desired trajectory tracking performance in free space, the exact values of these matrices are given in (3.41). That is, if the coefficients in the matrices  $K > 0$  and  $L > 0$  are too large the real trajectory of the robot may heavily overshoot and the robot may be only marginally stable with respect to the desired trajectory, *i.e.*, it may have damped or even sustained oscillations.

On the other hand, if the coefficients  $K > 0$ ,  $L > 0$  are too small the robot may heavily lag with respect to the desired position trajectory and the corresponding tracking error may be unacceptably high. Thus, the coefficients in the matrices  $K > 0$  and  $L > 0$  are chosen to ensure

a trade-off between the robot's stability and the trajectory error. The coefficients on the main diagonal of the matrix  $\Gamma > 0$  (3.30) control the speed of update of the parameter vector  $\hat{\theta}$ . Higher coefficients correspond to higher speed of update. The matrix  $\Gamma > 0$  was chosen such that the parameter estimates reach steady-state by the time the robot finishes executing the first part of the trajectory.

Overall, the two versions of the desired trajectory described above in Section 4.2 were used in the experiments. In order to evaluate the adaptation properties of the control algorithm, for each version of the desired trajectory, three initial estimates of the parameter vector  $\hat{\theta}^{init}$  were tested. The first initial value of  $\hat{\theta}_1^{init}$  comprises the parameter values found using linear regression estimation (see Section 3.2.4). That is, the initial values of the parameters  $\hat{\theta}_1^{init}$  are close to the actual parameters of the robot. The second initial condition for  $\hat{\theta}_2^{init}$  is chosen such that it differs from the estimates obtained through the linear regression by  $\pm 30\%$ . Lastly, the third initial value of  $\hat{\theta}_3^{init}$  used in the experiments is set equal to zero vector, *i.e.*,  $\hat{\theta}_3^{init} = \mathbb{O}_{12}$ .

The results of experiments with and without the scattering transformation for the first version of the trajectory (*i.e.*, with execution time = 10.7 s) are presented below in Figures 4.3-4.8. The results of the experiments for the second version of trajectory (execution time = 12.8 s) are presented in Figures 4.9-4.14. As can be seen, without the scattering transformation the robot typically starts oscillating as soon as it collides with the environment. The general reason for this instability is described in Section 1.2.2. It consists in the fact that even though both subsystems are passive, the nature of the subsystems interconnection makes the overall system non-passive. The root cause for oscillations is explained in the following section.

As can be seen from the experimental results, the trajectory tracking performance improves with time due to updates of the parameter vector  $\hat{\theta}$ . Typically, after collision with the environment, some values of the parameter vector tend to update to compensate for the discrepancy between the actual and desired position of the robot, as seen in Figures 4.3f-4.14, subfigures (f). Implementation of the scattering transformation stabilizes the robot-environment interaction (see Figures 4.4, 4.6, 4.8, 4.10, 4.12, and 4.14); at the same time, the trajectory tracking

performance in free space remains unaffected. Further, it can be seen that the force measurement noise does not significantly impact the trajectory tracking performance, which is in complete accordance with the theoretical consideration that the gap between the robot and the environment's dynamic cones provides for robustness with respect to perturbations.

## 4.4 Simulation Results

As shown in the previous section, the scattering transformation can be used to stabilize the system. As shown in Section 1.2.2, the reason for the original instabilities occurring in the system is that the system is non-passive. However, what is the root cause of this non-passivity and oscillations? The aim of this section is to answer this question. One of the advantages of simulations is that they enable experiments to be conducted that otherwise would not be possible with a physical robot. A well implemented simulation can show how the robot would behave if there were no sensor noise or delay.

### 4.4.1 Description of the Simulation Environment

The dynamic model of the robot used in the simulation corresponds to the dynamics of the physical robot used in the experiments described above. The dynamics (3.5), (3.10)-(3.13) that were found by fitting a function to the robot's movement recordings are about 3% different from the behaviour of the real robot. The environment in the simulation was chosen to match the characteristics of the environment used in the real experiments. The piece of silicon that is used in the experiments is somewhat similar to a very stiff spring that has a substantial amount of damping. The scheme of the environment is presented on Figure 4.15.

When the end-effector is in contact with the environment, it experiences a force defined as follows:

$$f = -(D \cdot v + K \cdot p_\delta) \quad (4.4)$$



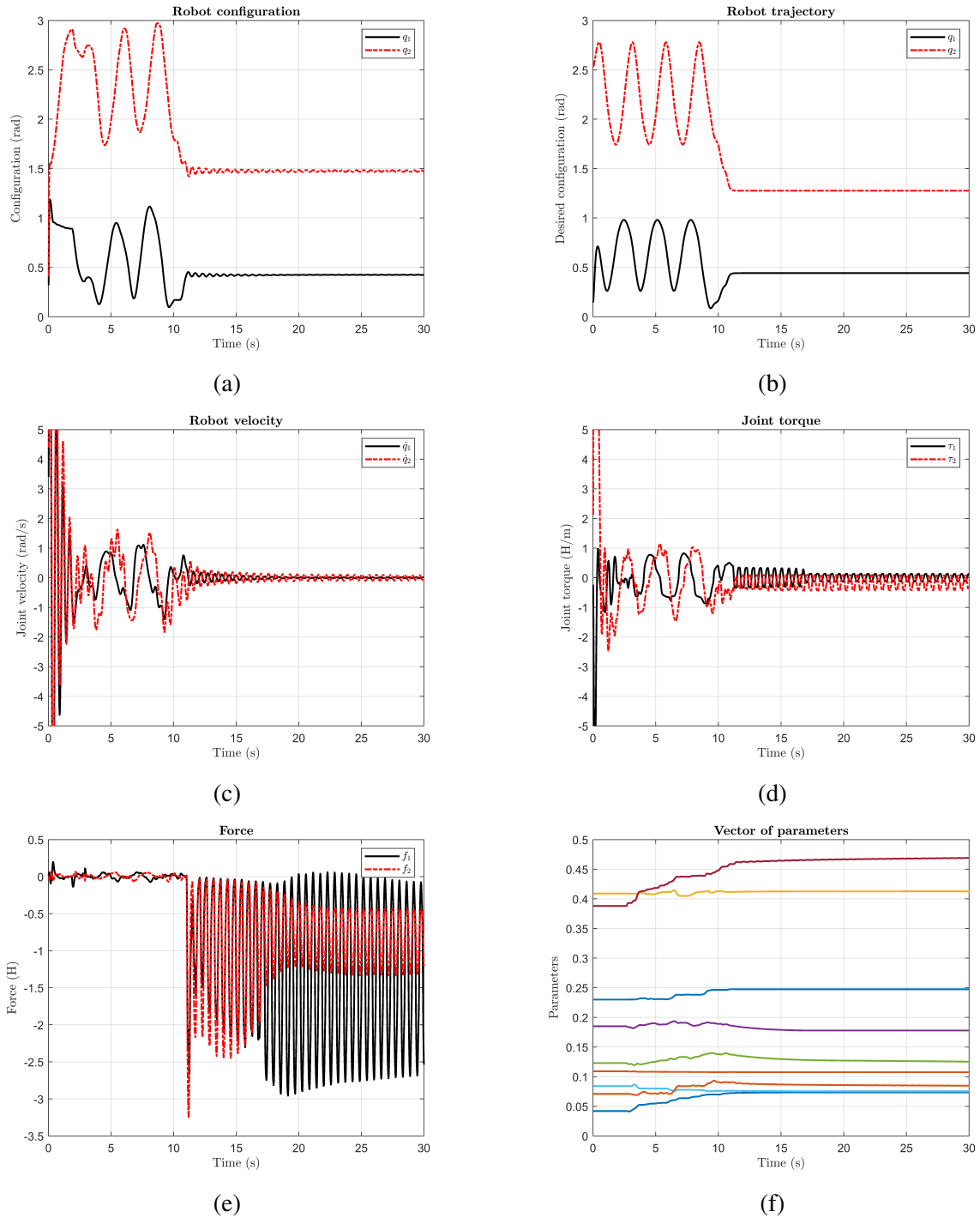


Figure 4.3: Experimental results: no scattering transformation, trajectory execution time 10.7 s, initial parameters  $\hat{\theta}(0) = \hat{\theta}_1^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates  $\hat{\theta}$ .

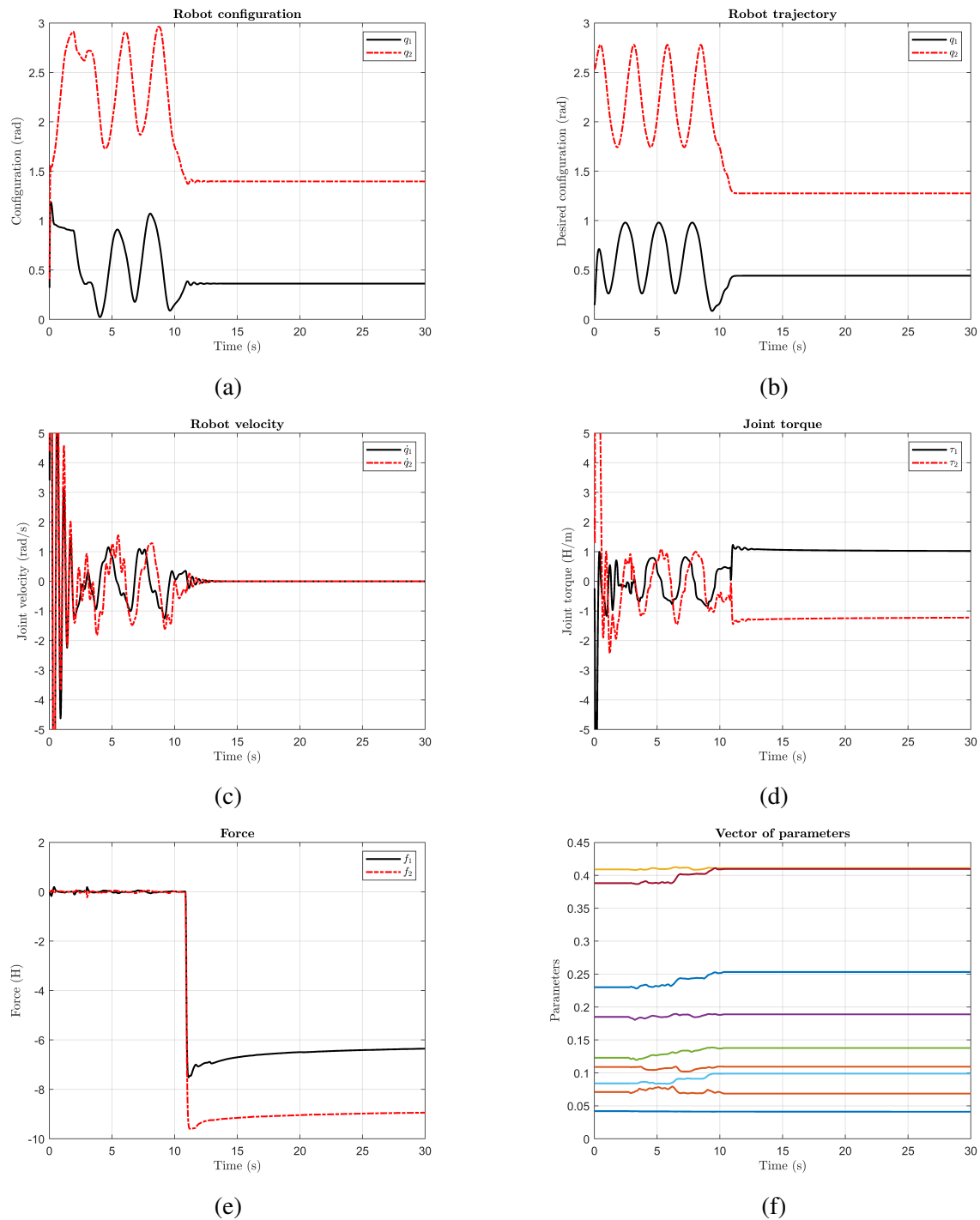


Figure 4.4: Experimental results: the scattering transformation is applied, trajectory execution time 10.7 s, initial parameters  $\hat{\theta}(0) = \hat{\theta}_1^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates  $\hat{\theta}$ .

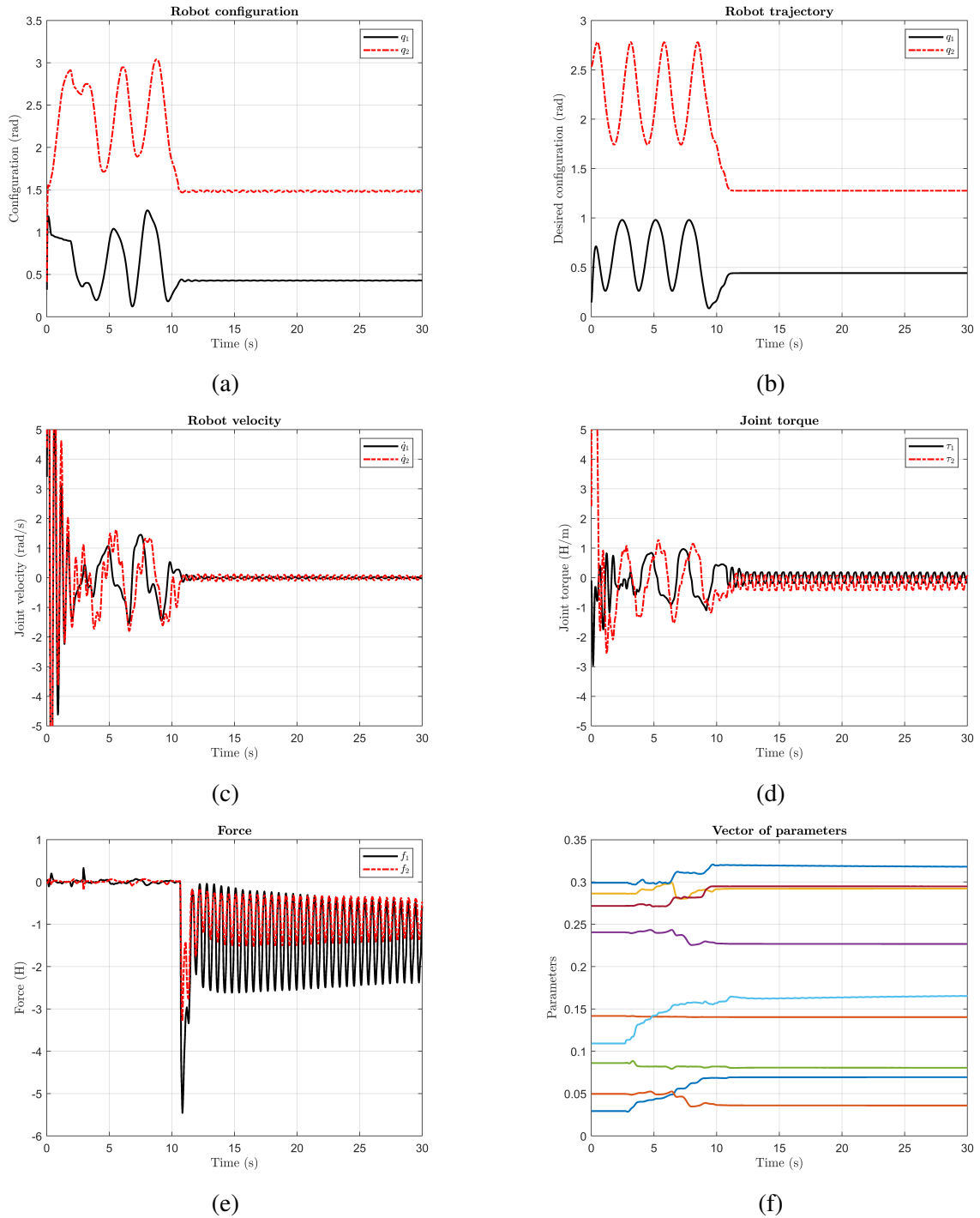


Figure 4.5: Experimental results: no scattering transformation, trajectory execution time 10.7 s, initial parameters  $\hat{\theta}(0) = \hat{\theta}_2^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates  $\hat{\theta}$ .

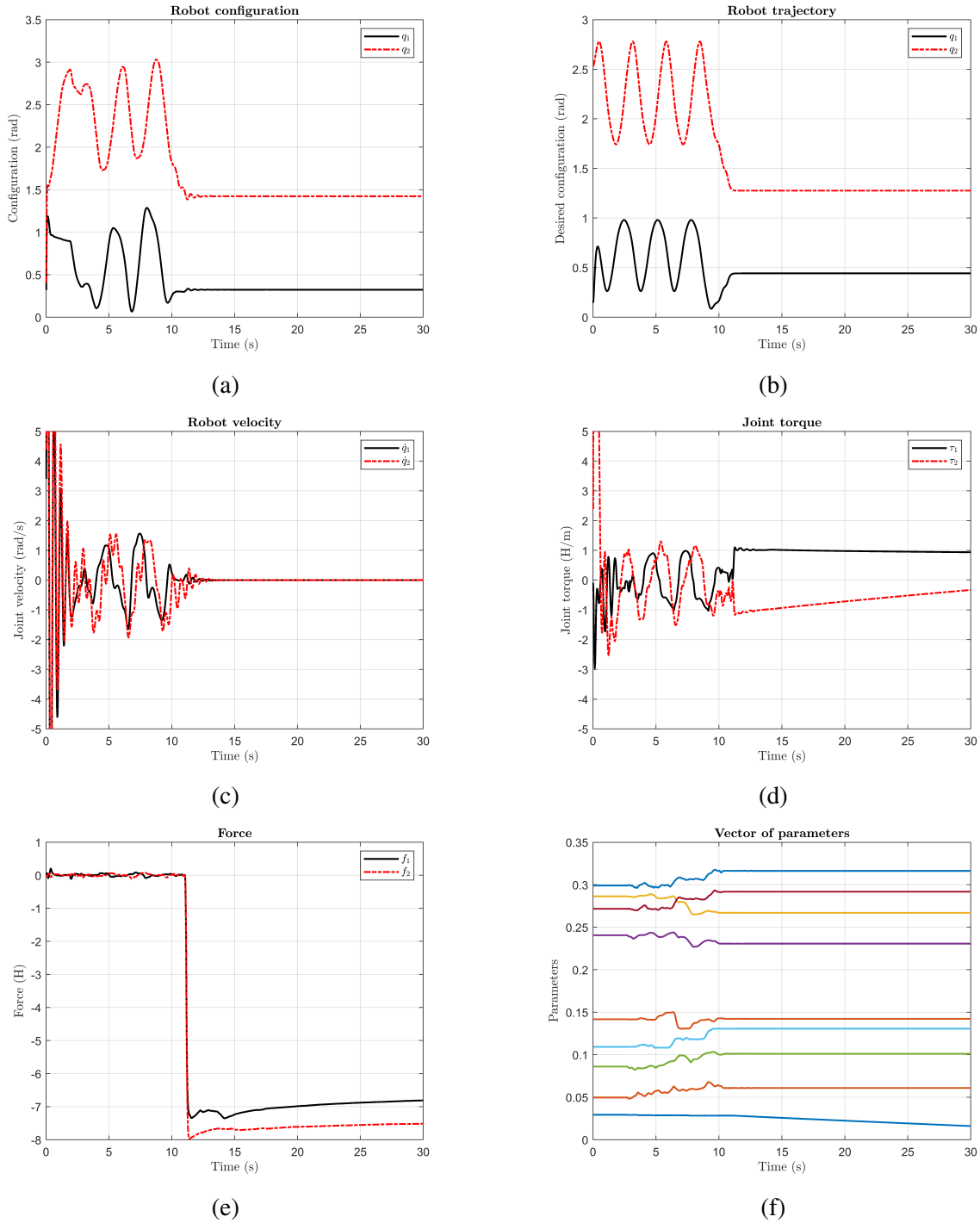


Figure 4.6: Experimental results: the scattering transformation is applied, trajectory execution time 10.7 s, initial parameters  $\hat{\theta}(0) = \hat{\theta}_2^{init}$ : (a) robot’s joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates  $\hat{\theta}$ .

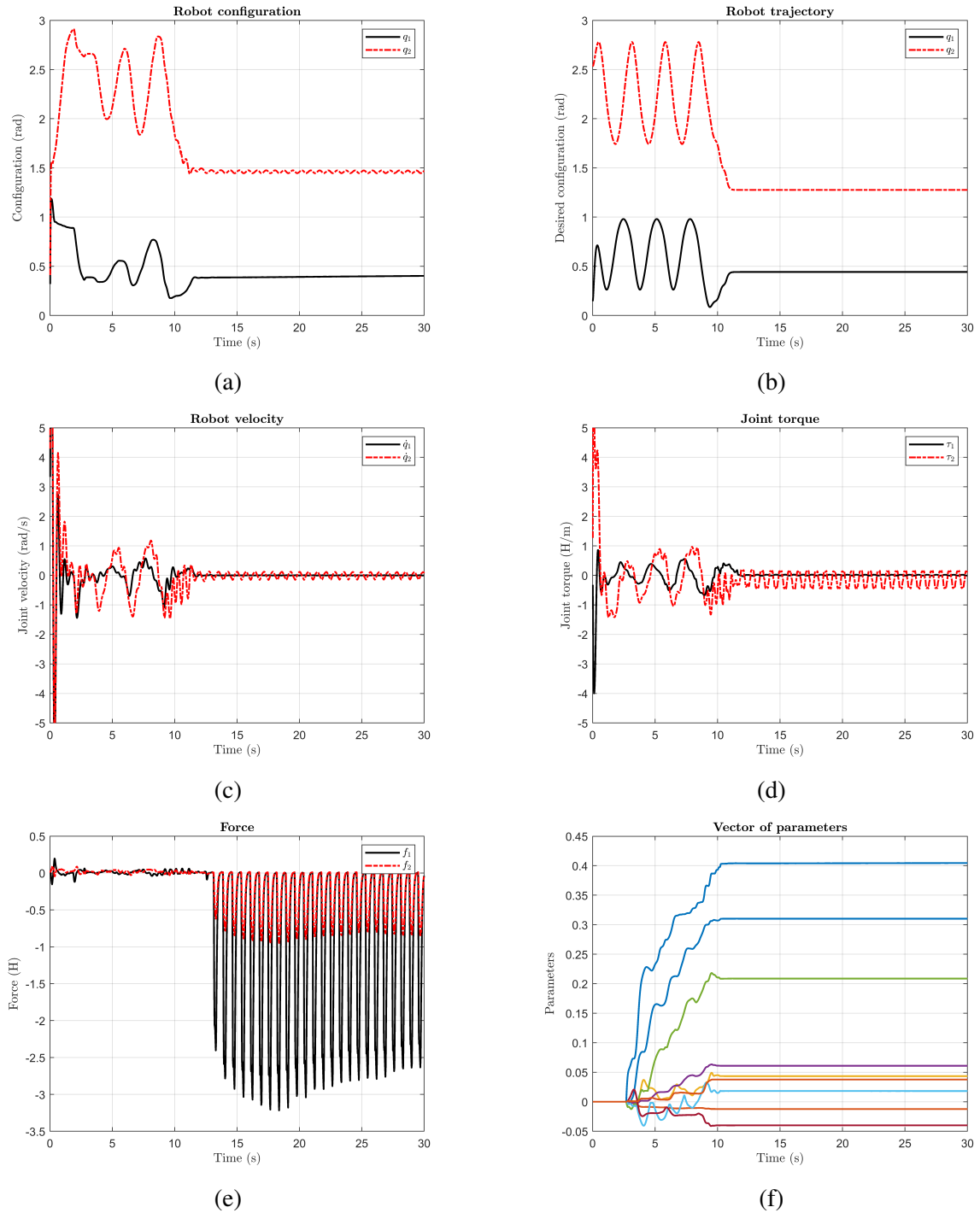


Figure 4.7: Experimental results: no scattering transformation, trajectory execution time 10.7 s, initial parameters  $\hat{\theta}(0) = \hat{\theta}_3^{init} = \mathbb{O}_{12}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates  $\hat{\theta}$ .

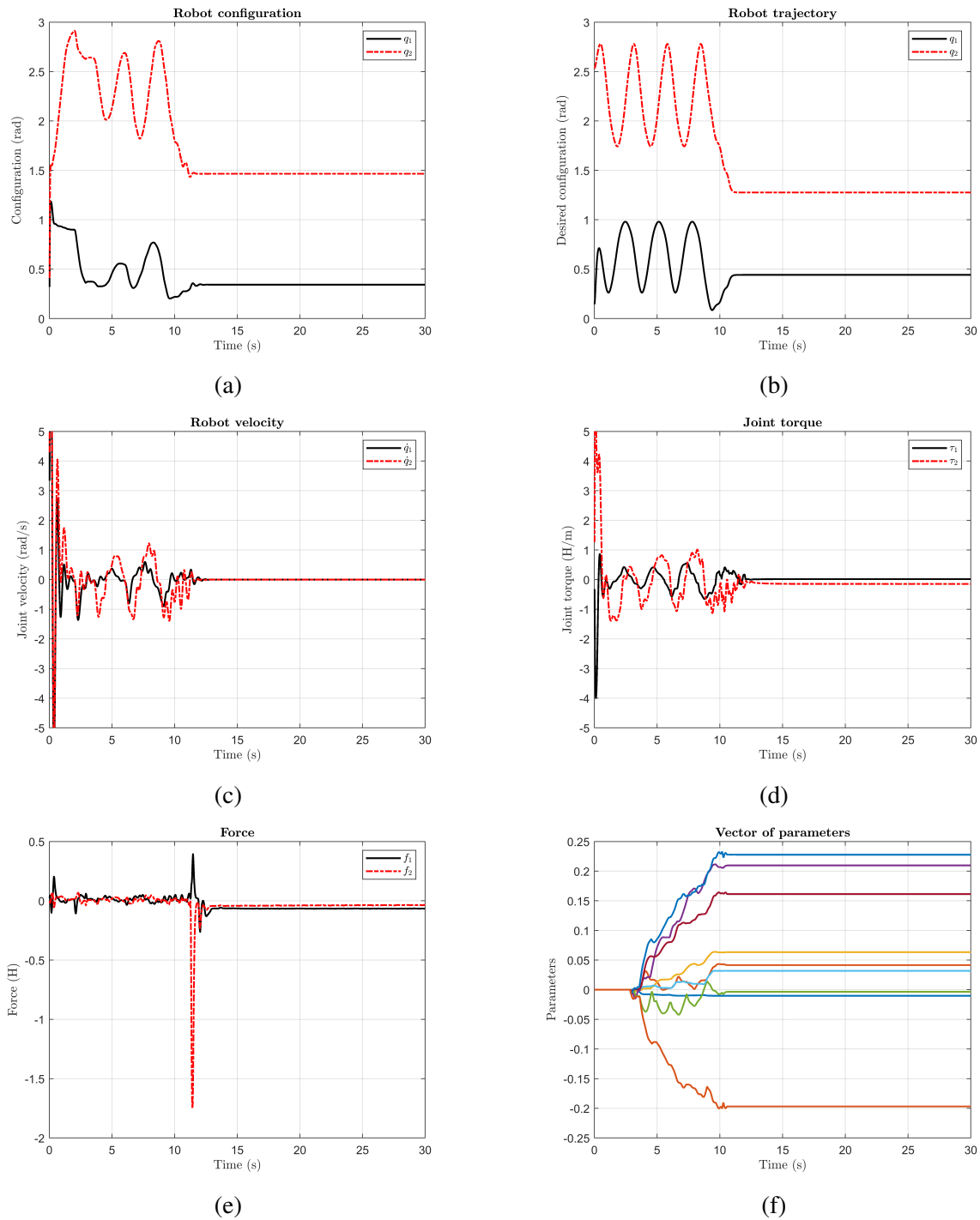


Figure 4.8: Experimental results: the scattering transformation is applied, trajectory execution time 10.7 s,  $\hat{\theta}(0) = \hat{\theta}_3^{init} = \mathbb{O}_{12}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates  $\hat{\theta}$ .

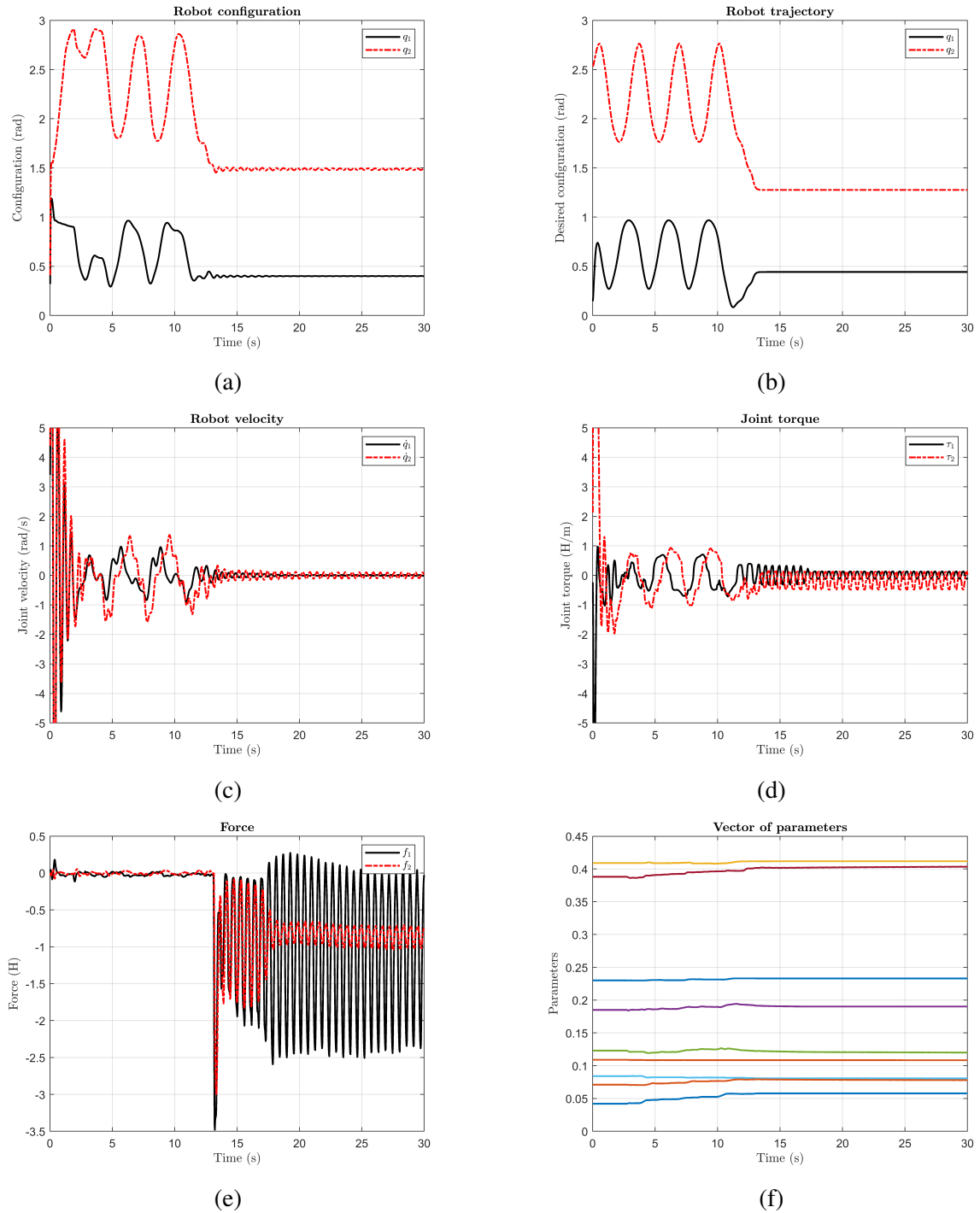


Figure 4.9: Experimental results: no scattering transformation, trajectory execution time 12.8 s, initial parameters  $\hat{\theta}(0) = \hat{\theta}_1^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates  $\hat{\theta}$

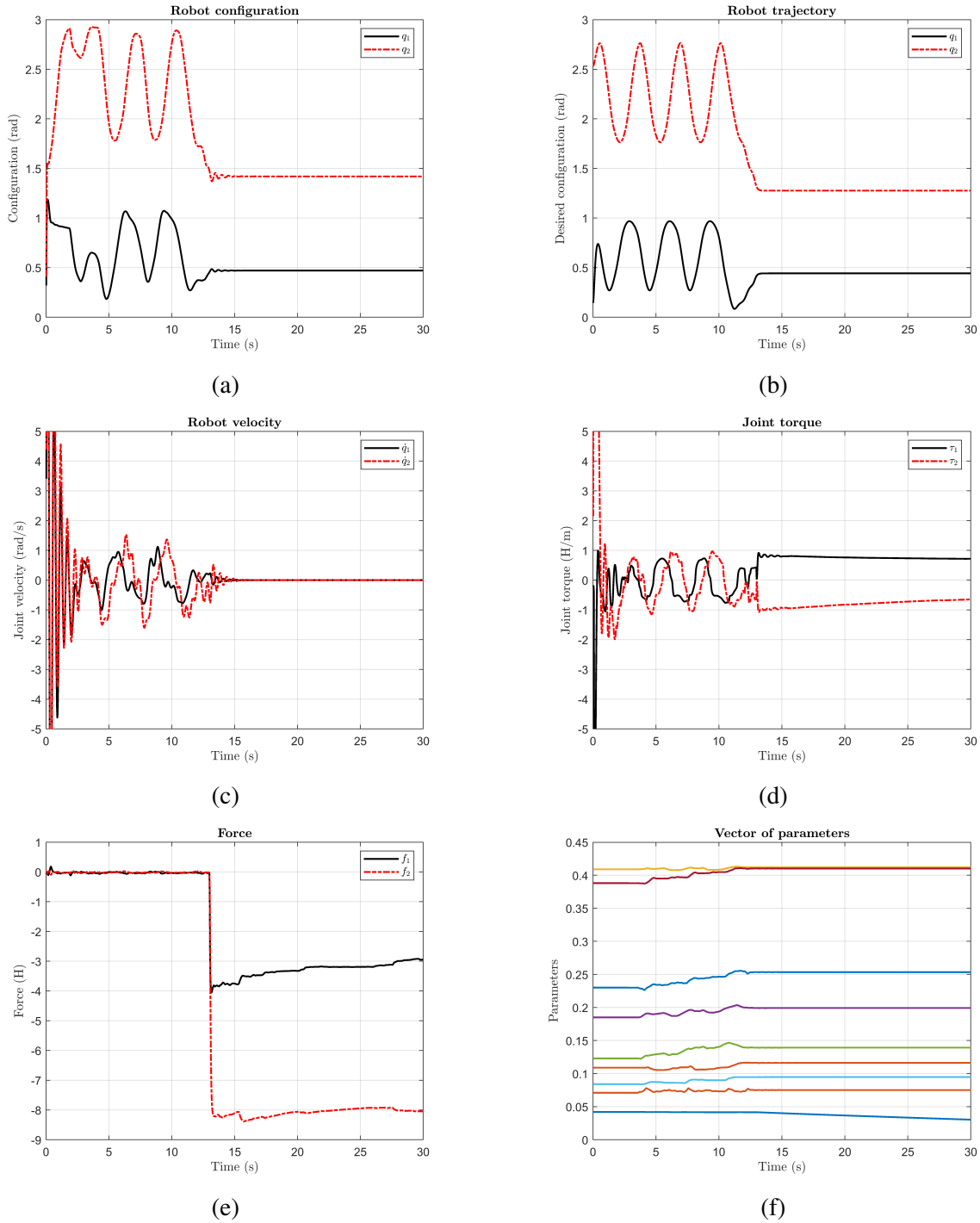


Figure 4.10: Experimental results: the scattering transformation is applied, trajectory execution time 12.8 s, initial parameters  $\hat{\theta}(0) = \hat{\theta}_1^{init}$ : (a) robot’s joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates  $\hat{\theta}$



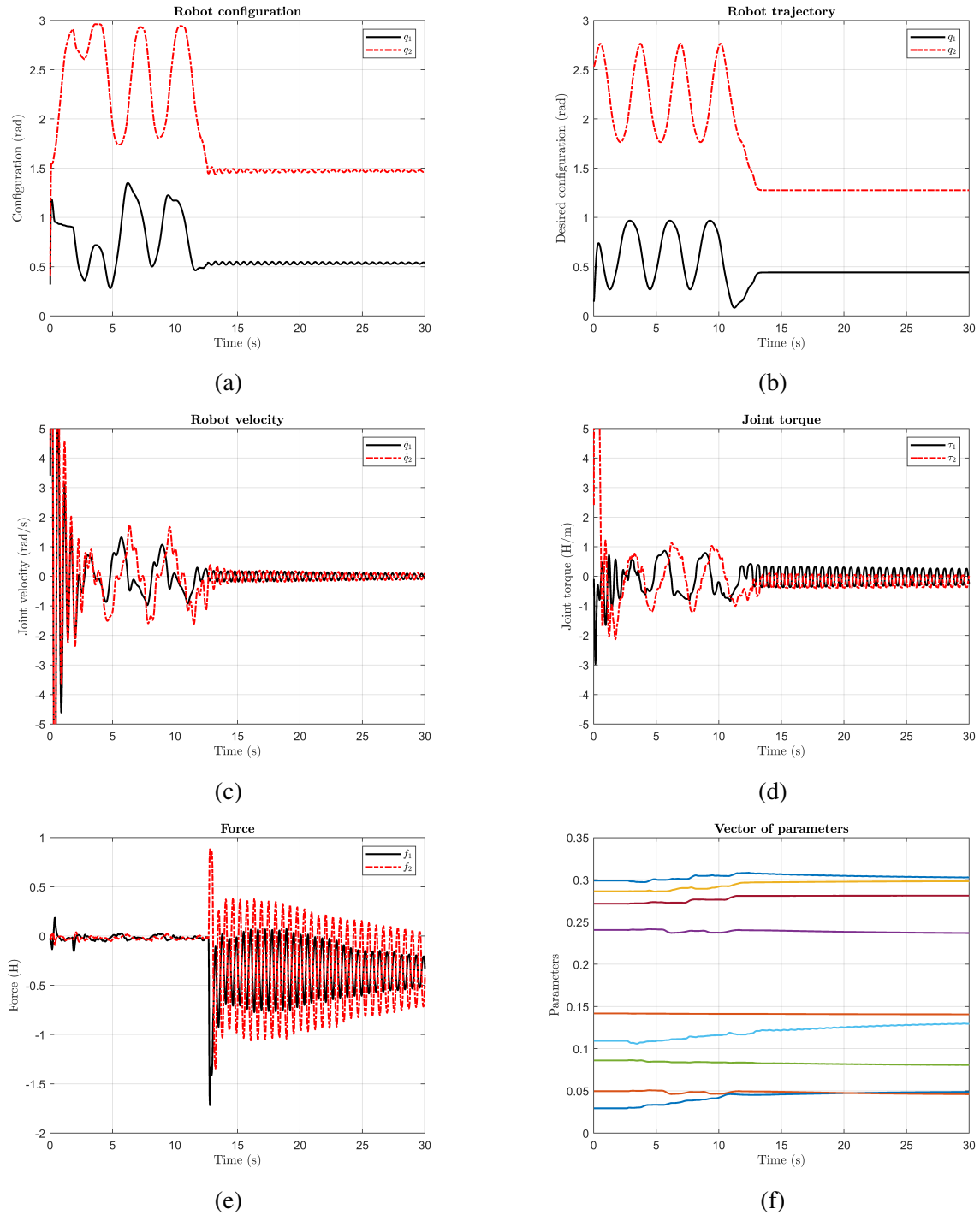


Figure 4.11: Experimental results: no scattering transformation, trajectory execution time 12.8 s, initial parameters  $\hat{\theta}(0) = \hat{\theta}_2^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates  $\hat{\theta}$ .

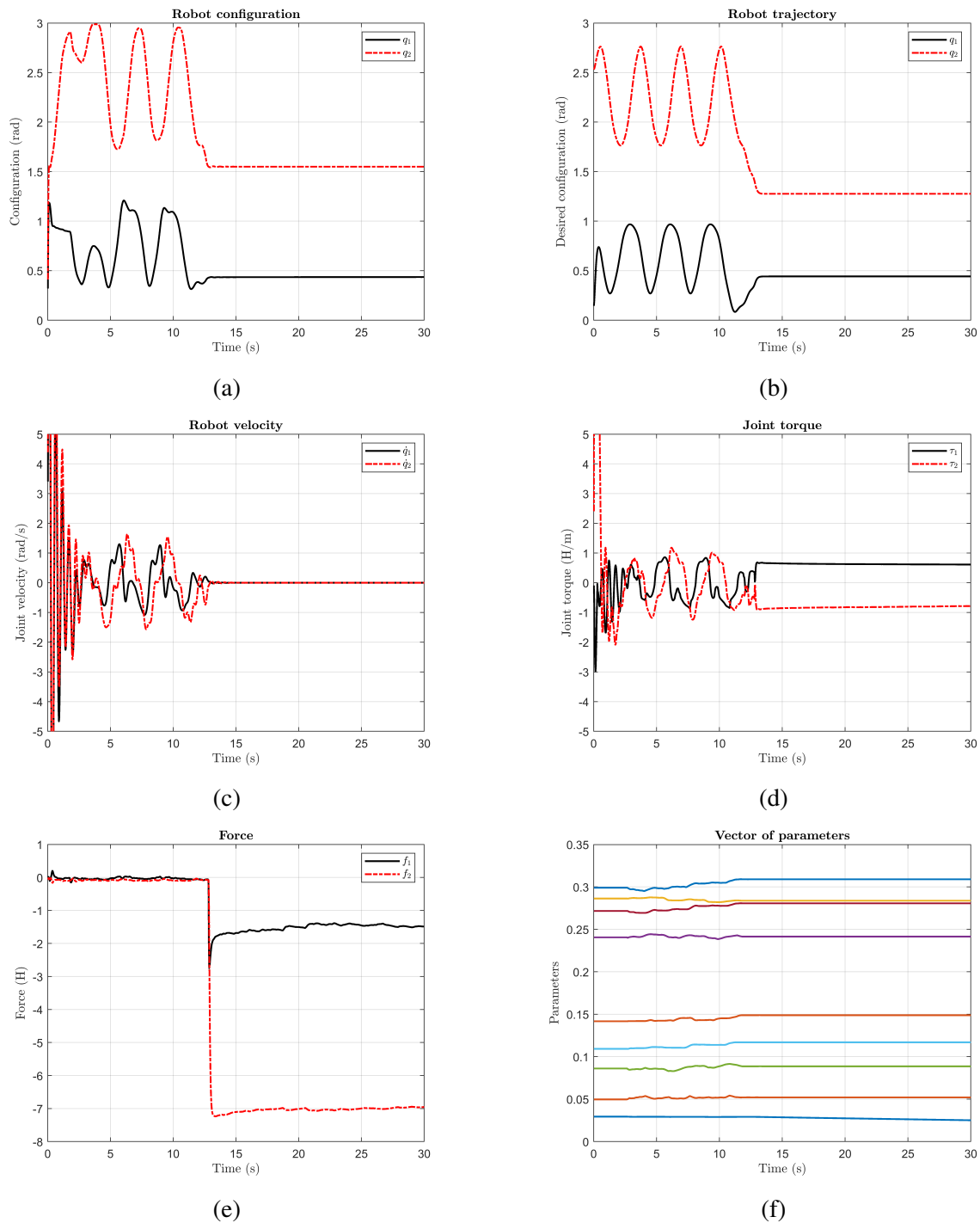


Figure 4.12: Experimental results: the scattering transformation is applied, trajectory execution time 12.8 s, initial parameters  $\hat{\theta}(0) = \hat{\theta}_2^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates  $\hat{\theta}$ .

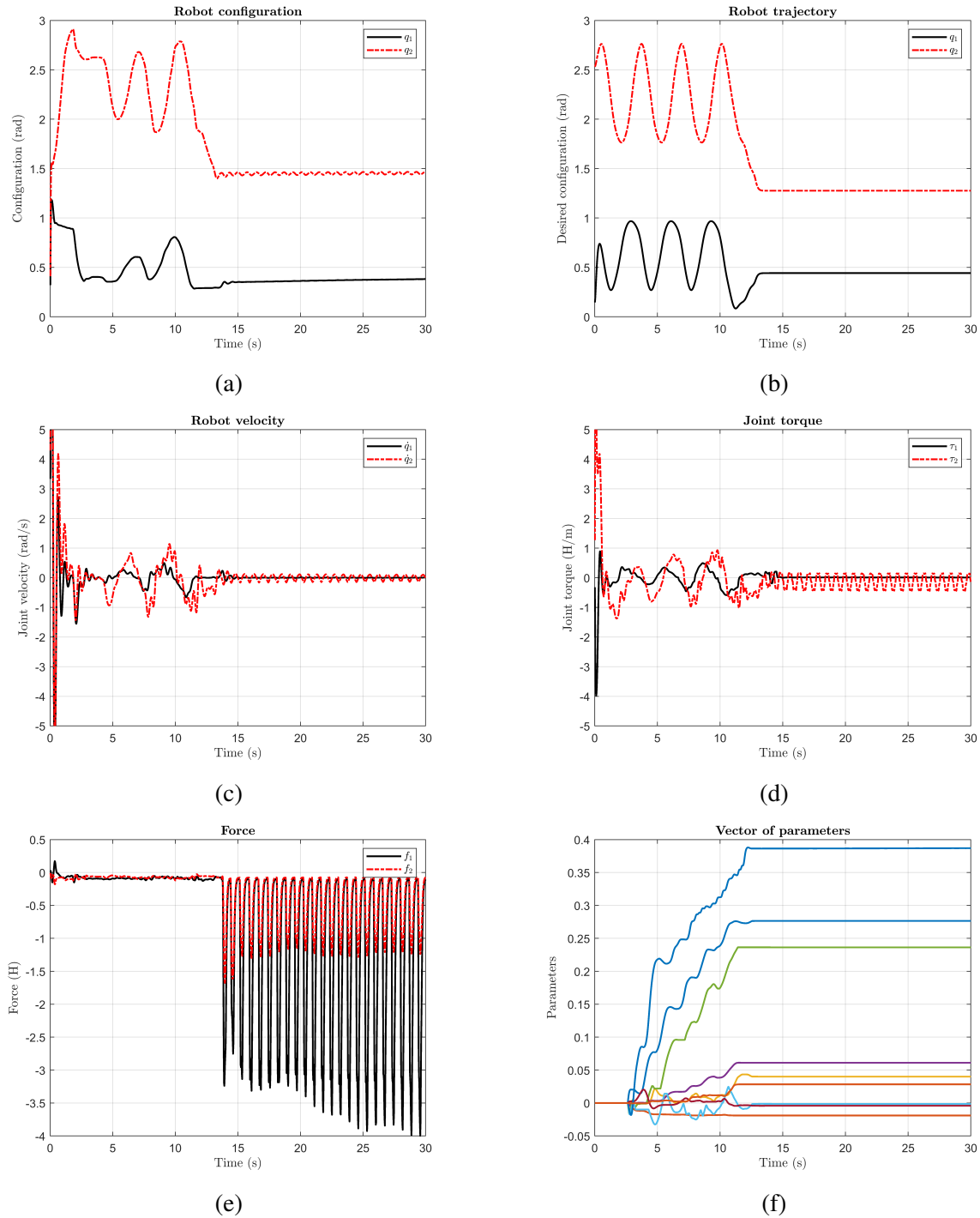


Figure 4.13: Experimental results: no scattering transformation, trajectory execution time 12.8 s, initial parameters  $\hat{\theta}(0) = \hat{\theta}_3^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates  $\hat{\theta}$ .

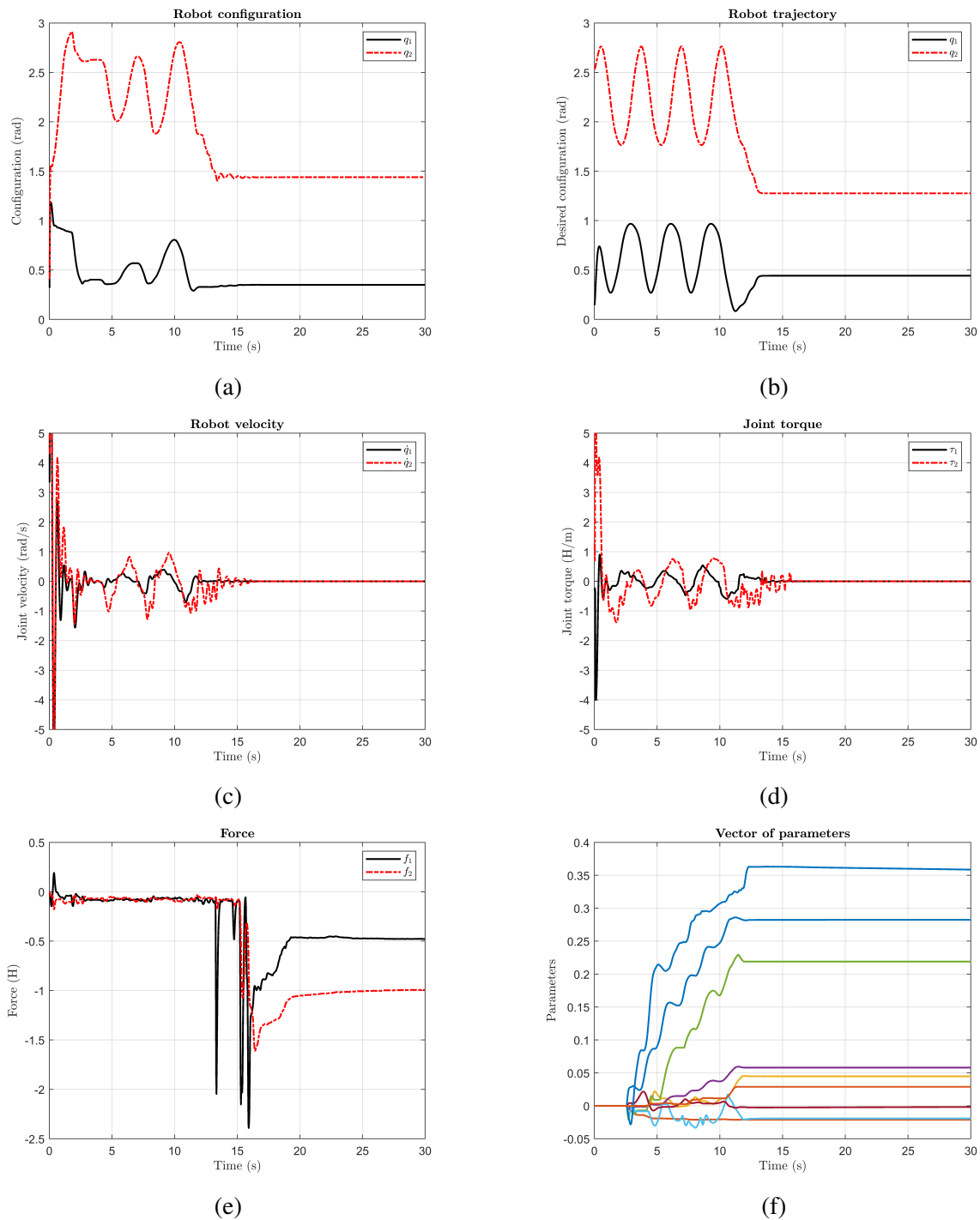


Figure 4.14: Experimental results: the scattering transformation is applied, trajectory execution time 12.8 s, initial parameters  $\hat{\theta}(0) = \hat{\theta}_3^{init}$ : (a) robot's joint trajectories, (b) desired joint trajectories, (c) joint velocities, (d) commanded torques, (e) end-effector forces, (f) parameter estimates  $\hat{\theta}$ .

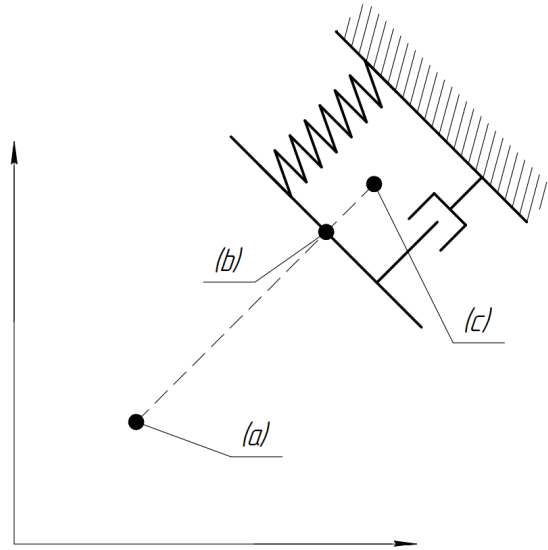


Figure 4.15: Scheme of the environment used in simulation. Trajectory is shown with the dashed line. The following are the trajectory waypoints: (a) - starting point, (b) - point of collision, (c) - desired end point.

where  $a$  and  $v$  are the robot's acceleration and velocity in task space,  $p_\delta$  is the delta vector that represents the compression of the environment. Vector  $p_\delta$  is oriented normally to the surface of the environment and its norm is equal to the amount of compression of the environment. This way, when the robot collides with the environment, the interaction force is always normal to the surface. The matrices  $D$  and  $K$  were chosen as:

$$K = \begin{bmatrix} 70 & 0 \\ 0 & 70 \end{bmatrix}, D = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}. \quad (4.5)$$

The exact coefficients in matrices  $K$  and  $D$  were chosen such that the dynamics of the simulated environment is very similar to the real environment used in experiments. The position of the end-effector was calculated using forward kinematics of the manipulator.

To make the simulation as realistic as possible, the following features were implemented. First, the joint positions measurements were quantized with step  $\delta_q = 7 \cdot 10^{-5}$ . Second, noise was added to the force readings. This noise has the same mean and standard deviation as

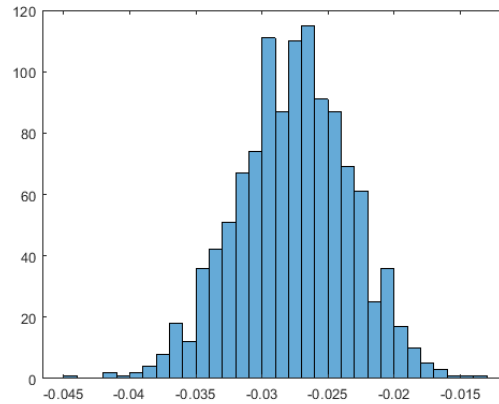


Figure 4.16: Distribution of the force along the  $x$  axis of the Nano 43 sensor when the external force is equal to zero. The mean  $\mu_x = -0.0276$  H,  $\mu_y = 0.0085$  H; the standard deviation  $\sigma_x = 0.0043$  H and  $\sigma_y = 0.0059$  H.

the noise of the sensor used in the experiments. It was assumed that this noise is normally distributed. The mean and standard deviation were found by fitting the Gaussian curve to the force readings when the external force is equal to 0. The distribution of the noise along the  $x$  axis of the force sensor is shown in Figure 4.16. Further, the acceleration of the robot was bounded by  $\pm 5 \text{ rad/s}^2$ . This is based on the fact that the acceleration of the real robot never exceeds this limit and thus this modification makes the behaviour of the robot in simulation closer to that of the real robot. The velocity and acceleration readings were calculated using a second-order low-pass filter. The desired trajectory of the manipulator is exactly the same as the one used in the real experiments, except it does not have the circular components which originally were intended to allow the robot to estimate its coefficients (3.29). That is, the trajectory starts at the beginning of the straight path (see Figure 4.1) and then goes towards the environment.

#### 4.4.2 Simulation Results

The derivatives of the joint positions were found using a second-order low-pass filter with cut-off frequencies of 50 Hz for the first derivative and 30 Hz for the second. The damping was

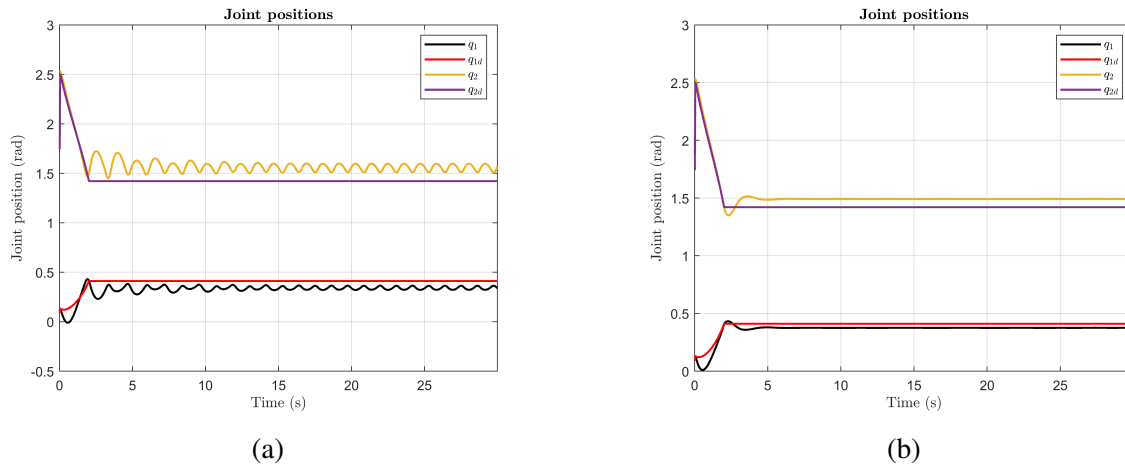


Figure 4.17: Desired and actual joint positions of the robot in simulation: (a) cutoff frequency of the low-pass filter is 50 Hz, (b) cutoff frequency of the low-pass filter is 100 Hz.

equal to 1 for both of derivatives. These values correspond to the values used in the filter in the real experiments. The desired and actual joint positions of the robot are shown in Figure 4.17a. It should be noted, that after collision with the environment, the robot starts to oscillate; similar behaviour can be observed in the real experiments. However, if we increase the cutoff frequency of the low-pass filter from 50 Hz to 100 Hz, the oscillations are not seen (see Figure 4.17b). This indicates that the root cause of the oscillations in the interconnected system is the presence of delays in sensor data, which in turn is caused by the filtering. It should be noted, that on a real robot the cutoff frequency of 100 Hz causes chattering in the commanded torque and hence in the robot motion. This is due to the noise in the joint position readings. Thus, to find the actual cause of the oscillations solely using the physical robot is not possible.

## 4.5 Conclusion

The results of the experiments with the scattering transformation on a real coupled system were presented in this chapter. The chapter started with a description of how velocities and accelerations are estimated based on the position measurements. That is, given that the robot used in the experiments does not provide joint velocity and acceleration measurements, these

signals have to be reconstructed by taking the derivatives from the joint positions. The implemented high-order sliding mode observer [49] provides precise estimation of the first and the second derivatives of the input signals. Experimentally it was verified that the velocity and acceleration estimates provided by the observer can be used as inputs to the trajectory control algorithm. The experiments were conducted for two different trajectories with three different initial estimates of parameter vector  $\hat{\theta}^{init}$ . The experiments showed that due to the non-passive nature of the interconnected system, instabilities occur. However, the application of the scattering transformation allows to stabilize the system. This is in a complete accordance with the theoretical developments in the previous chapter. Section 4.4 described the process of creating a digital twin of a coupled system. This indicated that the root cause of the instabilities lies in the design of a filter that is used to estimate joint velocities and acceleration.



# Chapter 5

## Conclusion

### 5.1 Summary

This thesis presented design, implementation, and experimental evaluation of a framework for stable robot-environment interaction. The framework is based on the use of a non-planar conic system formalism and the generalized scattering transformation techniques. A comprehensive overview of interaction control methods and the scattering transformation techniques was presented in Chapter 1. In addition, Chapter 1 summarized objectives and motivation behind the research. That is, the conventional passivity-based approaches for the coupled stability problem are limited to the case of passive interaction; however, there are many examples of subsystems that are non-passive. Likewise, even if subsystems are passive, the overall interconnected system may be non-passive if the outputs used for the interconnection are not passive outputs.

Chapter 2 presented a theoretical background on conic systems and scattering transformation. These theoretical developments form a basis for the implementation of the scattering-based framework for stabilization of the robot-environment interaction. All the necessary steps for the deployment of the scattering-based stabilization framework on a real physical system were presented in chapter 3. This chapter described the derivation of the dynamics equations for the robot, which is necessary for the implementation of the trajectory-tracking control al-

gorithm. Furthermore, the general structure of the dynamics equations is necessary for the implementation of the scattering transformation. To estimate unknown robot parameters and to capture unmodeled dynamics the two data-driven approaches were used. Both dynamics estimation methods: Linear regression and Neural Network-based showed similar performance. The mean absolute error on the testing set is less than 3% of the torque range. Next, the dynamics of the environment were described, which is also necessary for the estimation of the dynamic cone of the environment and further design of the scattering transformation. Further, the dissipativity properties and dynamic cones alignment of the robot and environment systems were analyzed. It was found that stability the closed-loop system cannot be guaranteed, which is in a complete accordance with the experimentally observed instability. Further, the scattering transformation design for stability of the robot-environment interaction and the complete control architecture of the system were presented. Thus, chapter 3 described how the developed stabilization framework can be applied on a real system.

The evaluation of the proposed framework on the robot-environment system was presented in Chapter 4. The following results have been achieved in this chapter. First, it was shown that the Lyapunov-based adaptive control algorithm shows stable behaviour and is able to reliably track the reference trajectory in free space. This, in turn, demonstrates that all the parts of the framework that were employed in the trajectory control algorithm are working correctly. That is, the high-order sliding mode observer that was implemented provides estimations of velocity and acceleration signals that are precise enough for desirable performance of the trajectory control algorithm. In addition, it demonstrates that the dynamics of the robot were derived correctly. Second, experiments showed that the robot-environment interaction is not stable due to its non-passive nature. However, the application of the proposed stabilization framework enabled stability to be achieved without affecting the robot's trajectory-tracking performance in free space. This result is in complete accordance with the theoretical developments. The method presented in this work constitutes a direct extension of the existing passivity based approaches for the coupled stability problem. Furthermore, chapter 4 described the process

of creating a digital twin of a coupled system. This provided more insights on the behaviour of a real system and allowed to understand that the root cause of instabilities lies in minor discrepancies between the real and the estimated values of the joint velocities and accelerations.

## 5.2 Future Research

Further research is required in order to extend the proposed method to benefit from its full potential as well as to identify its possible shortcomings. In particular, a complete analytical solution of the scattering-based design problem for coupled stability subject to constraints such as (3.50) is a topic for future research. Another future direction for this research work is to evaluate the performance of the proposed approach for haptic teleoperation in surgical applications. This involves solving a number of issues, particularly related to the design, fabrication, and modeling of the system. For example, there will be a requirement imposed on the slave trajectory-tracking performance. That is, the slave manipulator will be required to follow the trajectory that is being generated by the master manipulator. Another requirement can be that the system should avoid the so-called "wave reflection" phenomena observed in teleoperators. This can be mitigated by applying scattering transformation on the both sides of the communication channel, *i.e.*, both from the master and the slave's sides. Therefore, adaptation of the proposed stabilization framework to the teleoperator systems will require additional theoretical developments and it will be an important next step of this research. Finally, data-driven approaches for conic parameter estimation need to be explored. This will allow to avoid the modeling of a system and the search of parameters based on the model. Instead, it might be possible to estimate the parameters of a dynamic cone directly from the experiments.

# Bibliography

- [1] B.D.O. Anderson. The small-gain theorem, the passivity theorem and their equivalence. *Journal of the Franklin Institute*, 293(2):105–115, 1972.
- [2] Robert J. Anderson and Mark W. Spong. Bilateral control of teleoperators with time delay. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, volume 1, pages 131–138. IEEE, 1988.
- [3] Suguru Arimoto, Sadao Kawamura, and Fumio Miyazaki. Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems. In *The 23rd IEEE Conference on Decision and Control*, pages 1064–1069. IEEE, 1984.
- [4] Suguru Arimoto, Sadao Kawamura, and Fumio Miyazaki. Bettering operation of robots by learning. *Journal of Robotic systems*, 1(2):123–140, 1984.
- [5] Seyed Farokh Atashzar, Ilia G. Polushin, and Rajnikant V. Patel. A small-gain approach for nonpassive bilateral telerobotic rehabilitation: Stability analysis and controller synthesis. *IEEE Transactions on Robotics*, 33(1):49–66, 2016.
- [6] A.N. Atassi and H.K. Khalil. Separation results for the stabilization of nonlinear systems using different high-gain observer designs. *Systems & Control Letters*, 39(3):183–191, 2000.

- [7] N.E. Barabanov, A.K. Gelig, G.A. Leonov, A.L. Likhtarnikov, A.S. Matveev, V.B. Smirnova, and A.L. Fradkov. Frequency theorem (yakubovich-kalman lemma) in the control theory. *Avtomatika i Telemekhanika*, (9):3–40, 1996.
- [8] Giorgio Bartolini, Alessandro Pisano, Elisabetta Punta, and Elio Usai. A survey of applications of second-order sliding mode control to mechanical systems. *International Journal of Control*, 76(9-10):875–892, 2003.
- [9] Dimitri Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific, 2012.
- [10] David J. Braun, Florian Petit, Felix Huber, Sami Haddadin, Patrick Van Der Smagt, Alin Albu-Schäffer, and Sethu Vijayakumar. Optimal torque and stiffness control in compliantly actuated robots. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2801–2808. IEEE, 2012.
- [11] Jonas Buchli, Freek Stulp, Evangelos Theodorou, and Stefan Schaal. Learning variable impedance control. *The International Journal of Robotics Research*, 30(7):820–833, 2011.
- [12] Stephen P. Buerger and Neville Hogan. Complementary stability and loop shaping for improved human–robot interaction. *IEEE Transactions on Robotics*, 23(2):232–244, 2007.
- [13] Chien-Chern Cheah and Danwei Wang. Learning impedance control for robotic manipulators. *IEEE Transactions on Robotics and Automation*, 14(3):452–465, 1998.
- [14] Chien-Chern Cheah, Yu Zhao, and Jean-Jacques E. Slotine. Adaptive jacobian motion and force tracking control for constrained robots with uncertainties. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2226–2231. IEEE, 2006.

- [15] Moshe Cohen and Tamar Flash. Learning impedance parameters for robot control using an associative search network. *IEEE Transactions on Robotics and Automation*, 7(3):382–390, 1991.
- [16] Richard Colbaugh, Homayoun Seraji, and Kristin Glass. Direct adaptive impedance control of robot manipulators. *Journal of Robotic Systems*, 10(2):217–248, 1993.
- [17] J. E. Colgate. Coordinate transformations and logical operations for minimizing conservativeness in coupled stability criteria. *Journal of Dynamic Systems, Measurement, and Control*, 116(4):643–649, 12 1994.
- [18] J. Edward Colgate. Stability of manipulators interacting with unstructured and structured dynamic environments. In *Robotics, Mechatronics and Manufacturing Systems*, pages 269–274. Elsevier, 1993.
- [19] Emmanuel Cruz-Zavala, Jaime A. Moreno, and Leonid Fridman. Uniform second-order sliding mode observer for mechanical systems. In *2010 11th International Workshop on Variable Structure Systems (VSS)*, pages 14–19. IEEE, 2010.
- [20] Marcio S. De Queiroz, Jun Hu, Darren M. Dawson, Timothy Burg, and Sreenivasa R. Donepudi. Adaptive position/force control of robot manipulators without velocity measurements: Theory and experimentation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 27(5):796–809, 1997.
- [21] Dick De Roover, Okko H. Bosgra, and Maarten Steinbuch. Internal-model-based design of repetitive and iterative learning controllers for linear multivariable systems. *International Journal of Control*, 73(10):914–929, 2000.
- [22] Fotios Dimeas, Vassilis C Moulitanitis, and Nikos Aspragathos. Manipulator performance constraints in human-robot cooperation. *Robotics and Computer-Integrated Manufacturing*, 50:222–233, 2018.

- [23] Vincent Duchaine and Clement M. Gosselin. General model of human-robot cooperation using a novel velocity based variable impedance control. In *Second Joint Euro-Haptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07)*, pages 446–451. IEEE, 2007.
- [24] Christopher Edwards, Sarah K. Spurgeon, and Chee Pin Tan. On the development and application of sliding mode observers. In *Variable Structure Systems: Towards the 21st Century*, pages 253–282. Springer, 2002.
- [25] Fanny Ficuciello, Luigi Villani, and Bruno Siciliano. Variable impedance control of redundant manipulators for intuitive human–robot physical interaction. *IEEE Transactions on Robotics*, 31(4):850–863, 2015.
- [26] Rafael Fierro and Frank L. Lewis. Control of a nonholonomic mobile robot using neural networks. *IEEE Transactions on Neural Networks*, 9(4):589–600, 1998.
- [27] Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852*, 2016.
- [28] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*, pages 49–58. PMLR, 2016.
- [29] Charles Fox. *An introduction to the calculus of variations*. Courier Corporation, 1987.
- [30] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.
- [31] Mohammad-Hossein Ghajar, Mehdi Keshmiri, and Javad Bahrami. Neural-network-based robust hybrid force/position controller for a constrained robot manipulator with

- uncertainties. *Transactions of the Institute of Measurement and Control*, 40(5):1625–1636, 2018.
- [32] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. JHU press, 2013.
- [33] David J. Hill and Peter James Moylan. Stability results for nonlinear feedback systems. *Automatica*, 13(4):377–382, 1977.
- [34] Sandra Hirche, Tilemachos Matiakis, and Martin Buss. A distributed controller approach for delay-independent stability of networked control systems. *Automatica*, 45(8):1828–1836, 2009.
- [35] T. Hoshikawa and A. Sudou. Dynamic hybrid position/force control of robot manipulators: online estimation of unknown constraint. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 1231–1236 vol.2, 1990.
- [36] Ryojun Ikeura and Hikaru Inooka. Variable impedance control of a robot for cooperation with a human. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 3, pages 3097–3102. IEEE, 1995.
- [37] Ryojun Ikeura, Tomoki Moriguchi, and Kazuki Mizutani. Optimal variable impedance control for a robot and its application to lifting an object with a human. In *Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication*, pages 500–505. IEEE, 2002.
- [38] Rolf Johansson and Mark W. Spong. Quadratic optimization of impedance control. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 616–621. IEEE, 1994.
- [39] Velimir Jurdjevic and John P. Quinn. Controllability and stability. *Journal of differential equations*, 28(3):381–389, 1978.



- [40] N. Kalouptsidis and J. Tsinias. Stability improvement of nonlinear systems by feedback. *IEEE Transactions on Automatic Control*, 29(4):364–367, 1984.
- [41] Sungchul Kang, Kiyoshi Komoriya, Kazuhito Yokoi, Tetsuo Koutoku, Byungchan Kim, and Shinsuk Park. Control of impulsive contact force between mobile manipulator and environment using effective mass and damping controls. *International Journal of Precision Engineering and Manufacturing*, 11(5):697–704, 2010.
- [42] Seiichi Kobayashi and Katsuhisa Furuta. Frequency characteristics of levant’s differentiator and adaptive sliding mode differentiator. *International Journal of Systems Science*, 38(10):825–832, 2007.
- [43] C.M. Kwan. Robust adaptive force/motion control of constrained robots. *IEE Proceedings-Control Theory and Applications*, 143(1):103–109, 1996.
- [44] Pascal D. Labrecque and Clément Gosselin. Variable admittance for phri: from intuitive unilateral interaction to optimal bilateral force amplification. *Robotics and Computer-Integrated Manufacturing*, 52:1–8, 2018.
- [45] Philippe LeBel, Clément Gosselin, and Alexandre Campeau-Lecours. An anticipative kinematic limitation avoidance algorithm for collaborative robots: Three-dimensional case. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3075–3080. IEEE, 2017.
- [46] Alexandre Lecours, Boris Mayer-St-Onge, and Clément Gosselin. Variable admittance control of a four-degree-of-freedom intelligent assist device. In *2012 IEEE international Conference on Robotics and Automation*, pages 3903–3908. IEEE, 2012.
- [47] Kyun K. Lee and Aristotle Arapostathis. Remarks on smooth feedback stabilization of nonlinear systems. In *1986 American Control Conference*, pages 385–386. IEEE, 1986.

- [48] PM Lerman. Fitting segmented regression models by grid search. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 29(1):77–84, 1980.
- [49] Arie Levant. Higher-order sliding modes, differentiation and output-feedback control. *International Journal of Control*, 76(9-10):924–941, 2003.
- [50] Sergey Levine and Vladlen Koltun. Continuous inverse optimal control with locally optimal examples. *arXiv preprint arXiv:1206.4617*, 2012.
- [51] Frank L. Lewis and Draguna Vrabie. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits and Systems Magazine*, 9(3):32–50, 2009.
- [52] Weihua Li, Liang Ding, Haibo Gao, and Mahdi Tavakoli. Kinematic bilateral teleoperation of wheeled mobile robots subject to longitudinal slippage. *IET Control Theory & Applications*, 10(2):111–118, 2016.
- [53] Weihua Li, Haibo Gao, Liang Ding, and Mahdi Tavakoli. Trilateral predictor-mediated teleoperation of a wheeled mobile robot with slippage. *IEEE Robotics and Automation Letters*, 1(2):738–745, 2016.
- [54] Yanan Li, Shuzhi Sam Ge, and Chenguang Yang. Learning impedance control for physical robot–environment interaction. *International Journal of Control*, 85(2):182–193, 2012.
- [55] Ming C. Lin and Miguel Otaduy. *Haptic rendering: foundations, algorithms, and applications*. CRC Press, 2008.
- [56] Wei Lin. Global asymptotic stabilization of general nonlinear systems with stable free dynamics via passivity and bounded feedback. *Automatica*, 32(6):915–924, 1996.

- [57] Xing Liu, Shuzhi Sam Ge, Fei Zhao, and Xuesong Mei. Optimized impedance adaptation of robot manipulator interacting with unknown environment. *IEEE Transactions on Control Systems Technology*, 29(1):411–419, 2020.
- [58] R. Lozano and B. Brogliato. Adaptive hybrid force-position control for redundant manipulators. *IEEE Transactions on Automatic Control*, 37(10):1501–1505, 1992.
- [59] W-S Lu and Q-H Meng. Impedance control with adaptation for robotic manipulations. *IEEE Transactions on Robotics and Automation*, 7(3):408–415, 1991.
- [60] Roberto Martín-Martín, Michelle A Lee, Rachel Gardner, Silvio Savarese, Jeannette Bohg, and Animesh Garg. Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1010–1017. IEEE, 2019.
- [61] Matthew T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432, 1981.
- [62] M. Matinfar and Keyvan Hashtrudi-Zaad. Optimization-based robot compliance control: Geometric and linear quadratic approaches. *The International Journal of Robotics Research*, 24(8):645–656, 2005.
- [63] D. Nganga-Kouya, Maarouf Saad, and Louis Lamarche. Backstepping adaptive hybrid force/position control for robotic manipulators. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, volume 6, pages 4595–4600. IEEE, 2002.
- [64] Günter Niemeyer and J.J. Slotine. Stable adaptive teleoperation. *IEEE Journal of Oceanic Engineering*, 16(1):152–162, 1991.
- [65] Emmanuel Nuño, Luis Basañez, and Romeo Ortega. Passivity-based control for bilateral teleoperation: A tutorial. *Automatica*, 47(3):485–495, 2011.

- [66] Romeo Ortega and Mark W. Spong. Adaptive motion control of rigid robots: A tutorial. *Automatica*, 25(6):877–888, 1989.
- [67] I.G. Polushin, A.L. Fradkov, and D.J. Hill. Passivity and passification of non-linear systems. *Automatics and Telemechanics*, (3):3–37, 2000.
- [68] Iliia G. Polushin. A generalization of the scattering transformation for conic systems. *IEEE Transactions on Automatic Control*, 59(7):1989–1995, 2014.
- [69] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 06 1981.
- [70] Isura Ranatunga, Frank L. Lewis, Dan O. Popa, and Shaikh M. Tousif. Adaptive admittance control for human–robot interaction using model reference design and adaptive inverse filtering. *IEEE Transactions on Control Systems Technology*, 25(1):278–285, 2016.
- [71] Joel Rey, Klas Kronander, Farbod Farshidian, Jonas Buchli, and Aude Billard. Learning motions from demonstrations and rewards with time-invariant dynamical systems based policies. *Autonomous Robots*, 42(1):45–64, 2018.
- [72] Jaydeep Roy and Louis L. Whitcomb. Adaptive force control of position/velocity controlled robots: theory and experiment. *IEEE Transactions on Robotics and Automation*, 18(2):121–137, 2002.
- [73] Michael George Safonov. *Stability and Robustness of Multivariable Feedback Systems*. MIT press, 1980.
- [74] Shankar Sastry, Marc Bodson, and James F Bartram. Adaptive control: stability, convergence, and robustness, 1990.
- [75] Mark W. Spong and Romeo Ortega. On adaptive inverse dynamics control of rigid robots. *IEEE Transactions on Automatic Control*, 35(1):92–95, 1990.

- [76] Ranko Zotovic Stanisic and Ángel Valera Fernández. Adjusting the parameters of the mechanical impedance for velocity, impact and force control. *Robotica*, 30(4):583–597, 2012.
- [77] Da Sun, Fazel Naghdy, and Haiping Du. Wave-variable-based passivity control of four-channel nonlinear bilateral teleoperation system under time delays. *IEEE/ASME Transactions on Mechatronics*, 21(1):238–253, 2015.
- [78] Andrew R. Teel. On graphs, conic relations, and input-output stability of nonlinear feedback systems. *IEEE Transactions on Automatic Control*, 41(5):702–709, 1996.
- [79] Andrew R. Teel, T.T. Georgiou, L. Praly, and E. Sontag. Input-output stability. *The Control Handbook*, 4250, 1996.
- [80] Toshio Tsuji, Koji Ito, and Pietro G. Morasso. Neural network learning of robot arm impedance in operational space. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(2):290–298, 1996.
- [81] Toru Tsumugiwa, Ryuichi Yokogawa, and Kei Hara. Variable impedance control with regard to working process for man-machine cooperation-work system. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, volume 3, pages 1564–1569. IEEE, 2001.
- [82] Toru Tsumugiwa, Ryuichi Yokogawa, and Kei Hara. Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 1, pages 644–650. IEEE, 2002.
- [83] Mitsunori Uemura and Sadao Kawamura. Resonance-based motion control method for multi-joint robot through combining stiffness adaptation and iterative learning control.

- In *2009 IEEE International Conference on Robotics and Automation*, pages 1543–1548. IEEE, 2009.
- [84] Anastasiia A. Usova, Kanstantsin A. Pachkouski, Ilia G. Polushin, and Rajni V. Patel. Stabilization of robot-environment interaction through generalized scattering techniques. *IEEE Transactions on Robotics*, 2021.
- [85] Anastasiia A. Usova, Ilia G. Polushin, and Rajni V. Patel. A graph separation stability condition for non-planar conic systems. *IFAC-PapersOnLine*, 49(18):933–938, 2016.
- [86] Anastasiia A. Usova, Ilia G. Polushin, and Rajni V. Patel. Scattering transformation for non-planar conic systems. *IFAC-PapersOnLine*, 50(1):8478–8483, 2017.
- [87] Anastasiia A. Usova, Ilia G. Polushin, and Rajni V. Patel. Scattering-based stabilization of non-planar conic systems. *Automatica*, 93:1–11, 2018.
- [88] Vadim I. Utkin. *Sliding modes in control and optimization*. Springer Science & Business Media, 2013.
- [89] Arjan Van der Schaft. *L2-gain and passivity techniques in nonlinear control*. Springer, 2000.
- [90] Birgit Vogel-Heuser and Dieter Hess. Guest editorial industry 4.0—prerequisites and visions. *IEEE Transactions on Automation Science and Engineering*, 13(2):411–413, 2016.
- [91] Danwei Wang and Chien Chern Cheah. An iterative learning-control scheme for impedance control of robotic manipulators. *The International Journal of Robotics Research*, 17(10):1091–1104, 1998.
- [92] Fei-Yue Wang, Huaguang Zhang, and Derong Liu. Adaptive dynamic programming: An introduction. *IEEE Computational Intelligence Magazine*, 4(2):39–47, 2009.

- [93] Paul J. Werbos. Consistency of hdp applied to a simple reinforcement learning problem. *Neural Networks*, 3(2):179–189, 1990.
- [94] Paul J. Werbos. Intelligence in the brain: A theory of how it works and how to build it. *Neural Networks*, 22(3):200–212, 2009.
- [95] Paul J. Werbos, W.T. Miller, and R.S. Sutton. A menu of designs for reinforcement learning over time. In *Neural Networks for Control*, volume 3, pages 67–95. MIT press Cambridge, MA, 1990.
- [96] David A. White and Donald A. Sofge. Handbook of intelligent control: neural, fuzzy, and adaptive approaches. 1992.
- [97] M. Ronald Wohlers. *Lumped and distributed passive networks: A generalized and advanced viewpoint*. Academic press, 2017.
- [98] Jun Wu, Jinsong Wang, and Zheng You. An overview of dynamic parameter identification of robots. *Robotics and Computer-Integrated Manufacturing*, 26(5):414–419, 2010.
- [99] Di Xiao, Bijoy K. Ghosh, Ning Xi, and Tzyh Jong Tarn. Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment. *IEEE Transactions on Control Systems Technology*, 8(4):635–645, 2000.
- [100] Jian-Xin Xu, Badrinath Viswanathan, and Zhihua Qu. Robust learning control for robotic manipulators with an extension to a class of non-linear systems. *International Journal of Control*, 73(10):858–870, 2000.
- [101] Boo-Ho Yang and Haruhiko Asada. Progressive learning and its application to robot impedance learning. *IEEE Transactions on Neural Networks*, 7(4):941–952, 1996.
- [102] Shelten G. Yuen, Michael C. Yip, Nikolay V. Vasilyev, Douglas P. Perrin, Pedro J. del Nido, and Robert D. Howe. Robotic force stabilization for beating heart intracardiac

- surgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 26–33. Springer, 2009.
- [103] George Zames. On the input-output stability of time-varying nonlinear feedback systems part one: Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE transactions on Automatic Control*, 11(2):228–238, 1966.
- [104] Xiang Zhang, Liting Sun, Zhian Kuang, and Masayoshi Tomizuka. Learning variable impedance control via inverse reinforcement learning for force-related tasks. *IEEE Robotics and Automation Letters*, 6(2):2225–2232, 2021.
- [105] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.



# Curriculum Vitae

**Name:** Kanstantsin Pachkouski

**Post-Secondary Education and Degrees:** ITMO University  
Saint Petersburg  
2015–2019 B.Sc.

Western University  
London, Ontario, Canada  
2020–2022 M.E.Sc.

**Honours and Awards:** 2nd place at Robotex Drone Race, Tallinn, Estonia  
2018

6th place at RoboCup, Montreal, Canada  
2018

**Related Work Experience:** Teaching Assistant  
The University of Western Ontario  
2021

Robotic Systems Engineer  
ICOL Group  
2020

**Publications:** A. A. Usova, K. A. Pachkouski, I. G. Polushin and R. V. Patel, "Stabilization of Robot-Environment Interaction Through Generalized Scattering Techniques," in *IEEE Transactions on Robotics*, 2021

**Presentations:** A. A. Usova, K. A. Pachkouski, I. G. Polushin and R. V. Patel, "Stabilization of Robot-Environment Interaction Through Generalized Scattering Techniques," at *IEEE ICRA, 2022, Philadelphia, PA, US*