MULTISENSORY INTEGRATION FOR ADAPTIVE PHYSICAL HUMAN-ROBOT INTERACTION

by

ISURA RANATUNGA

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2015

To my mother Swarnapalie, father Jayantha, and brother Channa.

ACKNOWLEDGEMENTS

I would like to first thank Dr. Dan O. Popa my supervising professor for encouraging me to pursue a Ph.D. and his constant guidance and support. I appreciate his patience and understanding when I made mistakes. His constant support and encouragement to follow my own ideas. I would also like to thank Dr. Frank L. Lewis for his guidance and collaboration on much of my latest research. Much appreciation also goes to Dr. Michael T. Manry, Dr. Gian-Luca Mariottini, and Dr. Alan Bowling for taking an interest in my research and taking the time to serve on my dissertation committee.

I also thank the University of Texas at Arlington Research Institute (UTARI) and staff for the state of the art robotics facilities and collegial work environment. My thanks also goes to the many colleagues past and present at the Next Generation Systems (NGS) research group who made my time here enjoyable and stimulating. I especially like to thank Sven Cremer, Ahsan Habib, Fahad Mirza, Sumit Das, Namrata Balakrishnan, and Sandesh Gowda for their interesting company and collaboration on my research. I would like to thank my fellow team members of the DARPA Robotics Challenge, especially the humanoid walking team Andrew Somerville, Ghassan Atmeh, and Adrian Rodriguez. We learned and accomplished a lot during the year of the challenge. I also thank my collaborators at the University of North Texas Health Science Center (UNTHSC) for their help with the medical applications of this research.

My thanks go to many of my teachers who had a significant impact in my academic life. Specifically Mrs. Bella who encouraged me to study and Roovi Mendoza who introduced me to computer science.

I greatly appreciate my friends who supported and encouraged me. My childhood friends Sandeep Suresh, Alina Sebastian, and Viraj Shah. My friends Sunny Patel, Andie Bourgeois, Arvind Krishnan, Abhilash Markkassery, and Bito Irie.

I would like to acknowledge the following grants that have supported this research. NSF Grants NRI-1208623, ECCS-1405173, CPS-1035913, CNS-0923494. Defense Advanced Research Project Agency (DARPA) Track-B grant under contract FA8750-12-C-0298. TxMRC consortium grant: Human Robot Interaction System for Early Diagnosis and Treatment of Childhood Autism Spectrum Disorders (RoDiCA). I also thank the University of Texas at Arlington Research Institute (UTARI) for funding me during this time.

Finally much love, thanks, and gratitude to my family, my parents and brother whose unwavering support, love, and understanding helped me overcome many obstacles I faced and without whom I would not be here today.

April 16, 2015

ABSTRACT

MULTISENSORY INTEGRATION FOR ADAPTIVE PHYSICAL HUMAN-ROBOT INTERACTION

Isura Ranatunga, Ph.D.

The University of Texas at Arlington, 2015

Supervising Professor: Dan O. Popa

Automated systems have become increasingly prevalent in the 21st century. With their increased processing power, personal and home robotics has finally come within reach of the public. There are some challenges to be overcome for robots to finally work with and around humans. Some of these challenges include complex social interaction, safety, identifying human intent, unpredictable physical interaction, and uncertain dynamic environments. The focus of this research is safe, robust, and intuitive physical interaction with humans utilizing multiple sensor modalities.

In this dissertation, a solution for adaptive physical human-robot interaction is proposed. The proposed framework consists of three parts. Dynamics compensation: adaptive dynamics compensation is proposed and extended to control complex, non-linear, and constantly changing robotic systems. Adaptive force control: adaptive inverse control techniques are utilized to develop an admittance control system to compensate for changing task, sensor, and human conditions during interaction. Human motion study: Dynamic time warping is utilized as a method for movement

characterization and dynamic movement primitives are used to develop a scalable and adaptable movement representation system.

The proposed framework was inspired by the structure of the human motor cortex and somatosensory systems. An inner-loop control structure performs the function of the lower level fast dynamic compensation system while an outer-loop adaptive force controller enables task, sensor, and human specific control. This type of controller can adapt to the changing dynamics of the robot as well as compensate for the changing environmental sensing capacity and interaction scenario.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Motivation

As robots begin making their way into homes and workplaces, there is an increased interest in the interaction between humans and robots. An important part of this interaction is Physical Human-Robot Interaction (pHRI). Force data from robots can be used to improve human safety, interpret human intent, and facilitate human-guided behavior learning [5, 6]. Robotic skin with embedded pressure, temperature, and distance sensors are increasingly useful in safe and intuitive human-robot collaboration scenarios. According to Khatib et al., robots will be tasked with enhancing the human operator's abilities, providing assistance, and improving the general quality of life, while humans will be responsible for global knowledge and the correct execution of tasks [7].

To enable intuitive and comfortable physical interaction in human-robot teams there is a need to involve human intent in the robot controller design. Traditionally robot controllers have been developed with trajectory tracking as the objective. Advances like stable environmental physical interaction including force and impedance control has largely ignored human interaction. This has led to largely closed inflexible control structures that make it hard to adapt robots as co-workers and co-inhabitants in human spaces.

This motivates robust and stable pHRI systems that take into account the human dynamics and adapt to different users and different tasks. Such systems

will require estimation of human intent and will be better adapted to close physical interaction with humans [8].

This work on physical human robot interaction and human intent estimation will help overcome key challenges in personal robotics. Safe and intuitive physical interaction with robots will increase the workflow of robot-human teams in factories, workplaces, and homes. It will also make reliable personal assistants a reality. These technologies will one day enable efficient, reliable, and cheap solutions for the ageing population of the world.

1.2 Challenges

For effective human-robot interaction, specifically physical interaction there are a few challenges to be overcome. Robot dynamics: the robot is a highly nonlinear system that needs to be controlled to meet specific criteria for safety, performance, and adaptability. The dynamics of a robot changes as time passes, when picking up objects, and during physical contact. This motivates a need for controllers that can compensate for the different highly fluctuating dynamics. Robot sensors: the sensors of robotic systems need to be accurate to extract meaningful data for the controllers. However, such accurate sensors are difficult to manufacture, complex to calibrate, and hard to maintain. The sensors degrade overtime and will need recalibration, such a process is difficult for large scale arrays of sensors. Human preference: the different humans interacting with robot systems will have different physical characteristics, preferences, and intents. The robot controllers need to be aware of and adapt to these often changing parameters.

1.3 Contributions

In this dissertation, new controllers motivated by human factors studies and the human neurophysiological systems are proposed and validated. The controllers enable close human-robot interaction and co-operation using a two-loop control architecture. First, an inner-loop is employed to control the usually nonlinear dynamics of a robot. This controller is based on neuroadaptive inverse dynamics control. Then, an outer-loop adaptive controller makes the robot system combined with the human operator behave like a prescribed task model. This two controller structure decouples the design of the pHRI system. The research contributions of this dissertation are:

1. Inner-loop Cartesian space neuroadaptive controller.
    - An inner-loop robot controller that linearises the dynamics of the robot is proposed. This method uses online learning neural network function approximation techniques and is called neuroadaptive control. The advantages of the neuroadaptive controllers in joint space for pHRI was studied by conducting relevant experiments. This method was further extended to the Cartesian space control and experimentally validated.
2. Outer-loop Adaptive admittance controller using adaptive inverse control.
    - An outer-loop, task-specific controller is proposed to adapt the parameters of a prescribed robot admittance model so that the robot system assists the human to achieve task-specific objectives. A stability proof of the combined two-loop design was proposed. The controller was validated in simulation and experiments on robot hardware.
3. Calibration of tactile robot skin.
    - The proposed two-loop controller was extended to tactile robot skin. This resulted in a controller structure that intrinsically included the sensor calibration and human preference parameters into the same parametric model.

Experiments were conducted to validate the performance on a simplified tactile skin.

4. Human motion studies.
    - Understanding human motion helps design robots that can help in rehabilitation and human-robot interaction. The two-loop controller framework motivated an understanding of human motion for effective human robot collaboration. Two different techniques were employed and extended to enable better inclusion of human motion in the controller framework. Dynamic Time Warping (DTW) was used as a measure of human motion similarity, it was specifically applied to the domain of Autism characterization from motion. Dynamic Movement Primitives (DMP) was utilized for generalized human motion capture and regeneration in the controller framework.
5. Tools to test pHRI control methods - SkinSim and SkinLearn.
    - SkinSim, a new simulator framework for multi-modal, multi-resolution robot skin, aimed at solving complex design problems is proposed. Design problems for distributed arrays of skin sensors such as placement, optimizing resolution, networking, and use in feedback loops are difficult to solve, and highly dependent on the application. SkinSim is designed to answer questions related to this multi-modal design. It is implemented using the ROS and Gazebo simulation infrastructures which are supported by the Open Source Robotics Foundation (OSRF).
    - SkinLearn, is a multi-modal skin based human-robot interaction estimation, learning, and control system. The work in adaptive physical human-robot interaction, intent estimation, and tactile awareness will be implemented as libraries in SkinLearn.

The technologies developed will help the robotics community by providing open-source software tools: SkinSim and SkinLearn.

1.4 Dissertation Organization

Chapter 2 provides background and a literature survey on the subjects of this dissertation. In Chapter 3 the inner-loop neuroadaptive control structure is proposed and experimentally validated. In Chapter 4 an outer-loop adaptive admittance controller is proposed using adaptive inverse control techniques; a case study is also proposed. In Chapter 5 human motion is studied and two new methods for motion comparison and generation are proposed. Chapter 6 describes the development of SkinSim and SkinLearn, which are opensource software for simulating multi-modal robotic skin and adaptive control algorithms respectively. Finally, Chapter 7 provides a conclusion and describes planned future work.

1.5 List of Publications

1.5.1 Journal Papers

1.5.1.1 Published

1. H. Modares, I. Ranatunga, F. L. Lewis, and D. Popa, "Optimized assistive human-robot interaction using reinforcement learning," *IEEE Transactions on Cybernetics*, 2015

1.5.1.2 Under Review

1. I. Ranatunga, S. M. Tousif, D. O. Popa, and F. L. Lewis, "Adaptive Admittance Control for Human-Robot Interaction Using," *IEEE Transactions on Control Systems Technology*, vol. 00, no. 0, pp. 1–10, 2015

2. B. AlQaudi, I. Ranatunga, S. M. Tousif, F. L. Lewis, and D. O. Popa, "Model Reference Adaptive Impedance Control for Physical Human-Robot Interaction," *Autonomous Robots*, 2015

1.5.1.3 To Be Submitted

1. I. Ranatunga, I. Wijayasinghe, N. Balakrishnan, N. Bugnariu, and D. O. Popa, "Human-Robot Gesture Analysis for Diagnosis of Autism," *International Journal of Social Robotics*, 2015
2. I. Ranatunga, S. Cremer, D. O. Popa, and F. L. Lewis, "Tactile Calibration Independent Adaptive Admittance Control," *IEEE Trans. Robot.*, 2015

1.5.2 Conference Papers

1.5.2.1 Published

1. I. Ranatunga, S. Cremer, D. O. Popa, and F. L. Lewis, "Intent Aware Adaptive Admittance Control for Physical Human-Robot Interaction," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, Seattle, Washington, USA, 2015
2. G. M. Atmeh, I. Ranatunga, D. Popa, K. Subbarao, F. Lewis, and P. Rowe, "Implementation of an Adaptive, Model Free, Learning Controller on the Atlas Robot *," in *American Control Conference (ACC), 2014*, 2014
3. S. Cremer, I. Ranatunga, and D. O. Popa, "Robotic Waiter with Physical Comanipulation Capabilities," in *IEEE Int. Conf. Autom. Sci. Eng.* IEEE, 2014
4. A. Habib, I. Ranatunga, K. Shook, and D. O. Popa, "SkinSim : A Simulation Environment for Multimodal Robot Skin," in *IEEE Int. Conf. Autom. Sci. Eng.* IEEE, 2014

5. N. Bugnariu, C. Young, K. Rockenbach, R. M. Patterson, C. Garver, I. Ranatunga, M. Beltran, N. Torres-Arenas, and D. Popa, "Human-robot interaction as a tool to evaluate and quantify motor imitation behavior in children with Autism Spectrum Disorders," in *2013 International Conference on Virtual Rehabilitation (ICVR)*. IEEE, Aug. 2013, pp. 57–62
6. J. Sanford, I. Ranatunga, and D. Popa, "Physical human-robot interaction with a mobile manipulator through pressure sensitive robot skin," in *Proc. 6th Int. Conf. PErvasive Technol. Relat. to Assist. Environ. - PETRA '13*. New York, New York, USA: ACM Press, 2013, pp. 1–6
7. I. Ranatunga, M. Beltran, N. A. Torres, N. Bugnariu, R. M. Patterson, C. Garver, and D. O. Popa, "Human-Robot Upper Body Gesture Imitation Analysis for Autism Spectrum Disorders," in *International Conference on Social Robotics*, ser. Lecture Notes in Computer Science, G. Herrmann, M. Pearson, A. Lenz, P. Bremner, A. Spiers, and U. Leonards, Eds., vol. 8239, no. Cdc. Springer International Publishing, 2013, pp. 218–228
8. J. Rajruangrabin, I. Ranatunga, and D. O. Popa, "Adaptive Interface Mapping for Intuitive Teleoperation of Multi-DOF Robots," in *Advances in Autonomous Robotics*, ser. Lecture Notes in Computer Science, G. Herrmann, M. Studley, M. Pearson, A. Conn, C. Melhuish, M. Witkowski, J.-H. Kim, and P. Vadakkepat, Eds., vol. 7429. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 49–60
9. I. Ranatunga and D. O. Popa, "Collision-free trajectory generation on assistive robot Neptune," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '12*. ACM, 2012

10. I. Ranatunga, N. A. Torres, R. Patterson, N. Bugnariu, M. Stevenson, D. O. Popa, and F. Worth, "RoDiCA: a Human-Robot Interaction System for Treatment of Childhood Autism Spectrum Disorders," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '12. New York, NY, USA: ACM, 2012
11. N. A. Torres, N. Clark, I. Ranatunga, and D. Popa, "Implementation of Interactive Arm Playback Behaviors of Social Robot Zeno for Autism Spectrum Disorder Therapy," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '12. New York, NY, USA: ACM, 2012, p. 1
12. P. Kanajar, I. Ranatunga, J. Rajruangrabin, D. O. Popa, and F. Makedon, "Neptune : Assistive Robotic System for Children with Motor Impairments," in *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '11. New York, NY, USA: ACM, 2011, pp. 1–6
13. I. Ranatunga, J. Rajruangrabin, D. O. Popa, and F. Makedon, "Enhanced therapeutic interactivity using social robot Zeno," *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '11*, p. 1, 2011

1.5.2.2 Under Review

1. I. Ranatunga, S. Cremer, F. L. Lewis, and D. O. Popa, "Neuroadaptive Control for Safe Robots in Human Environments: A Case Study," in *IEEE International Conference on Automation Science and Engineering.* IEEE Press, 2015
2. I. Ranatunga, N. Balakrishnan, and D. O. Popa, "Trajectory Adaptation for Robot Differential Teaching," in *Proceedings of the 8th International Conference*

on PErvasive Technologies Related to Assistive Environments. New York, NY, USA: ACM, 2015

1.5.3 Other Contributions

1. G. M. Atmeh, I. Ranatunga, D. O. Popa, and K. Subbarao, "Indoor Quadrotor State Estimation using Visual Markers Categories and Subject Descriptors," in *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '14*. ACM, 2014
2. B. Komati, M. R. Pac, I. Ranatunga, C. Clevy, D. O. Popa, and P. Lutz, "Explicit force control vs impedance control for micromanipulation," in *Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. ASME, 2013
3. C. McMurrough, I. Ranatunga, A. Papangelis, D. O. Popa, and F. Makedon, "A development and evaluation platform for non-tactile power wheelchair controls," in *Proc. 6th Int. Conf. PErvasive Technol. Relat. to Assist. Environ. - PETRA '13*. New York, New York, USA: ACM Press, 2013, pp. 1–4
4. A. Papangelis, I. Ranatunga, D. O. Popa, and F. Makedon, "An assistive object manipulation system," in *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '13*, 2013, pp. 1–4

CHAPTER 2

Background and Literature Survey

The application of humans and robots working in collaboration has promising results in terms of efficiency and collaborative performance. Work by ABB provides evidence that a hybrid human-robot cell can improve efficiency as compared to either a human or robot working alone [33]. Erden et al. present results on impedance measurement in a collaborative human-robot welding task [34]. Recent developments in wearable robotic limbs for manufacturing assistance involves close coordination between human and robotic limbs [35].

2.1 Robot Learning Control

Motivated by the automation of industrial processes requiring repetitive predefined motions, robot controllers have by and large been designed for trajectory control [36–39]. Techniques such as Computed Torque control have been designed to achieve this objective with great success [39]. A major challenge with computed torque control is the need to have a good model of the robot dynamics. Techniques that overcome this modelling need include adaptive control, which estimates the model parameters, and robust control, which makes the controller resistant to unknown parameters [39]. Several methods have been proposed to overcome linear-in-the-parameters assumptions of adaptive controllers, including Neural Networks, Support Vector Machines, Gaussian Mixture Models and Reinforcement Learning. These learn the full nonlinear dynamics of the robot online [40–43].

The use of neural networks (NN) in feedback control systems was first proposed by [44] and [45]. Overviews of the initial work in NN control are provided by [46] and [47]. The properties of interest for trajectory tracking using NN based controllers are the tracking error and NN estimation errors. Some of the first results that included internal stability, weight bounds, tracking performance guarantees, and controller robustness was provided in 1995 [48]. Nonlinear-in-the-parameters NN were first used by F.-C. Chen [49] who used backprop with deadzone weight tuning, and Lewis [50], who used Narendra's e-mod term in backprop. Multilayer NN were rigorously used for discrete-time control by Jagannathan and Lewis [51]. Other rigorous work with stability proofs was done by Ge et al. [41], Polycarpou [52], Rovithakis and Christodoulou [53], Poznyak [54], Rovithakis [55] and others. Books on NN feedback control include [56], [40], [57], [58], Ge et al. [41, 59].

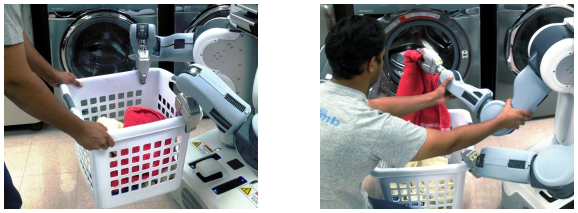
It has been observed in human factors studies that when a human learns to use a robot system, his learning has two components [60, 61]. The first component consists of learning a model to compensate for the dynamics of the robot [61]. The second component is the learning of a task-specific loop that makes the combined human-robot model successfully execute the task. This dual approach has been used for adaptive impedance control of human-robot interactions in task performance by Suzuki and Furuta [61]. These studies support the idea that a trajectory following controller should not be closed around the robot alone. Instead, all task-specific objectives should be closed in an outer task-specific loop that also includes the human transfer characteristics.

2.2 Physical Human-Robot Interaction

For robots to work in collaboration with humans good physical Human-Robot Interaction (pHRI) is vital. According to De Santis et al. [4] safe and dependable pHRI systems should be developed before introducing robots into human environments.

There are many challenges to moving a robot and accomplishing tasks in environments that are inherently cluttered and dynamic. Static obstacles such as furniture and dynamic obstacles such as humans have to be detected and avoided. Dynamic situations require fast and real-time reaction to avoid damage, variations in pose requires good situational awareness and maneuvering skills. Other issues include specialized tools, heterogeneous architectural structures like doors and stairs, deformable objects like curtains to manipulate, great variation in the objects to correctly identify, and noisy imprecise sensors [62].

Current research in robotics has focused on the extensive use of vision, planning, and optimization to solve these problems [63]. Tasks are performed by systems that are increasingly complex and depend on vision including 3D sensors [64]. Although these techniques are powerful, they are not mature enough to deploy in a home environment where reliability and repeatability are important [65]. The challenge with using vision is the time required to process and plan, occlusions, and unsuitable lighting conditions.



(a) Comanipulation. (b) Teaching.

Figure 2.1. pHRI for everyday tasks with PR2.

2.2.1 Force Control

Direct and indirect force control has been extensively studied in terms of performance and stability [66]. Impedance control pioneered by Hogan [67] is the most popular form of indirect force control. It allows stable contact by the end effector of the robot and overcomes many instability issues associated with explicit force control. However, impedance control requires identifying the robot model as well as the environmental contact dynamics [68,69]. Recent work by Ficuciello et al. [70] exploits redundancy to improve the Cartesian space inertia decoupling of a 7 DOF robot arm. Admittance control, the dual of impedance control, has also been applied in various robotic applications [71].

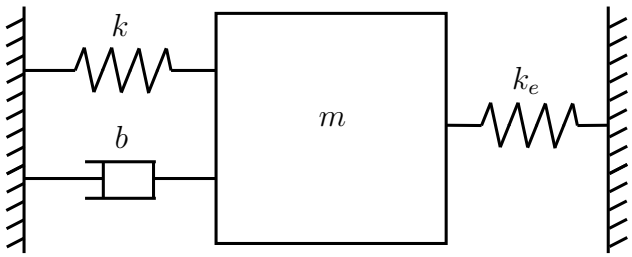


Figure 2.2. Simple robot model and force interaction with environment.

Given a simple system model as in Figure 2.2

$$m\ddot{x} + b\dot{x} + kx = u - f_e \tag{2.1}$$

where, m is the mass, b is the damping, k is the spring constant, f_e is the external force, and u is the control input.

2.2.1.1 Explicit Force Control

Involves controlling the contact forces directly using feedback control. In classical explicit force control schemes the object is to maintain a desired fixed force f_d, the controller u is

$$u = K_{pf}(f_d - f_e) + K_{vf}(\dot{f}_d - \dot{f}_e) + K_{if}\int(f_d - f_e) \tag{2.2}$$

This does not work in all interaction situations because the interaction moves from contact to single/multi contact scenarios sporadically in real situations. These controllers can become unstable due to changes in contact environment [39,68,72,73].

2.2.1.2 Hybrid Force/Position Control

Craig [74] extended the explicit force control technique to control force in the constrained directions and positions in the unconstrained directions, this enabled them to move along a known geometry while simultaneously maintaining desired force in the constrained directions.

2.2.1.3 Impedance Control

In impedance control schemes the robot is given a desired trajectory and the deviation from this prescribed trajectory reference q_d or the error dynamics is made to follow a pre set model.

$$u = (m - m_d)\ddot{x} + (b - b_d)\dot{x} + (k - k_d)x + m_d\ddot{x}_v + b_d\dot{x}_v + k_d x_v \tag{2.3}$$

where m_d, b_d, k_d are the desired interaction mass, damping and spring constant. The impedance control method requires a desired trajectory to track or a virtual trajectory $\ddot{x}_v$, $\dot{x}_v$, x_v. Hybrid force control is similar but contact and non contact scenarios are handled using a hybrid force position scheme.

2.2.1.4 Admittance Control

Recent work [71,75,76] has used admittance control which is the dual of impedance control. Human forces were used to generate robot motions. This work used admittance control to allow the robot to perform a trajectory control objective. Recently admittance control has been used in different forms in haptic devices [77] and exoskeletons [78] to allow compliant human-robot interaction without a trajectory following objective.

$$f_e = m_m(\ddot{x}_m - \ddot{x}) + b_m(\dot{x}_m - \dot{x}) + k_m(x_m - x) \tag{2.4}$$

where m_m, b_m, k_m are the desired model mass, damping and spring constant. The admittance model is used to generate the desired model trajectories $\ddot{x}_m$, $\dot{x}_m$, and x_m given a force input f_e.

The objective is to generate end effector motion from sensed forces. An admittance model is usually represented as a transfer function [71, 79]. Some authors have called this an impedance filter [75]. Trajectories generated by the admittance model are sent to a position tracking inner-loop [71, 75]. Marayong et al. [77] used admittance control in a haptic device.

2.2.2 Adaptive physical Human-Robot Interaction

Adaptive impedance control was studied by [68, 80–84]. The work on adaptive impedance and admittance can be divided into two types: environment parameter estimation to increase the performance of the impedance controller, and adaptation of the desired mass-spring-damper coefficients to improve interaction with a human. Human intent has been used to adapt the impedance parameters of robot systems [85]. For instance, Dimeas et al. [86] implements a Fuzzy logic controller for admittance model adaptation considering human intent. Their method involves offline tuning of

the parameters. Lecours et al. [87] reports on performance tuning of an assist device based on separation of the position and velocity space of the robot.

2.2.3 Safety

Safety in an industrial context was explored by Haddadin et al. in several studies which measured and characterized the results of robot collisions with humans [88–91]. In their work, novel hardware capable of accurate joint torque sensing was developed to reduce impact forces.

Interest in safe pHRI has motivated hardware modifications including gravity compensation using a counter balance such as the PR2 [92]. Recent developments in hardware based safety features include accurate joint torque sensing, gravity compensations, reduced robot weight, and joint torque limiting. Although these methods have reduced the risk of high energy collisions, they have also significantly reduced the speed and accuracy of robot manipulators.

In household robot applications the robots are generally compliant and safe with changing often imprecise dynamic models. The tasks involved also change from pick and place tasks requiring accurate trajectory tracking, to physical interaction with humans, as depicted in Fig. 2.1. Model-based computed torque controllers can achieve good results but they depend on known dynamic models. Even if a robot model is known, the presence of a payload comparable with the mass of a lightweight robot will alter the performance of such compensation schemes. Furthermore, nonlinearities due to the inherent flexibility of lightweight transmission systems will also increase the uncertainty of such models.

2.2.4 Robot Skin

Robotics research has developed at a fast pace in the past few decades. It has matured to a point where real world applications have started to emerge. A major milestone in applications oriented robotics was achieved by the DARPA robotics challenge. Robots were developed to assist as teleoperated units in various disaster rescue scenarios.

This type of work involved some major challenges in robotics, some of which included dynamics and constraint aware control and planning, contact forces and torque estimation, dynamic balancing of humanoid robots, etc. The majority of the tasks involved multiple heterogeneous contact scenarios. The ability of the robot to correctly sense these contact scenarios and adjust its behaviour appropriately was crucial.

Research has been conducted into various force sensing devices that are relevant to robotics. The force/torque sensor or load cell is the most commonly used device in robotics force control. Tactile sensors have also been developed which includes technologies like capacitive, resistive, optical, etc. Most currently available tactile sensors are in discrete form consisting of individual sensing elements. Major research projects Roboskin and CellularSkin have developed arrays of tactile elements. These are still at the research level.

Dahiya [1] outlines the need for flexible, stretchable electronic systems to realize multi-functional electronic skin as seen in Figure 2.3, using the ability to fabricate single-crystal nanowires 4-50um in width on ultra-thin (10um) flexible polyimide. Further work on ultra-thin substrates was demonstrated in [93] where hybrid flexible tactile sensors based on low temperature polycrystalline silicon thin film transistors were fabricated.

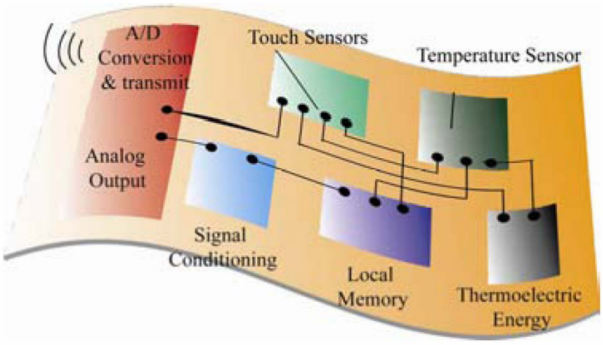
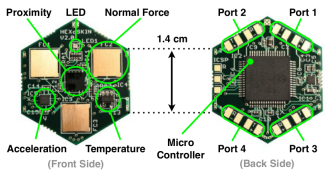


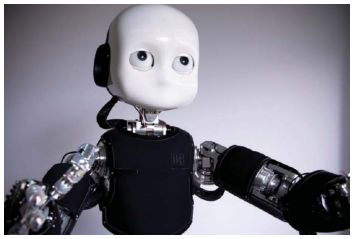
Figure 2.3. The electronic skin concept with multiple functionalities integrated over ultra-thin flexible substrates like polyimide [1].

CellulARSkin is a multi-modal, self-organizing electronic skin [94]. The modular units of the skin perform temperature, acceleration, proximity and normal force sensing (Figure 2.2.4). Each cell has the ability to explore connections and self-organize which, when combined with the accelerometers, allow the units to determine topology and their position on the robot. RoboSKIN (Figure 2.2.4) is another advanced skin with the ability to communicate between adjacent units. In [95] RoboSKIN is integrated into the robot iCub along with a method to compensate for temperature drift of the capacitive tactile sensors (Figure 2.2.4).



(a) Sensor electronics of the CellulARSkin unit cell [94].

(b) RoboSKIN patch: each triangle hosts 10 sensors [95].

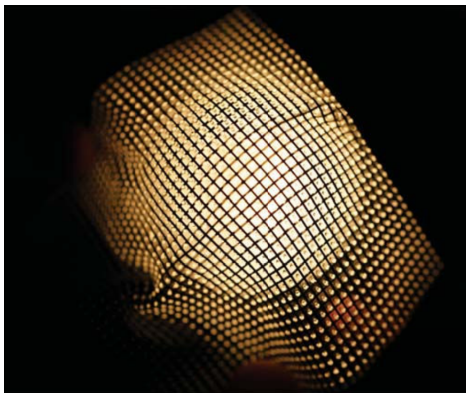


(c) iCub robot with RoboSKIN patches on arms and torso [95].

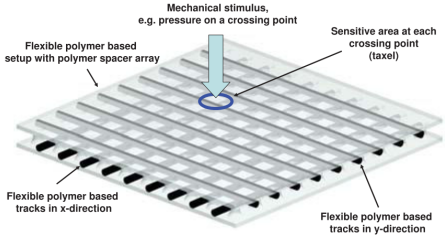
Figure 2.4. Robot Skin.

A discussion of robot skin sensitivity can be found in [96], where rubber skin is added to an optical three-axis tactile sensor. The rubber, initially implemented to protect the sensor surface from large applied forces, has the additional effect of increasing the sensing surface and resolution. In addition to this, [97] demonstrates hyperacuity in the RoboSKIN. When the sensor encounters an object multiple taxels are activated simultaneously resulting in a ten-fold improvement of position acuity over taxel resolution.

An important consideration for robot skin in unstructured environments is the ability to withstand unintended collisions. The DLR Artificial Skin (Figure 2.5) [2] is an all-polymer approach to tactile sensing. Orthogonally oriented polymer circuit tracks provide sensitivity to light touch while also being able to withstand indentation forces up to 50N. Additional soft sensing systems in [98] include elastomers containing embedded micro-channels filled with liquid metal are used as tactile sensors on complex shapes. These sensors are designed to limit the physical impact on the host system, human or robot, while providing necessary pressure and/or strain information.



(a) DLR Artificial Skin wrapped around a light bulb.



(b) Sensing principle.

Figure 2.5. DLR Artificial Skin [2].

2.2.5 Human Sensorimotor Ranges

From robotics/kinesiology literature, we learn that realistic Human-Robot Interaction and haptics requires a fast response depending on sensory modality including [99]:

- Vision - 30 Hz refresh rate
- Force (texture) - 1 KHz bandwidth
- Force (pushing) - bandwidth
- Heat - 10 Hz
- Human motion - 10 Hz
- Displacement - 1 mm

Furthermore, in the human nervous system, interaction requires minimal time delays, contextual understanding (at O(Hz) bandwidths), robust classification and detection (at O(10 Hz) bandwidths), learning and adaptation (at O(10 Hz) bandwidths), and motor control (at O(100 Hz) bandwidths.

Human-inspired sensors, include touch/temperature/vibration sensors on the body, while kinesiology studies have measured human motion performance using typical motion detection hardware such as: camera Arrays and Markers (these are expensive and somewhat intrusive), (there are magnetic sensors - susceptible to noise and intrusive), and Inertial sensors (these need sensor fusion and are intrusive).

Typical human motions experience the following dynamic characteristics [100, 101]:

- Average fast body motion linear speed is at 7-8 m/s
- Average fast body angular velocity is at 1.5-2pi rad/s
- Average cycle for head motion is 1 Hz
- Motion bandwidth < 10 Hz
- Control loop closed at

2.2.6 Social Robots

Autism Spectrum Disorder (ASD) is a developmental disorder characterized by deficiencies in social interaction, speech, cognition, motor coordination, and imitation [102]. According to the Autism and Developmental Disabilities Monitoring (ADDM) Network of the Centers for Disease Control and Prevention (CDC) an average of 1 in 88 children in the US is diagnosed with an ASD [103]. Although the cognitive capacity of individuals in the Autism spectrum vary greatly, most of the individuals have sensorimotor abnormalities. Although Autism was recognized as early as 1943 by Kanner and accepted to have a biomedical origin by the 1980s, there is a lack of quantitative diagnosis tools. Currently ASD diagnosis mainly focuses on qualitative behavior observation which results in imprecise and sometimes arbitrary categorization of individuals in the Autism spectrum [102, 104].

Robotic systems have been developed for use in the therapy of individuals in the autism spectrum such as FACE, AuRoRa, Kaspar, Nao, and Keepon [104, 105], but many of them do not engage in dynamic gestural interaction in a truly autonomous, interactive manner. Studies show that the appearance of the robot plays an important role in how children relate to and interact with such robots [106–108] and suggest that imitation and turn-taking are types of interactions useful in motivating and engaging children with ASD. In these types of projects, the interaction capacity of the robot is restricted due to lack of objective criteria to rate imitative gestural Human-Robot Interaction (HRI). A robot called Bandit was used to guide older adults to perform imitative exercises [109]. This project had a robot perform upper body gestures that the subject imitated, performance criteria were related to achievement of target poses. Projects involving robots interacting with individuals, specifically children with ASD tend to be interdisciplinary projects that have a small number of participants, which makes a good clinical conclusion difficult. These projects and others involving the

use of robots for assisting humans in a social, collaborative setting can be considered part of a relatively new field called Socially Assistive Robotics (SAR) [104].

2.2.6.1 Philip K Dick Android

The Philip K Dick (PKD) android was developed by David Hanson as a copy of the science fiction writer Philip K Dick. This robot is capable of high fidelity facial expressions. It is related to many such androids built by David Hanson including FACE.

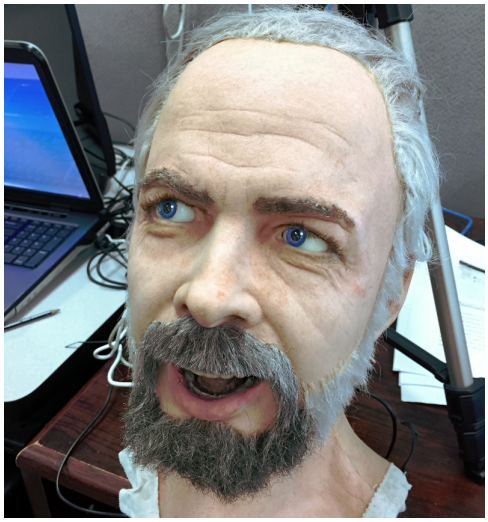


Figure 2.6. PKD Android.

2.2.6.2 Zeno

Zeno is a child-size humanoid robot by RoboKind. The appearance of Zeno is based on a fictitious character - it looks like a 4-7 year old child, and its head is about 1/4 of a size of an adult human head. Its unique features include life-like skin made of Frubber material, which is the intellectual property of Hanson Robotics. The appearance of Zeno is a game changing experience thanks to this material, and to

Figure 2.7. Zeno.

the robot aesthetics. The head of Zeno is powered by 3 Cirrus CS-101 STD 4g Micro Servos, 5 Hitec HS-65MG Mighty Metal Gear Feather Micro Servos, 1 Dynamixel AX-12+ Robot Actuator from Robotis, has 3 degrees of freedom at the neck joint, and it is capable of panning, tilting the head back and forth as well as left to right. It also has 2 degrees of freedom in each eye (pan and tilt), and 4 of the servos are used for generating facial expressions (eye blink, jaw motion for smile, eyebrow motion for frown, etc.). The visual feedback is obtained from a Mini Color Snake Pinhole Camera in the left eye.

2.2.7 Dynamic Time Warping

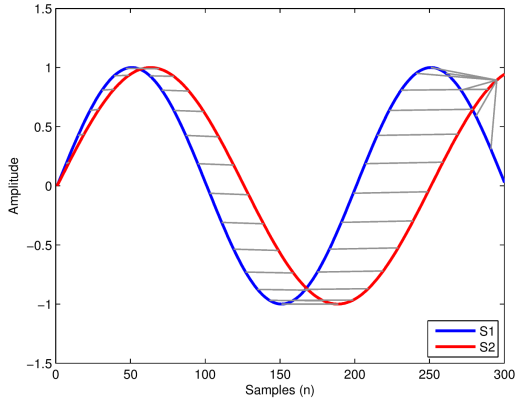


Figure 2.8. Example DTW match between signals S_1 and S_2.

Dynamic Time Warping (DTW) is an established signal processing method that offers a distance measure between signals similar to the Euclidean distance. It is used for measuring similarity between two sequences which may vary in time or speed [110]. Time-warping is applied to signals to align them "optimally", prior to measuring the distance. Optimal alignment in this context, is the alignment of the signal time samples that makes the total distance between the signals as small as possible. This alignment induces a non-linear mapping between the two signals, e.g. warping of the signals. A good description of the DTW algorithm is given by Keogh et al. [111]. It has been used for Automatic Speech Recognition (ASR). Optimizations to the DTW algorithm are outlined in Sakoe and Chiba [110].

Fig. 2.8 shows a typical result when using the DTW algorithm from 1, the gray lines depict the nonlinear map between the signals. It can be seen from the right side of Fig. 2.8, where many points from S_1 are mapped to a single point of S_2, this is because DTW matches every point of the signals together.

2.2.8 Dynamic Movement Primitives

Generating human like motion and the metrics associated with it have been studied by many researchers [112–114]. Gaussian and Bernoulli mixture models have been used to generalize joint angle movements and to encode human motion in robots [115, 116]. Ijspeert et al. introduced a general method to encode human motion by treating them as nonlinear dynamic systems [117]. This was called Dynamic Movement Primitives (DMP) and was demonstrated and extended to higher dimensions and multiple robots [118,119]. DMPs can encode highly nonlinear motion and generalize them. A motion can be modified by changing the time, beginning and end points of the trajectory, and the amplitude. DMPs are also robust against perturbations [119] see Figure 2.9.

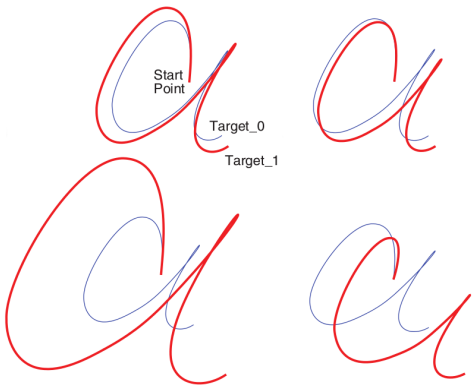


Figure 2.9. DMP Example [3] .

CHAPTER 3

Neuroadaptive Inverse Dynamic Control

The objective of this chapter is to extend and study the behavior of Neuroadaptive controllers originally developed by Lewis et. al [40, 48] applied on a real PR2 robot. This work demonstrates the effectiveness of the Neuroadaptive controllers in joint trajectory following tasks and extends the concepts to Cartesian trajectories.

This chapter is organized as follows: Section 3.1 outlines the joint space controller formulation and results from simulations and real experiments on a PR2 robot are presented. Then Section 3.2 outlines the Cartesian space controller formulation and results from simulations and real experiments on a PR2 robot are presented. Finally, Section 3.3 provides a summary of the chapter.

3.1 Joint Space Controller Formulation

In this section the original Neuroadaptive controller formulation developed by Lewis et. al [40, 48] is briefly described. The controller formulation used in this chapter is in joint space assuming desired joint space trajectories.

The general robot dynamic equation with actuator dynamics is [39]

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = \tau \tag{3.1}$$

where n is the number of DOF of the robot, $q \in \mathbb{R}^n$ are the joint positions, $M(q)$ is the inertia matrix, $V(q,\dot{q})$ is the Coriolis/centripetal vector, $G(q)$ is the gravity vector, and $F(\dot{q})$ is the friction term. The disturbance torque is $\tau_d \in \mathbb{R}^n$ and $\tau \in \mathbb{R}^n$ is the control torque. The robot dynamics (3.1) is formulated in joint space. The

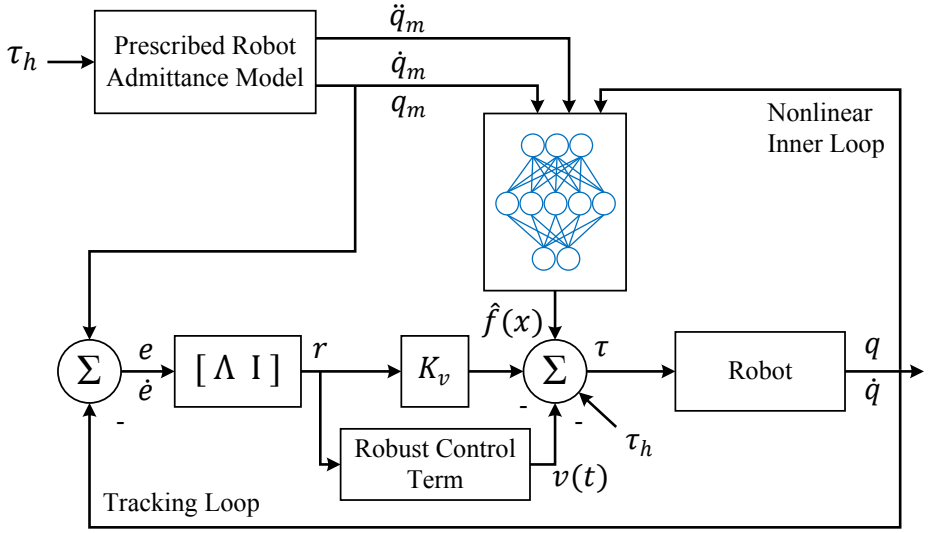


Figure 3.1. Model Reference Neuroadaptive Controller.

following derivation also holds in Cartesian space form with standard modifications involving the Jacobian [39].

Define the trajectory-following error as

$$e = q_r - q \tag{3.2}$$

and the sliding mode error is

$$r = \dot{e} + \Lambda e \tag{3.3}$$

where Λ is a positive definite design parameter matrix.

Using (3.1), (3.2) and (3.3) yields the robot model following error dynamics

$$M(q)\dot{r} = -V(q,\dot{q})r + f(x) + \tau_d - \tau - \tau_h \tag{3.4}$$

where

$$f(x) = M(q)(\ddot{q}_m + \Lambda\dot{e}) + V(q,\dot{q})(\dot{q}_m + \Lambda e) + F(\dot{q}) + G(q) \tag{3.5}$$

is a nonlinear function of unmodeled robot parameters.

Let an approximation based controller be

$$\tau = \hat{f}(x) + K_v r - v(t) \tag{3.6}$$

where $\hat{f}(x)$ is the approximation of the robot function $f(x)$ in (3.5), $K_v r$ is the gain of the outer PD tracking loop, $K_v = K_v^T > 0$ is a diagonal outer-loop gain matrix, and $v(t)$ is a robustifying signal that compensates for unmodelled and unstructured disturbances.

Putting (3.6) in (3.4) and simplifying yields the closed loop error dynamics

$$M(q)\dot{r} = -V(q,\dot{q})r - K_v r + \tilde{f}(x) + \tau_d + v(t) \tag{3.7}$$

where $\tilde{f}(x) = f(x) - \hat{f}(x)$ is the function approximation error.

The learning loop performance and stability proof is based on prior works [40, 41, 48].

The nonlinear function $f(x)$ in (3.5) is unknown. This function can be approximated by a neural network

$$f(x) = W^T \sigma(V^T x) + \varepsilon \tag{3.8}$$

where W and V are ideal unknown weights and $\sigma(.)$ is the activation function. Let the neural network approximation property given by (3.8) hold for the function $f(x)$, specified by (3.5) with a given accuracy $\|\varepsilon\| \leq \varepsilon_N$ on a compact set [40, 41].

The ideal weights for the NN are unknown, therefore a weight tuning algorithm is used to update the approximate NN weights $\hat{W}$ and $\hat{V}$. The input to the NN is $x = \left[\ e^T\ \dot{e}^T\ q_r^T\ \dot{q}_r^T\ \ddot{q}_r^T\ \right]^T$. Then the control input is

$$\tau = \hat{W}^T \sigma(\hat{V}^T x) + K_v r - v \tag{3.9}$$

The robustifying signal $v(t)$ is

$$v(t) = -K_z(\|\hat{Z}\|_F + Z_B)r \tag{3.10}$$

where K_z is the gain of the robustifying term

$$\hat{Z} = \begin{bmatrix} \hat{W} & 0 \\ 0 & \hat{V} \end{bmatrix}$$

$\|.\|_F$ is the Frobenius norm, and Z_B is a bound on the NN weights.

The following are the NN weight update equations

$$\dot{\hat{W}} = F\hat{\sigma}r^T - F\hat{\sigma}'\hat{V}^T x r^T - \kappa F\|r\|\hat{W} \tag{3.11}$$

$$\dot{\hat{V}} = Gx(\hat{\sigma}'^T\hat{W}r)^T - \kappa G\|r\|\hat{V} \tag{3.12}$$

$$\hat{\sigma}' = diag\left\{\sigma(\hat{V}^T x)\right\}\left[I - diag\left\{\sigma(\hat{V}^T x)\right\}\right] \tag{3.13}$$

where F and G are positive definite matrices, and $\kappa > 0$ is a small design parameter. (3.13) assumes that $\sigma(.)$ is a sigmoid activation function.

3.1.1 Joint Space Neuroadaptive Controller Proof

Assumption 1

The model trajectory is bounded with q_B a known scalar bound as follows

$$\left\| \begin{array}{c} q_m(t) \\ \dot{q}_m(t) \\ \ddot{q}_m(t) \end{array} \right\| \leq q_B \tag{3.14}$$

Assumption 2

The approximation (3.8) holds with a given accuracy $\|\varepsilon\| \leq \varepsilon_N$ on a compact set [40, 48].

Assumption 3

Assume that Z_B is a scalar bound on the NN weights such that $\|\hat{Z}\|_F < Z_B$.

Proof

Define the weight estimation errors as $\tilde{V} = V - \hat{V}$, $\tilde{W} = W - \hat{W}$, and $\tilde{Z} = Z - \hat{Z}$.

Define the Lyapunov function for the inner-loop as

$$L_1[r, \tilde{W}, \tilde{V}] = \frac{1}{2} r^T M(q) r + \frac{1}{2} tr\{\tilde{W}^T F^{-1} \tilde{W}\} + \frac{1}{2} tr\{\tilde{V}^T G^{-1} \tilde{V}\} \tag{3.15}$$

Differentiating this results in

$$\dot{L}_1 = r^T M(q) \dot{r} + \frac{1}{2} r^T \dot{M}(q) r + \frac{1}{2} tr\{\tilde{W}^T F^{-1} \dot{\tilde{W}}\} + \frac{1}{2} tr\{\tilde{V}^T G^{-1} \dot{\tilde{V}}\} \tag{3.16}$$

substituting from (3.7) and (3.8)

$$\begin{aligned} \dot{L}_1 = -r^T K_v r + \frac{1}{2} r^T (\dot{M} - 2V) r + tr\{\tilde{W}^T (F^{-1} \dot{\tilde{W}} + \hat{\sigma} r^T)\} \\ + tr\{\tilde{V}^T (G^{-1} \dot{\tilde{V}} + \varphi r^T \hat{W}^T \hat{\sigma}')\} \end{aligned} \tag{3.17}$$

as shown in [40] this results in

$$\dot{L}_1 = -r^T K_v r \tag{3.18}$$

and as a result $L_1[r, \tilde{W}, \tilde{V}] > 0$ and $\dot{L}_1 \leq 0$.

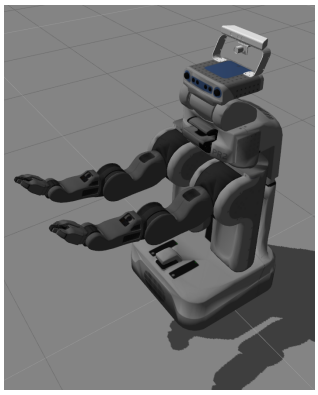
Define $S_r \equiv \{r \mid \|r\| < \frac{b_x - q_B}{c_0 + c_2}\}$, where c_0, c_2 are computable positive constants. If $r(0) \in S_r$, then the approximation property holds. More details on Uniform Ultimate Bounds (UUB) of both $\|r\|$ and $\|\hat{Z}\|_F$ so that the approximation property holds throughout, can be found in [40, 41].

3.1.2 Simulation Results

In this section results from simulations on a PR2 robot in the Gazebo simulation environment is presented. The PR2 is a dual arm robot originally developed at Stanford [92], see Figure 3.2. It was fully developed and commercialized by Willow garage. The 7 DOF right arm of this robot was used to conduct the simulations. The

controller was implemented in using the real-time controller manager framework of the PR2 in ROS Groovy.

(a) Real robot.



(b) Simulated robot in Gazebo.

Figure 3.2. PR2 service robot.

Gazebo is a robot simulation software that was originally developed in the Player/Stage project. It is designed to be highly modular and supports ODE [120], Bullet [121], Simbody [122], and DART [123] physics engines.

The controller parameters used were $K_v = 10I_7$, $\Lambda = 0.5I_7$, $F = 100I_7$, $G = 20I_7$, $\kappa = 0.07$, $K_z = 0.001$, and $K_b = 100$, where I_7 is the 7×7 identity matrix. A two-layer Neural Network with 36 inputs including the bias input, 10 hidden layer neurons, and 7 outputs was used. The sigmoid function $\sigma(x)$ was used for the activation functions. The weights $\hat{W}$ and $\hat{V}$ of the network were initialized to zero.

Table 3.1. Table of Neuroadaptive controller performance on the PR2 in Gazebo for $\kappa = 0.07$, $K_v = 3$, $\lambda = 9$, $Z_b = 1000$, $F = 100$, and $G = 20$. e_i and τ_i are the position tracking error and the control torque for joint i

NN	K_z	e_1	e_3	τ_1	τ_3
Off	1	3.1637	16.6189	90.3642	264.6455
On	1	0.2076	7.4942	253.4427	247.5655
On	0	0.2106	7.4947	254.6869	247.7194

Table 3.1 shows the performance of the neuroadaptive controller in Gazebo. The low joint error values e_1 and e_3 show a lower joint trajectory error using the neuroadaptive controller. The gains in the outer PD loop was held constant throughout the experiment.

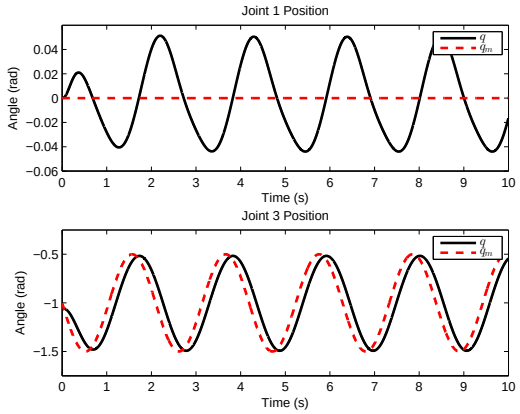


Figure 3.3. Joint Position Tracking without Neuroadaptive control.

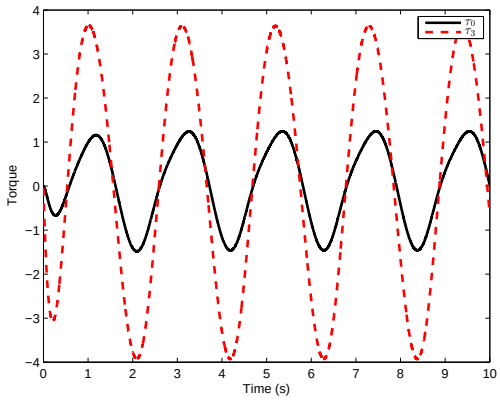


Figure 3.4. Control torques without Neuroadaptive control.

Fig. 3.3 shows trajectory tracking with a PD controller; the neural network component is turned off. The effect of dynamic coupling can be seen in the oscillation in joint 1. The control torques seen in Fig. 3.4 show the low control torques for joint 1, to correct this the PD controller can be tuned manually.

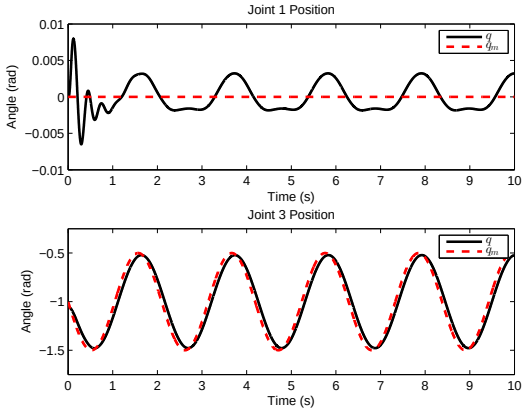


Figure 3.5. Position Tracking with Neuroadaptive control.

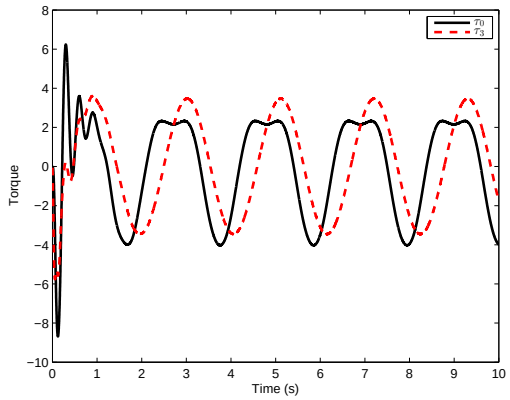
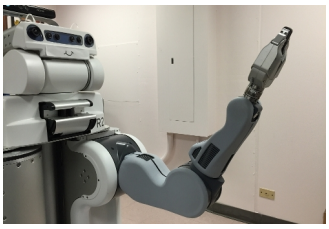


Figure 3.6. Control torques with Neuroadaptive control.

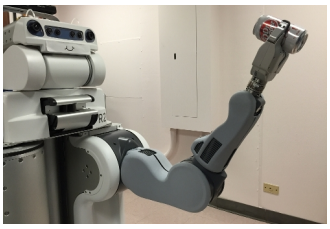
The Neuroadaptive controller decouples the joints by cancelling the nonlinear effects, this is seen in Fig. 3.5 the position error is an order of magnitude smaller than in Fig. 3.3. The torques in Fig. 3.6 are around the same magnitude as in Fig. 3.4, but the torque of joint 1 is now higher to compensate for the nonlinear dynamics. Figs 3.3 and 3.5 show the effect of using the Neuroadaptive controller.

3.1.3 Experimental Results

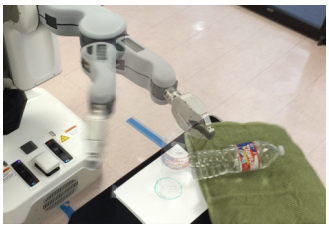
The controller parameters used were $K_v = 5I_7$, $\Lambda = 5I_7$, $F = 100I_7$, $G = 50I_7$, $\kappa = 0.07$, $K_z = 0.001$, and $Z_b = 100$, where I_7 is the 7×7 identity matrix. A two-layer Neural Network with 36 inputs including the bias input, 10 hidden layer neurons, and 7 outputs was used. The sigmoid function $\sigma(x)$ was used for the activation functions. The weights $\hat{W}$ and $\hat{V}$ of the network were initialized to small random values.



(a) No payload



(b) With payload



(c) Collision

Figure 3.7. Experimental setups for comparing NN and PID controllers.

Three different types of experiments were conducted to demonstrate the effectiveness of the neuroadaptive controller as compared to the standard robot PID joint controllers (Fig. 3.7). The experiments conducted were:

A. Free space motion without payload, in which the arm is following a desired joint trajectory.

B. Free space motion with payload, in which the end-effector is carrying an object of unknown mass.

C. Collision experiments, in which an unknown obstacle is encountered during robot motion.

In all experiments a sinusoidal trajectory was applied to joint 3, i.e. the elbow joint of the PR2 arm. In the first two experiments, the amplitude of the joint motion was set to 0.5 radians. Joints 0 through 6 were positioned at $q = [0, 0, 0, 1.0, 0, 0, 0]^\intercal$, corresponding to the shoulder pan, shoulder lift, upper arm roll, elbow flex, forearm roll, wrist flex, and wrist roll joints.

Five different rates or angular frequencies were tested in experiment *A*. In experiment *B*, a soda can weighing 355 grams was used as a payload. The highest rate was not tested because the motion became too distorted due to torque saturation and the desired reference trajectory could no longer be followed. In experiment *C*, the amplitude was changed to 0.75 radians and the arm joints were set to

$q = [0, 0, 1.57, 0.75, 0, 0, 0]^\intercal$. A 1 liter water bottle was placed in the path of the gripper directly in front of the PR2 (Fig. 3.7c). Ten collision were performed with each controller by executing the sinusoidal trajectory with a rate of 3 radians per second.

3.1.3.1 Free space motion

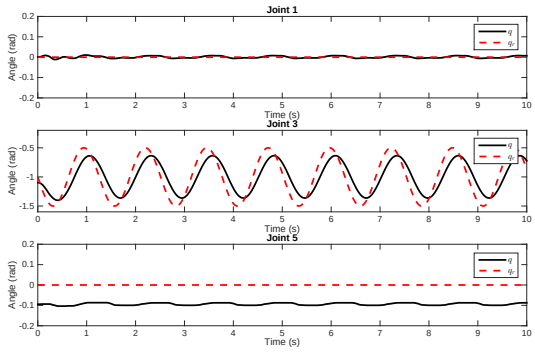
The experiments without a payload was carried out to compare the performance of the controllers in free space. To quantify the joint tracking performance, the error (3.2) at each time step $\Delta t = 0.001$sec was computed over a period of 10 seconds. The 2-norm was computed for each joint and then summed:

$$\sum_{i=0}^{6} \|q_r - q\|_2 \tag{3.19}$$

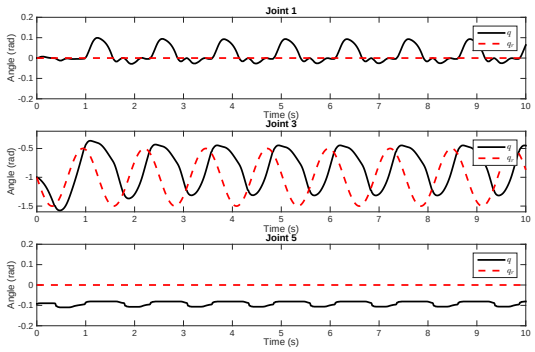
where i is the joint number. Similarly, the norm of the torques was computed for each joint and then combined into a torque performance value:

$$\sum_{i=0}^{6} \|\tau\|_2 \tag{3.20}$$

Figs. 3.8a and 3.8b show the executed trajectory q and desired reference trajectory q_r for joints 1, 3, and 5, which correspond to the shoulder lift, elbow flex, and wrist flex joints. As expected, the shoulder pan and the roll joints displayed little to no error and are therefore not depicted. At 5 radians per second both controllers have degraded performance, with the PID controller performing worse. Fig. 3.12a shows the the total error of the two controllers at five different rates. The tracking error is initially lower for the PID controller, but increases more dramatically with the joint velocities. The joint tracking is more consistent for the neuroadaptive controller because it can compensate for changes in the robot dynamics. The standard PID controller can only be tuned for a limited range of joint velocities and fails when those are exceeded.

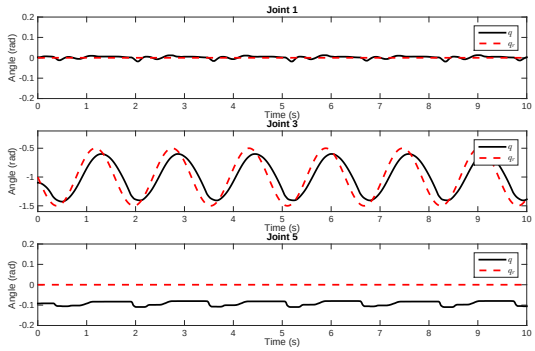


(a) Neuroadaptive

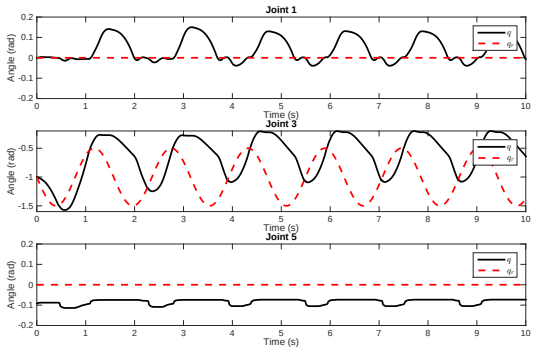


(b) PID

Figure 3.8. Joint tracking performance without a payload at 5 rad/s.



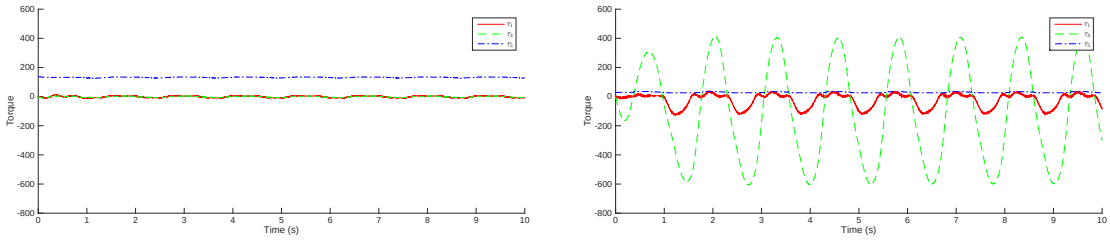
(a) Neuroadaptive



(b) PID

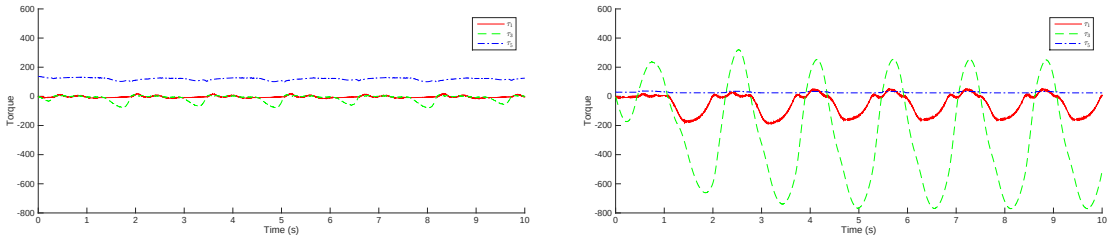
Figure 3.9. Joint tracking performance with a 355g payload at 4 rad/s.

The neuroadpative controller also performs better from a safety point of view when considering the lower torque values clearly shown in Figs. 3.10a and 3.10b. Interestingly, joint 5 (wrist flex) has the largest control effort while the PID controller generates the highest torques for joint 3 (elbow flex). The total joint torque is depicted in Fig. 3.13a. At 1 radians per second the performance is comparable. As the joint velocities increase, the PID control torque increases rapidly and diverges faster than the neuroadaptive controller.



(a) Neuroadaptive (b) PID

Figure 3.10. Control torque without payload at 5 rad/s.



(a) Neuroadaptive (b) PID

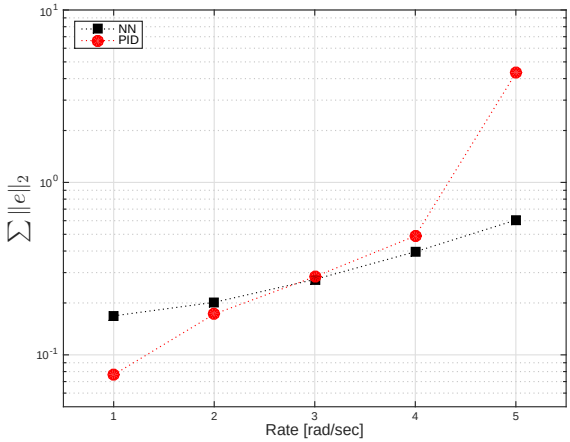
Figure 3.11. Control torque with a 355g payload at 4 rad/s.

3.1.3.2 Free space motion with Payload

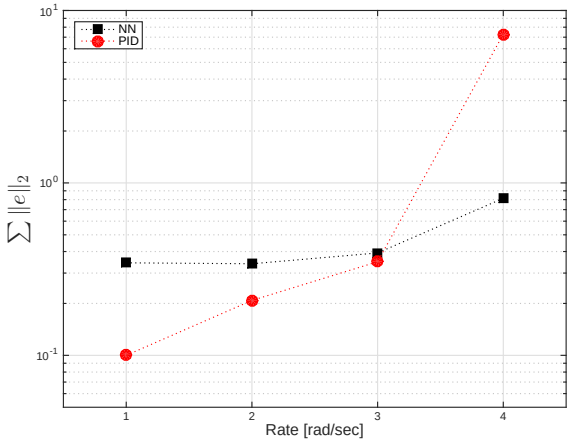
The added payload changes the manipulator dynamics by increasing the end-effector inertia. In traditional computed torque or inverse dynamics controllers, this change in manipulator dynamics has to be detected and explicitly added to the controller. Since the weight of the object is unknown to the robot in this experiment, the tracking performance is worse even at a lower rates.

Similarly to the previous section, the joint tracking performance of the neuroadaptive controller outperforms the PID controller at the highest tested rate as shown in Figs. 3.9a and 3.9b. In addition, the control torques generated by the PID controller are much higher than those generated by the neuroadaptive controller (Figs. 3.11a and 3.11b). However, at lower rates the difference is less profound as depicted in Figs. 3.12b and 3.13b.

The results for Sections 3.1.3.1 and 3.1.3.2 are summarized in Table 3.2.



(a) No payload



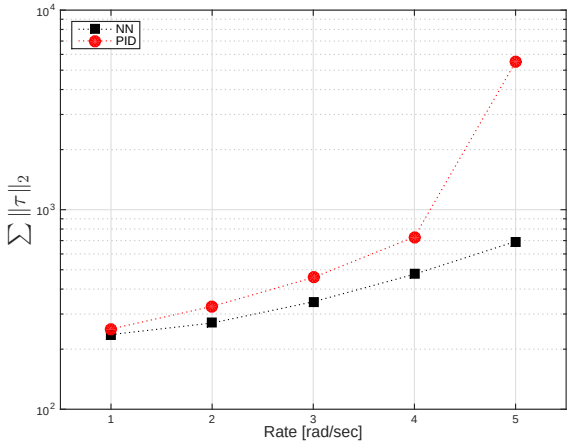
(b) With payload

Figure 3.12. Total joint tracking error for movement frequencies (log scale).

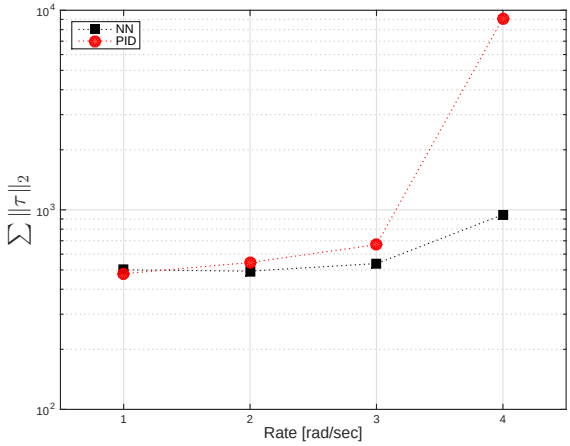
3.1.3.3 Results with Collision

In this section the contact forces experienced at the end effector of the robot during collisions is presented. The force tangential to the circular motion of the elbow flex joint (joint 3) was measured during a 2.5 second time interval. Typical results for the neuroadaptive and PID controller are shown in Fig. 3.14.

The results of running ten trials are represented in Table 3.3. The maximum force $\|F\|_\infty$ and impulse $\|F\|_1 \Delta t$ was computed for each trial. The neuroadaptive controller produces an average maximum contact force of 4.85N, which is lower than



(a) No payload



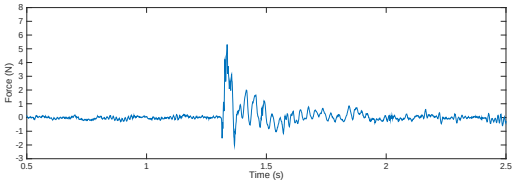
(b) With payload

Figure 3.13. Total control torque for movement frequencies (log scale).

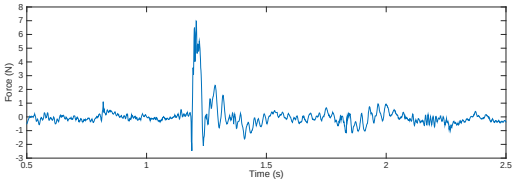
the PID value of 6.99N. Furthermore, the average impulse is lower at 0.56Ns compared to 0.89Ns. Hence, the neuroadaptive controller results in lower collision energies and could therefore be considered a safer alternative to the standard PID controller. This is especially important in human environments, which are highly dynamic and unpredictable. In physical HRI interaction scenarios, the controller has to be accurate and responsive, while minimizing the risk of human injury.

Table 3.2. Neuroadaptive (NA) and PID controller joint error and torque performance results for experiment A and B

Rate	Payload	$\sum \|e\|_2$		$\sum \|\tau\|_2$	
(rad/s)	(g)	NA	PID	NA	PID
1	0	0.17	0.08	236.62	252.54
2		0.20	0.17	270.49	327.73
3		0.27	0.28	345.46	457.11
4		0.40	0.49	474.83	730.07
5		0.60	4.31	693.16	5492.36
1	355	0.34	0.10	500.32	478.82
2		0.34	0.21	493.11	544.68
3		0.39	0.35	537.95	669.06
4		0.81	7.28	945.80	9119.96



(a) Neuroadaptive



(b) PID

Figure 3.14. Contact force versus time during the collision experiment.

Table 3.3. Neuroadaptive (NA) and PID controller collision performance results for 10 trails

Controller	$\|F\|_\infty$ (N)		$\|F\|_1 \Delta t$ (N s)	
	Mean	STD	Mean	STD
NA	4.8532	0.7982	0.5577	0.0456
PID	6.9937	0.5683	0.8867	0.0474

3.2 Cartesian Space Controller Formulation

Given this formulation for joint space neuroadaptive robot control in Section 3.1, a neural network controller can now be extended to Cartesian space dynamics as shown in Fig 3.15.

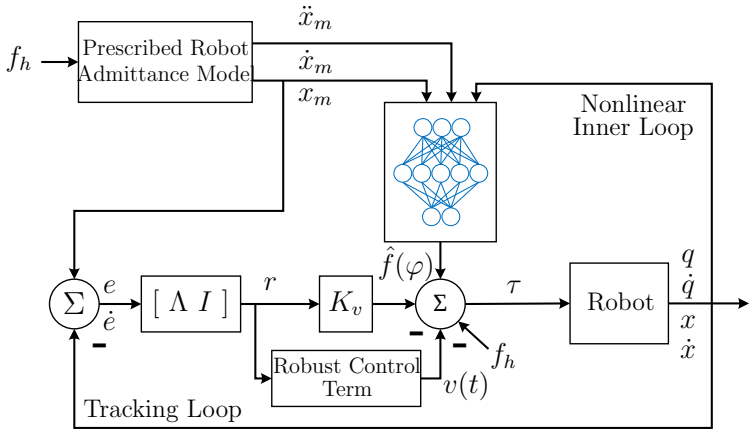


Figure 3.15. Model reference neuroadaptive controller.

Given the model output $x_m(t)$ and robot state $x(t)$, the model following error will be $e = x_m - x$. To drive e to zero, define a sliding mode error $r = \dot{e} + \Gamma e$ where $\Gamma = \Gamma^T > 0$ is a positive definite matrix.

From (3.1), the robot model following error dynamics can be written as

$$\Lambda(q)\dot{r} = -\mu(q,\dot{q})r + f(\varphi) - f_c - f_h \tag{3.21}$$

where

$$f(\varphi) = \Lambda(q)(\ddot{x}_m + \Gamma\dot{e}) + \mu(q,\dot{q})(\dot{x}_m + \Gamma e) + J^{\dagger T}F(\dot{q}) + g_x(q) \tag{3.22}$$

is a nonlinear function of unmodeled robot parameters and argument

$$\varphi = \left[\ e^T\ \dot{e}^T\ x_m^T\ \dot{x}_m^T\ \ddot{x}_m^T\ q^T\ \dot{q}^T\ \right]^T \tag{3.23}$$

This function can be approximated by a neural network

$$f(\varphi) = W^T\sigma(V^T\varphi) + \varepsilon \tag{3.24}$$

where W and V are ideal unknown weights, $\sigma(.)$ is an activation function, and ε is the approximation error, as detailed in [40].

The ideal weights W and V for the neural network (NN) are unknown. Therefore the weight tuning algorithms of [40, 48] are used to update approximate NN weights $\hat{W}$ and $\hat{V}$.

Take the control input as

$$f_c = \hat{W}^T\sigma(\hat{V}^T\varphi) + K_v r - v(t) - f_h \tag{3.25}$$

where $K_v = K_v^T > 0$ is a diagonal outer-loop gain matrix,

$$v(t) = -K_z(\|Z\|_F + Z_B)r \tag{3.26}$$

is the robustifying signal as detailed in [40], K_z is the gain of the robustifying term, $\hat{Z} = \begin{bmatrix} \hat{W} & 0 \\ 0 & \hat{V} \end{bmatrix}$, $\|.\|_F$ is the Frobenius norm, and Z_B is a scalar bound on the NN weights such that $\|\hat{Z}\|_F < Z_B$.

The NN weight update equations are

$$\dot{\hat{W}} = F\sigma(\hat{V}^T\varphi)r^T - F\sigma'(\hat{V}^T\varphi)\hat{V}^T\varphi r^T - \kappa F\|r\|\hat{W} \tag{3.27}$$

$$\dot{\hat{V}} = G\varphi(\sigma'(\hat{V}^T\varphi)^T\hat{W}r)^T - \kappa G\|r\|\hat{V} \tag{3.28}$$

where F and G are positive definite matrices, $\sigma'(\xi) = \frac{d\sigma(\xi)}{d\xi}$, and $\kappa > 0$ is a small design parameter. It can be formally shown using a Lyapunov argument, that under reasonable assumptions, the error signal e will converge to zero. Therefore this inner-loop control scheme tracks Cartesian space trajectories x_m generated by the human and the outer-loop controller [40, 48].

3.2.1 Simulation Results

First the inner-loop controller in Fig. 3.15 was simulated. The model used for simulation is the 2-link planar robot of Example 3.2-2 in [40]. The parameters are $m_1 = 0.8kg$, $m_2 = 2.3kg$, $l_1 = 1m$, $l_2 = 1m$, and $g = 9.8m/s^2$. The prescribed robot admittance model parameters are $M_m = 1$, $D_m = 1$, and $K_m = 1$. The controller parameters are $K_v = 10I_2$, $\Lambda = 20I_2$, $F = 100I_2$, $G = 20I_2$, $\kappa = 0.1$, $K_z = 1$, and $Z_B = 100$, where I_2 is the 2×2 identity matrix. A two-layer Neural Network with 18 inputs, including the bias input, 20 hidden layer neurons and 2 outputs was used. The sigmoid function $\sigma(\xi) = \dfrac{1}{1 + e^{-\xi}}$ is used as the activation function. The weights $\hat{V}$ of the neural network are initialized to random values. The weights $\hat{W}$ are initialized to zero, because random values could destabilize the controller.

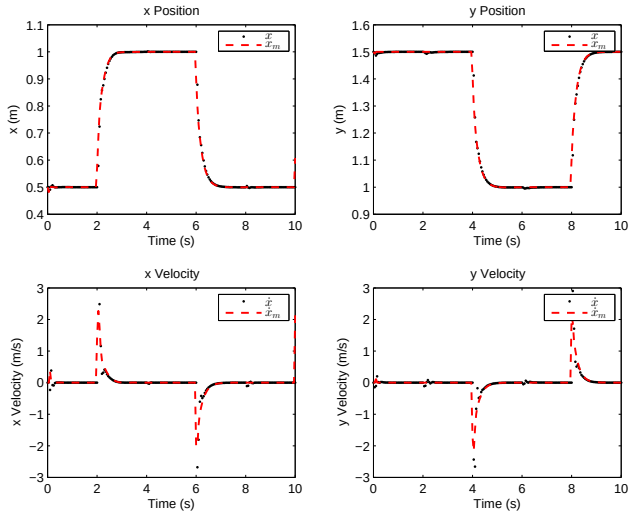


Figure 3.16. Simulation Response with Neuroadaptive Controller.

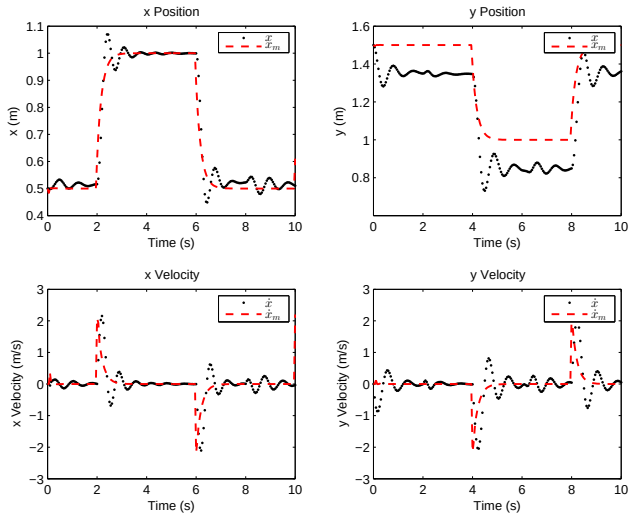
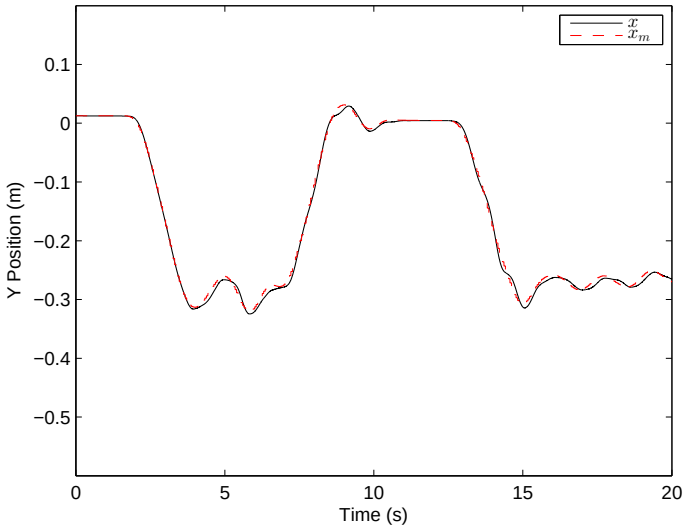


Figure 3.17. Simulation Response without Neuroadaptive Controller.

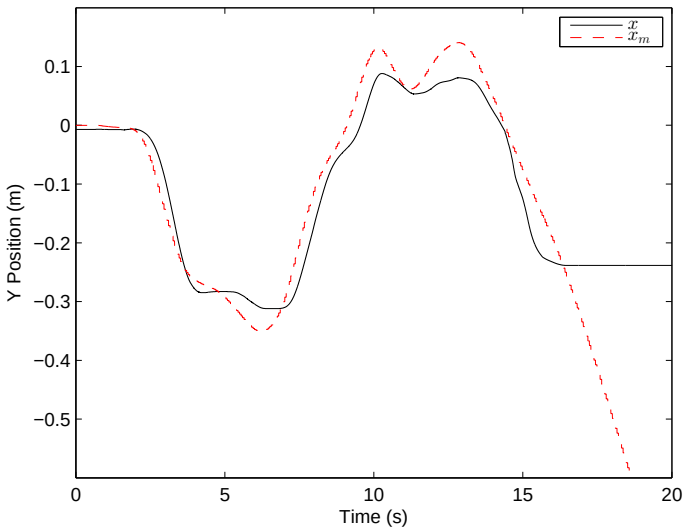
Fig. 3.16 shows that the motion generated by the prescribed robot admittance model x_m and the robot system response x to the force input f_h are virtually identical. Since the Neuroadaptive controller compensates the nonlinear dynamics,

the PD gains of the outer loop can be small. Fig. 3.17 shows the low performance achieved by low PD control gains when the controller (3.25) is implemented without the NN compensation term $\hat{W}^T\sigma(\hat{V}^T\varphi)$ and robustifying term $v(t)$. The performance is noticeably worse than that in Fig. 3.16.

3.2.2 Experimental Results



(a) Inner-loop neuroadaptive controller enabled.



(b) Inner-loop neuroadaptive controller disabled.

Figure 3.18. Inner-loop actual robot vs prescribed robot admittance model PR2 Cartesian space position.

The controller parameters are $K_v = 5I_6$, $\Lambda = 20I_6$, $F = 100I_6$, $G = 200I_6$, $\kappa = 0.3$, $K_z = 0.001$, and $Z_B = 100$, where I_6 is the 6×6 identity matrix. A two-layer Neural Network with 44 inputs including the bias input, 10 hidden layer neurons,

and 6 outputs was used. The sigmoid function $\sigma(\xi)$ is the activation function. As in the simulation the weights $\hat{V}$ of the neural network are initialized to random values and the weights $\hat{W}$ are initialized to zero.

3.3 Summary

In this chapter a neuroadaptive control scheme in joint space and Cartesian space was presented. A stability proof was also outlined. The performance of the neuroadaptive controller was compared against independent joint control via experiments on a PR2 robot. Results demonstrated the effectiveness of the neuroadaptive controllers in joint trajectory following tasks, and during handling of objects with unknown mass. In particular, our controller had superior tracking performance at high joint rates, and much lower joint torques while lifting payloads. Tests were also conducted to demonstrate the performance of the neuroadaptive controller compared to a PID controller during impact. This test was conducted to demonstrate the inherent safety afforded by the neuroadaptive controller by reducing the impact forces.

CHAPTER 4

Adaptive Admittance Control using Adaptive Inverse Filtering

This chapter develops a second loop using an adaptive admittance control for human-robot interaction. This approach follows the human factors studies [60, 61] that indicate human learning in task performance has two components: a robot-specific component whereby a robot dynamics model is learned to compensate for robot nonlinearities, and a component where task-related details are learned. An inner robot torque controller is first designed as shown in Chapter 3 that contains no task information. The objective of this controller is to cause the robot to behave like a prescribed admittance model. This is in contrast to most of the work in robot impedance control [67, 83] and robot neural network control [40, 41] in that the objective is not trajectory following. This robot-specific controller is shown in Fig. 4.1.

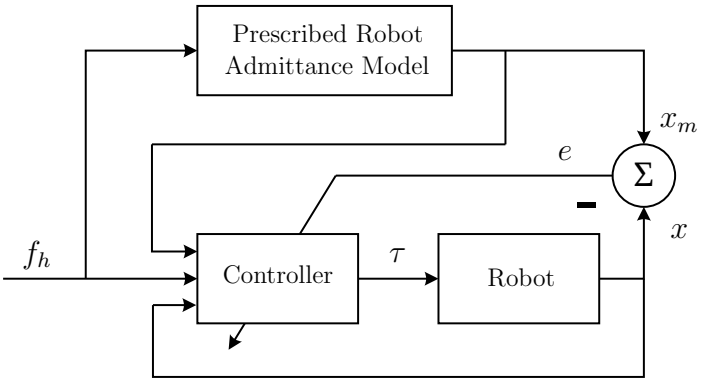


Figure 4.1. Inner-loop robot-specific model reference neuroadaptive control.

Given the inner-loop robot torque control, a task-specific controller is next designed in an outer-loop that takes into account the human dynamics model, which may be unknown, and the task performance objectives. This outer task-specific loop is shown in Fig. 4.2 and adapts the robot admittance model in Fig. 4.1. The robot-specific controller is designed in Section 4.1, and the task-specific control loop is designed in Section 4.4.

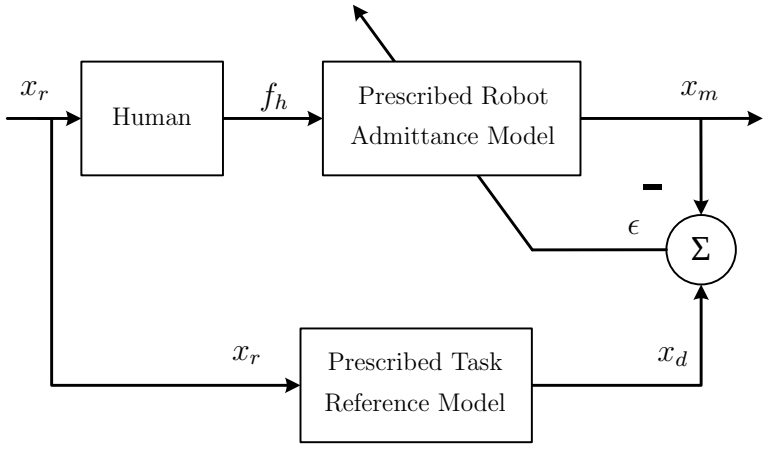


Figure 4.2. Outer-loop task-specific control for adapting robot admittance.

The approach presented here has the additional advantage that if the same task is performed with the same human and a different robot, the inner-loop control automatically adjusts the robot controller so the robot appears to the human as the same admittance model. Also, if the human operator changes, the outer-loop control automatically adapts the robot admittance model to the new human dynamics.

This chapter is organized as follows: In Section 4.1 an inner-loop robot controller that causes a robot to behave like a prescribed admittance model is proposed. In Section 4.4 an outer-loop controller is proposed. This adapts the parameters of the prescribed robot admittance model so that the robot system assists the human achieve

task-specific objectives. In Section 4.4.1 results from simulating the proposed controller on a 2-link planar robot in MATLAB is presented. Experimental results from an implementation case study of the proposed inner-loop and outer-loop controllers on a PR2 robot is presented in Section 4.4.2. A case study to test the controller is proposed in Section 4.7. Finally, Section 4.8 provides a summary of the chapter.

4.1 Inner-loop Robot Controller for Admittance Model Following

4.1.1 Prescribed Robot Admittance Model and Error Dynamics

To change the behavior of the robotic system a prescribed robot admittance model is selected

$$M_m\ddot{x}_m + D_m\dot{x}_m + K_m x_m = f_h \tag{4.1}$$

where M_m is a prescribed mass matrix, D_m is a prescribed damping matrix, and K_m is a prescribed spring constant matrix.

Remark 2

The admittance model (4.1) is the desired robot response from human input force f_h to Cartesian space motion. This is not the same as standard robot control approaches where it is desired for the trajectory tracking error dynamics to have a prescribed impedance model [40, 41, 82, 83].

To design a torque control to make the robot dynamics (3.1) behave like the prescribed admittance model (4.1) define the model-following error

$$e = x_m - x \tag{4.2}$$

The rest of the controller derivation follows that of the Cartesian space controller from Section 3.2.

4.2 Outer Task Loop for Adaptation of Robot Admittance Model in Continuous Time

In the previous section it was shown how to design a neuroadaptive controller to make a robot behave like a prescribed admittance model. That design did not require any task or trajectory information. In this section an outer-loop task-specific controller is proposed. This adapts the parameters of the prescribed robot admittance model (4.1) so that the robot system assists the human to achieve task-specific objectives. This separation of robot-specific design and task-specific design follows human factors studies [60,61] and is facilitated by the fact that the admittance model parameters in (4.1) appear nowhere in the control design in Chapter 3, e.g. equations (3.5), (3.6). See Remark 2. This enables independent design of the inner and outer-loop controllers.

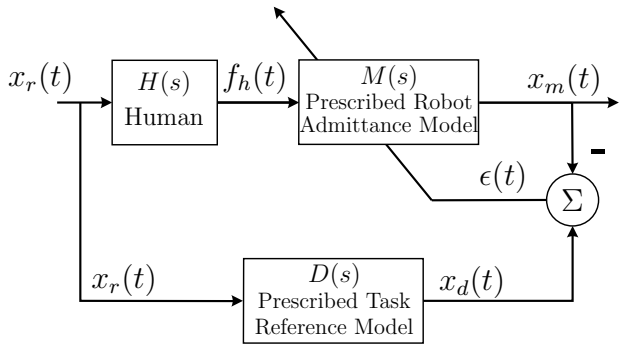


Figure 4.3. Prescribed robot admittance model adaptation using adaptive inverse filtering.

The human, robot admittance model, and task reference model are shown in Fig 4, which further details Fig. 4.2. The unknown human transfer function is denoted as $H(s)$. Studies of the human neuromuscular system have shown that humans adapt to unknown dynamics and task-specific objectives [60, 61]. There is

further evidence [61] that the skilled human operator achieves task-specific objectives by making the combined human-robot system behave like a linear system with wide bandwidth. This is called the crossover model. Motivated by this evidence, the prescribed task reference model is taken as a linear system $D(s)$, which depends on the task being performed.

Our design separation into an inner robot control loop and an outer task performance loop admits the performance of a broad range of tasks. For instance, in medical applications such as stroke rehabilitation repetitive motion tasks based on following medically motivated trajectories are often used [124]. Therefore, we consider here a broad class of tasks which require the human-robot combined system to follow a reference trajectory. In this case, $x_r(t)$ in Fig. 4.3 is the reference trajectory.

The adaptive inverse control approach pioneered by Widrow et al. [125] is used to develop the outer-loop task-specific controller in Fig. 4.3. The approach is to design a method for tuning the admittance model in (4.1), denoted here by $M(s)$, so that the overall human-robot transfer characteristics are given by a prescribed task model $D(s)$.

The derivation of the model following robot control loop in the previous section is valid for either joint space or task space coordinates. To be compatible with these human performance observations, in this section it is assumed that the robot dynamics (3.1) are given in Cartesian space, and that the prescribed admittance model (4.1) is likewise given in Cartesian space. Then, the outer task loop design in Fig. 4.3 uses the measured human force $f_h(t)$ and the desired human-robot combined response $x_d(t)$.

Using the Wiener-Hopf equation and Fourier transform theory [125], the required Wiener filter $M(s)$ that achieves the objective of $M(s)H(s) = D(s)$ can be

solved for in terms of the power spectral densities (PSD) $\Phi_{f_h x_d}$ and $\Phi_{f_h f_h}$ [125, 126] as

$$M(s) = \frac{\Phi_{f_h x_d}(s)}{\Phi_{f_h f_h}(s)} = \frac{D(s)}{H(s)} \tag{4.3}$$

this solution does not require knowledge of the transfer characteristics $H(s)$ or $D(s)$ in Fig. 4.3. It requires knowledge of the complete statistical properties of $f_h(t)$ and $x_d(t)$ to design $M(s)$. Since these are not generally available in pHRI scenarios, an adaptive inverse filter is used to estimate $M(s)$. This only requires measuring $f_h(t)$ and $x_d(t)$. It is known that properly designed adaptive inverse filters converge to the Wiener solution in the mean [125, 126].

To design an adaptive inverse filter the transfer function of the Weiner filter $M(s)$ is taken as

$$M(s) = \frac{b_1 s^{m_\theta - 1} + b_2 s^{m_\theta - 2} ... + b_{m_\theta}}{s^{n_\theta} + a_1 s^{n_\theta - 1} + a_2 s^{n_\theta - 2} ... + a_{n_\theta}} \tag{4.4}$$

where n_θ is the degree of the denominator of $M(s)$ and $m_\theta - 1$ the degree of its numerator. $M(s)$ is implemented as an adaptive ARMA filter using the D3 direct form state-space realization [127] defined as follows.

$$\begin{aligned} \dot{\chi}_i &= \chi_{i+1} \qquad i = 1, ..., m_\theta - 1 \\ \dot{\chi}_{m_\theta} &= f_h(t) \end{aligned} \tag{4.5}$$

$$\begin{aligned} \dot{\phi}_i &= \phi_{i+1} \qquad i = 1, ..., n_\theta - 1 \\ \dot{\phi}_{n_\theta} &= x_d(t) \end{aligned} \tag{4.6}$$

The ideal Wiener filter is written as

$$x_d(t) = h(t)\theta(t) \tag{4.7}$$

where

$$h(t) = [\phi_{n_\theta}, \phi_{n_\theta - 1}, ..., \phi_1, \chi_{m_\theta}, \chi_{m_\theta - 1}, ..., \chi_1] \tag{4.8}$$

$$\theta(t) = [-a_1, -a_2, ..., -a_{n_\theta}, b_1, b_2, ..., b_{m_\theta}]^T \tag{4.9}$$

The robot admittance model output in Fig. 4.3 is generated by the Weiner filter $M(s)$ written as

$$x_m(t) = h(t)\hat{\theta}(t) \tag{4.10}$$

where $\hat{\theta}(t)$ is the estimated parameter vector of Wiener filter coefficients. This parameter vector is updated based on newly observed data $x_d(t)$, $f_h(t)$ by using a Kalman filter [125, 126, 128]. This effectively computes the Weiner-Hopf solution (4.3). The ideal filter coefficients are constant so that

$$\dot{\theta}(t) = 0 \tag{4.11}$$

To set up the Kalman filter, the desired output $x_d(t)$ from the prescribed task model and the input $f_h(t)$ to the adaptive filter are used as the input signals to (4.5),(4.6). Then, the coefficients $\theta(t)$ of the Weiner filter are estimated using the Kalman filter covariance and estimate update equations

$$K(t) = P(t)h^T(t)R^{-1}(t) \tag{4.12}$$

$$\dot{P}(t) = -K(t)h(t)P(t) \tag{4.13}$$

$$\dot{\hat{\theta}}(t) = K(t)[x_d(t) - h(t)\hat{\theta}(t)] \tag{4.14}$$

where $K(t)$ is the filter gain, $P(t)$ is the covariance, $\hat{\theta}(t)$ is the estimated Wiener filter parameter vector, $N_{\hat{\theta}} = n_\theta + m_\theta$ is the size of $\hat{\theta}(t)$. The Kalman filter is initialized by setting $\hat{\theta}(0) = 0$ and $P(0) = \delta^{-1}I_{N_{\hat{\theta}}}$, $I_{N_{\hat{\theta}}}$ is the $N_{\hat{\theta}} \times N_{\hat{\theta}}$ identity matrix. The regularization parameter δ is chosen as a small positive constant for high Signal to Noise ratio (SNR), a large positive constant for low SNR [129]. The Kalman filter converges to the Wiener solution by driving the error residual $\epsilon(t) = x_d(t) - x_m(t) = h(t)[\theta(t) - \hat{\theta}(t)]$, to zero.

It is shown in Theorem 1 in Section 4.3 the Kalman filter parameter vector $\hat{\theta}(t)$ converges to the ideal filter parameter vector $\theta(t)$ of the adaptive inverse filter $M(s)$ which is effectively the prescribed robot admittance model (4.1). Then, $M(s)$ is used to generate the robot admittance model output $x_m(t)$ in (4.2) and its derivatives as required in the robot controller (3.6). The derivatives $\dot{x}_m(t)$ and $\ddot{x}_m(t)$ are obtained using a filtered derivative. The robot admittance model signals $x_m(t)$, $\dot{x}_m(t)$, $\ddot{x}_m(t)$ are finally fed to the robot inner control loop in Fig. 3.15.

4.3 Combined System Stability Analysis

In this section the overall closed-loop stability of the entire proposed two-loop system in Fig. 3.15 and Fig. 4.3 is studied. The following standard assumptions are needed.

Assumption 1

The model trajectory is bounded with q_B a known scalar bound as follows

$$\left\| \begin{array}{c} x_d(t) \\ \dot{x}_d(t) \\ \ddot{x}_d(t) \end{array} \right\| \leq q_B \tag{4.15}$$

Assumption 2

The approximation (3.24) holds with a given accuracy $\|\varepsilon\| \leq \varepsilon_N$ on a compact set [40, 48].

Assumption 3

W_B is a scalar bound on the NN weights such that $\|\hat{W}\|_F < W_B$, where $\|.\|_F$ is the Frobenius norm.

Define the inner-loop weight estimation error as

$$\tilde{W} = W - \hat{W} \tag{4.16}$$

Define the outer-loop adaptive filter parameter estimation error error as

$$\tilde{\theta}(t) = \hat{\theta}(t) - \theta(t) \tag{4.17}$$

The next theorem is the main result in this section.

Theorem 1

Given that the Assumptions 1, 2, and 3 hold. Let the control be designed as in (3.9) and (4.10). Let the NN weight tuning be (3.12), and let $\hat{\theta}(t)$ in (4.10) be updated using (4.12), (4.13), and (4.14). Then the the sliding mode error (3.3), the NN weight estimation error (4.16), and the adaptive filter parameter estimation error (4.17) are UUB.

Proof

The derivative of (4.17) is

$$\dot{\tilde{\theta}}(t) = \dot{\hat{\theta}}(t) - \dot{\theta}(t) \tag{4.18}$$

substituting (4.14) and (4.11) we obtain

$$\dot{\tilde{\theta}}(t) = K(t)[x_d(t) - h(t)\hat{\theta}(t)] \tag{4.19}$$

Furthermore from (4.7) and (4.17) we obtain

$$\dot{\tilde{\theta}}(t) = K(t)[h(t)\theta(t) - h(t)\hat{\theta}(t)] \tag{4.20}$$

$$\dot{\tilde{\theta}}(t) = -K(t)h(t)\tilde{\theta}(t) \tag{4.21}$$

Define the Lyapunov function for the combined inner-loop outer-loop system as

$$\begin{aligned} L[r, \tilde{W}, \tilde{V}, \tilde{\theta}] = \frac{1}{2} r^T M(q) r + \frac{1}{2} tr\{\tilde{W}^T F^{-1} \tilde{W}\} \\ + \tilde{\theta}^T(t) P^{-1}(t) \tilde{\theta}(t) \end{aligned} \tag{4.22}$$

Taking the derivative of $P(t)P^{-1}(t)$ [130] results in

$$\frac{d}{dt}[P(t)P^{-1}(t)] = \dot{P}(t)P^{-1}(t) + P(t)\dot{P}^{-1}(t) = 0 \tag{4.23}$$

rearranging and substituting (4.12), (4.13) gives

$$\dot{P}^{-1}(t) = P^{-1}(t)P(t)h^T(t)R^{-1}(t)h(t)P(t)P^{-1}(t) \tag{4.24}$$

$$\dot{P}^{-1}(t) = h^T(t)R^{-1}(t)h(t) \tag{4.25}$$

Differentiating (4.22) results in

$$\begin{aligned} \dot{L} =& r^T M(q)\dot{r} + \frac{1}{2} r^T \dot{M}(q) r + tr\{\tilde{W}^T F^{-1} \dot{\tilde{W}}\} \\ &+ \dot{\tilde{\theta}}^T(t) P^{-1}(t) \tilde{\theta}(t) + \tilde{\theta}^T(t) P^{-1}(t) \dot{\tilde{\theta}}(t) \\ &+ \tilde{\theta}^T(t) \dot{P}^{-1}(t) \tilde{\theta}(t) \end{aligned} \tag{4.26}$$

substituting from (3.7), (3.24), (3.12), (4.25), and (4.21) and simplifying results in

$$\begin{aligned} \dot{L} =& - r^T K_v r + \frac{1}{2} r^T (\dot{M} - 2V) r + r^T (\varepsilon + \tau_d) \\ &+ \kappa \|r\| tr\{\tilde{W}^T (W - \tilde{W})\} \\ &- \tilde{\theta}^T(t) h^T(t) R^{-1}(t) h(t) \tilde{\theta}(t) \end{aligned} \tag{4.27}$$

Since $\dot{M} - 2V$ is skew symmetric, $\frac{1}{2} r^T (\dot{M} - 2V) r = 0$, and therefore

$$\begin{aligned} \dot{L} \leq & - \zeta(t)^T \zeta(t) - \|r\| [K_{v_{min}} \|r\| \\ &+ \kappa \|\tilde{W}\|_F (\|\tilde{W}\|_F - W_B) - (\varepsilon_N + d_B)] \end{aligned} \tag{4.28}$$

where $K_{v_{min}}$ is the minimum singular value of K_v, d_B is the bound of the disturbance, and $\zeta(t) \equiv \sqrt{R^{-1}}h(t)\tilde{\theta}(t)$.

$$\begin{aligned}\dot{L} \leq & -\zeta(t)^T\zeta(t) - \|r\|[\kappa(\|\tilde{W}\|_F - W_B/2)^2 \\ & - \kappa W_B^2/4 + K_{v_{min}}\|r\| - (\varepsilon_N + d_B)]\end{aligned} \tag{4.29}$$

The term inside the square brackets is guaranteed positive as long as

$$\|r\| > \frac{\kappa W_B^2/4 + (\varepsilon_N + d_B)}{K_{v_{min}}} \equiv b_r \tag{4.30}$$

or

$$\|\tilde{W}_F\| > W_B/2 + \sqrt{\kappa W_B^2/4 + (\varepsilon_N + d_B)/\kappa} \equiv b_W \tag{4.31}$$

Now consider the dynamics (4.21) with output $\zeta(t) = \sqrt{R^{-1}}h(t)\tilde{\theta}(t)$. Then $\zeta(t) \to 0$ implies $\tilde{\theta}(t) \to 0$ if (4.21) and $\zeta(t)$ is uniformly completely observable. This occurs if $\zeta(t)$ is persistently exciting. That is, there exists $T > 0$, $\alpha > 0$ s.t. $\alpha I \leqslant \int_{t-T}^{t} h^T(t)R^{-1}h(t)dt$.

Therefore $\|r\|$, $\|\tilde{W}_F\|$, $\tilde{\theta}(t)$ are UUB and $\tilde{\theta}(t) \to 0$. Thus the two-loop controller is stable. □

4.4 Outer Task Loop for Adaptation of Robot Admittance Model Discrete Time Version

In the previous section the continuous time derivation of the Outer Task Loop was proposed. In this section a discrete time version is proposed. This is more suitable for implementation with a sampled real-time robot controller.

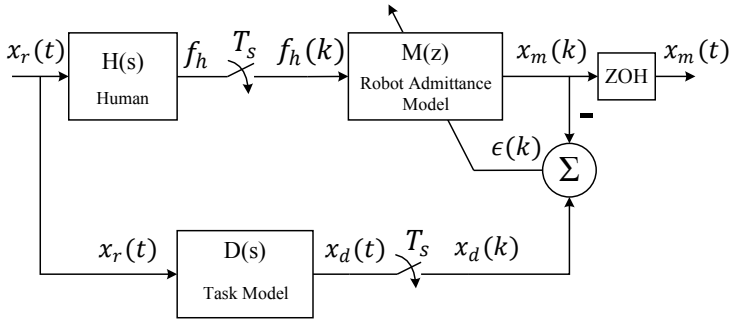


Figure 4.4. Prescribed robot admittance model adaptation using adaptive inverse filtering.

Adaptive inverse control is generally implemented in discrete-time. Therefore let discrete-time models of the human (4.32), inner-loop prescribed robot admittance model (4.1), and outer-loop prescribed task model (4.47) be $H(z)$, $M(z)$, and $D(z)$ respectively.

The human dynamics used is

$$H(s) = \frac{K_d s + K_p}{Ts + 1}, \tag{4.32}$$

this model is widely described in the literature [61]. It is usually described as part of a closed form loop with error as the input; here it is assumed that the input is the reference position and the human is working in open loop.

It is desired to adapt the prescribed robot admittance model $M(z)$ to make the combined human-robot transfer characteristic $H(z)M(z)$ equal the task model $D(z)$, as seen in Fig. 4.4. This outer task loop design uses sampled versions of f_h and the desired model output x_d with a sampling period of T_s, namely $f_h(k) = f_h(kT_s)$, and $x_d(k) = x_d(kT_s)$.

Let $M(z)$ be given in ARMA form by

$$\begin{aligned} x_m(k) = &- a_1 x_m(k-1) ... - a_{n_\theta} x_m(k-n_\theta) + b_0 f_h(k) \\ &+ b_1 f_h(k-1) + ... + b_{m_\theta} f_h(k-m_\theta) \end{aligned} \tag{4.33}$$

where n_θ is the degree of the denominator of $M(z)$ and m_θ the degree of its numerator.

Then $M(z)$ is implemented as

$$x_m(k) \equiv h^T(k)\theta \tag{4.34}$$

where the measured regression vector is

$$\begin{aligned} h^T(k) = [&- x_m(k-1), ..., -x_m(k-n_\theta), \\ &f_h(k), f_h(k-1), ..., f_h(k-m_\theta)] \end{aligned} \tag{4.35}$$

and the ideal ARMA parameter vector is

$$\theta = [a_1 ... a_{n_\theta}, b_0, b_1, ..., b_{m_\theta}] \tag{4.36}$$

The estimated ARMA parameter vector $\hat{\theta}(k)$ can be updated based on newly observed data $x_m(k+1)$, $f_h(k+1)$ by using a Recursive Least Squares (RLS) algorithm [126]. This effectively computes the Weiner-Hopf solution (4.3). The desired output $x_d(k)$ from the prescribed task model and the input $f_h(k)$ to the adaptive filter are used as the input signals to the RLS algorithm. This converges to the Wiener solution by driving the error $\epsilon(k) = x_d(k) - x_m(k)$, to zero which makes the combined human and prescribed robot admittance model behave like the prescribed task reference model by solving $H(z)M(z) = D(z)$.

The RLS covariance and estimate update equations are

$$K(k+1) = \frac{P(k)h(k+1)}{\lambda + h(k+1)P(k)h(k+1)} \tag{4.37}$$

$$P(k+1) = \frac{P(k) - K(k+1)h^T(k+1)P(k)}{\lambda} \tag{4.38}$$

$$\hat{\theta}(k+1) = \hat{\theta}(k) + K(k+1)(x_d(k+1) - h^T(k+1)\hat{\theta}(k)) \tag{4.39}$$

where $K(k)$ is the filter gain, $P(k)$ is the covariance, $\hat{\theta}(k)$ is the ARMA parameter vector, $N_{\hat{\theta}} = n_\theta + m_\theta$ is the size of $\hat{\theta}(k)$, $0 < \lambda \leq 1$ is the forgetting factor.

The RLS algorithm is initialized by setting $\hat{\theta}(0) = 0$ and $P(0) = \delta^{-1} I_{N_{\hat{\theta}}}$, $I_{N_{\hat{\theta}}}$ is the $N_{\hat{\theta}} \times N_{\hat{\theta}}$ identity matrix. The regularization parameter δ is chosen as a small positive constant for high Signal to Noise ratio (SNR), a large positive constant for low SNR [129].

The RLS algorithm converges to the ideal ARMA parameter vector θ of the adaptive inverse filter $M(z)$ which is effectively the prescribed robot admittance model (4.1). It is used to generate the robot admittance model output $x_m(t)$ in (4.2) and its derivatives as required in the robot controller (3.6). The derivatives $\dot{x}_m(t)$ and $\ddot{x}_m(t)$ are obtained by applying zero-order hold (ZOH) and backward difference to $x_m(k)$.

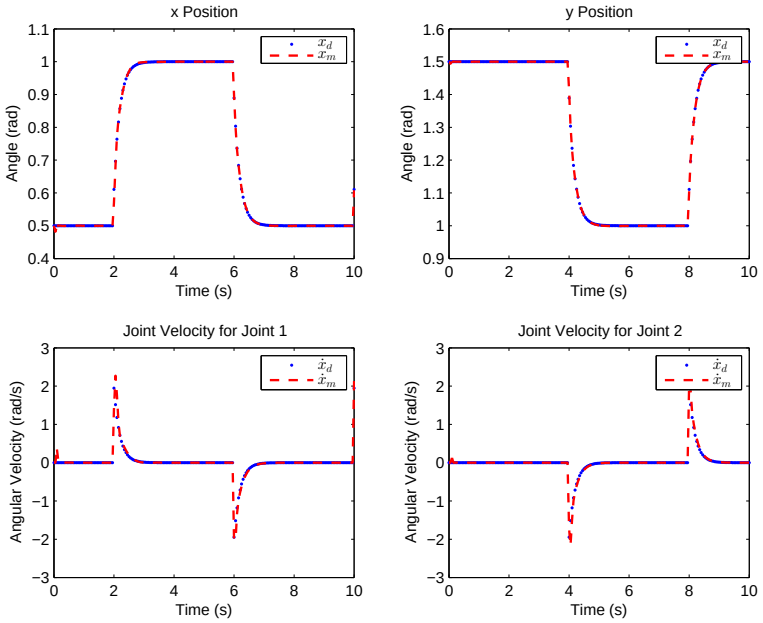
Remark 3

The outer-loop adaptation using RLS with forgetting factor $\lambda < 1$, enables tracking of time-varying human operator dynamics $H(s)$.

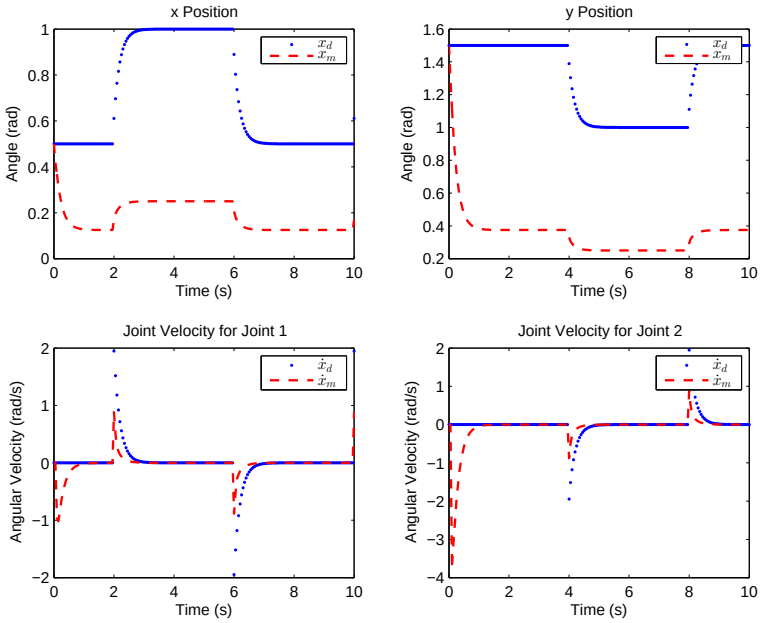
4.4.1 Simulation Results

In this section, the results from simulating the proposed controller on a 2-link planar robot in MATLAB is presented.

The simulation of the inner-loop robot controller is the same as in the previous Chapter. Simulation trials were conducted using 9 different combinations of a_h and b_h values for the human model (4.32), see Table 4.1. These simulate 9 different human operators engaging in pHRI. Two sets of simulations were performed for each of the 9 different combinations of a_h and b_h.



(a) Outer-loop robot admittance model adaptation enabled.



(b) Outer-loop robot admittance model adaptation disabled.

Figure 4.6. Outer-loop simulation response for the human dynamics case H_3.

Table 4.1. Simulation trials with and without outer-loop adaptation for different human models

	Human Model		Set 1 RLS ON	Set 2 RLS OFF
	a_h	b_h	d_e	d_e
H_1	5	5	0.211	7.151
H_2	5	10	0.227	147.036
H_3	5	20	0.276	220.151
H_4	10	5	0.077	287.553
H_5^*	10	10	0.071	0.071
H_6	10	20	0.080	146.256
H_7	20	5	0.041	869.040
H_8	20	10	0.037	294.016
H_9	20	20	0.028	3.491

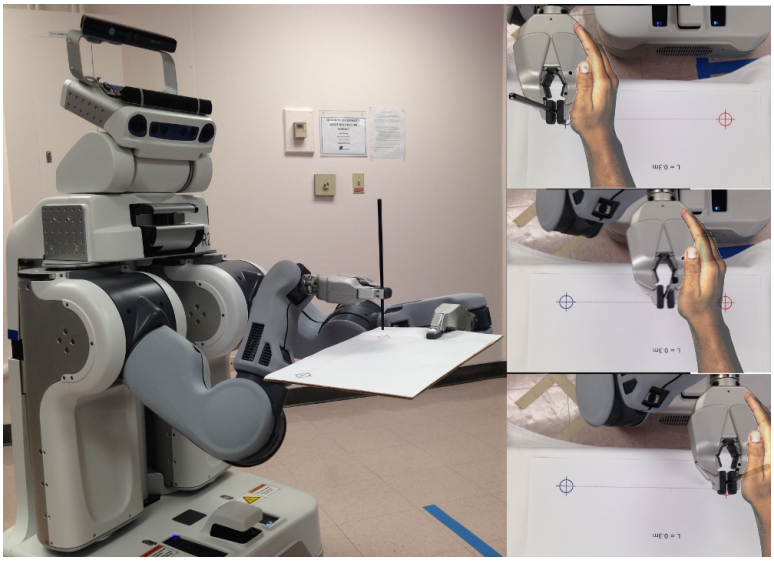
In Set 1, the RLS adaptation in Section 4.4 is enabled, and adapts the robot admittance model. In Set 2, the RLS adaptation is disabled. The tuned ARMA parameters for human model H_5^* are used as the constant prescribed robot admittance model for all the trials with no adaptation. The parameters of the task reference model (4.47) is selected as $a_d = b_d = 10$. The reference task goal x_r is given as a step function, this corresponds to a point-to-point motion in Cartesian space. The prescribed robot admittance model (4.1) is adapted by using the RLS algorithm described in Section 4.4. The outer-loop operates at a sampling frequency of 20 Hz, and uses sampled versions of the inner loop signals f_h and x_d. The size of the ARMA parameter vector (4.36) is $N_{\hat{\theta}} = 8$, and the forgetting factor λ is 0.98. Models are constant during each simulation.

Table 4.1 shows the errors between the desired task model, $x_d(k)$, and prescribed robot admittance model trajectories, $x_m(k)$. The trials with no adaptation, in Set 2, have errors that are orders of magnitude higher than in Set 1. Specifically, Table 4.1 shows the effect of changes to the human model with and without the outer-loop adaptation, where $d_e = \sum_{k=1}^{N} \|e_k\|_2$ with $\|e\|_2 = \sqrt{e_x^2 + e_y^2}$, the 2-norm of the error $\epsilon(k) = x_d(k) - x_m(k)$ for the x and y axis respectively. Moreover, when the human model changes, in Set 1 the adaptive mechanism results in lower outer-loop errors d_e, but when the adaptive mechanism is disabled, in Set 2, the admittance model $x_m(k)$ deviates significantly from the prescribed task model $x_d(k)$, causing large errors d_e. The results show that the outer-loop adaptation makes the combined human-robot system behave like the prescribed task reference model $D(s)$ (4.47) for all the 9 human operator characteristics.

Fig. 4.6a and Fig. 4.6b show the trajectories for a representative run, with and without the RLS adaptation respectively. Here x_d is the desired task reference output and x_m is the desired robot admittance model output. The human model parameters used were $a_h = 5$ and $b_h = 20$, that is case H_3. The tuned ARMA parameters (4.36) for H_5^* were used in the no adaptation case. Fig. 4.6a shows the performance of the controller when the RLS adaptation mechanism is enabled, the admittance model trajectories x_m clearly follow the prescribed task model trajectories x_d. Fig. 4.6b shows the performance when the adaptation mechanism is disabled and fixed ARMA parameters (4.36) are used. The admittance model output x_m does not follow the prescribed task reference model trajectories x_d.

(a) Human interaction.

(b) Point to point motion task.



(c) Experimental setup.

Figure 4.7. Experimental setup with the PR2 in the UTARI Living Lab.

4.4.2 Experimental Results

In this section a case study of the inner-loop and outer-loop controllers implemented on the 7 DOF right arm of the PR2 robot is presented.

The experiments were conducted on the UT Arlington Research Institute PR2 robot, which is currently used in its Living Laboratory, see Fig. 4.7a. The controller was implemented using the real-time controller manager framework of the PR2 in ROS Groovy. The real-time loop on the PR2 runs at 1000 Hz and communicates with the sensors and actuators on an EtherCAT network. Human force is measured using an ATI Mini40 Force/Torque (FT) Sensor attached between the gripper and forearm

of the PR2. The inner-loop runs at a loop rate of 1000 Hz and the outer-loop runs at 20 Hz.

The experimental setup was the same for both the inner-loop and outer-loop experiments. A person was asked to sit in front of the PR2, at a fixed distance as seen in 4.7a. They were asked to hold the gripper of the PR2 with their right hand and move between two points $P1$ (Red) and $P2$ (Blue) which are along the Y axis of the robot frame seen in Figs. 4.7a, 4.7b, and 4.7c. The controller used all 7 joints of the right arm, and used the FT sensor to detect human force input f_h.

In this experiment the outer-loop task specific controller in Fig. 4.4 is enabled. This experiment demonstrates the ability of the outer-loop task specific controller to adapt to different human dynamics. Two sets of experiments were conducted with three male human subjects between the ages of 20 and 30. In Set 1, the experiments were conducted with the RLS adaptation enabled. In Set 2, the RLS adaptation was disabled and the ARMA filter parameters (4.36) were fixed to those tuned for subject S_1^*. The point-to-point reference task goal x_r is sent to the user via voice commands. The PR2 says '*red*' when the goal is P_1 and '*blue*' when the goal is P_2. This keeps the task reference internal to the robot and external to the human in sync.

Table 4.2. Mean μ and standard deviation σ of error d_e for different human subjects with and without outer-loop adaptation

	Set 1 RLS ON		Set 2 RLS OFF	
	Mean μ	σ	Mean μ	σ
S_1^*	69.7241	8.2262	273.8417	28.6228
S_2	76.2001	29.6413	182.1220	18.7060
S_3	55.6336	4.2584	257.2599	92.8642

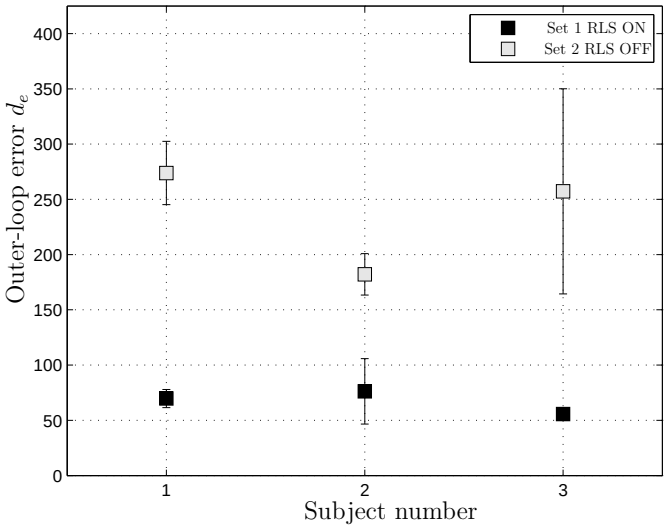
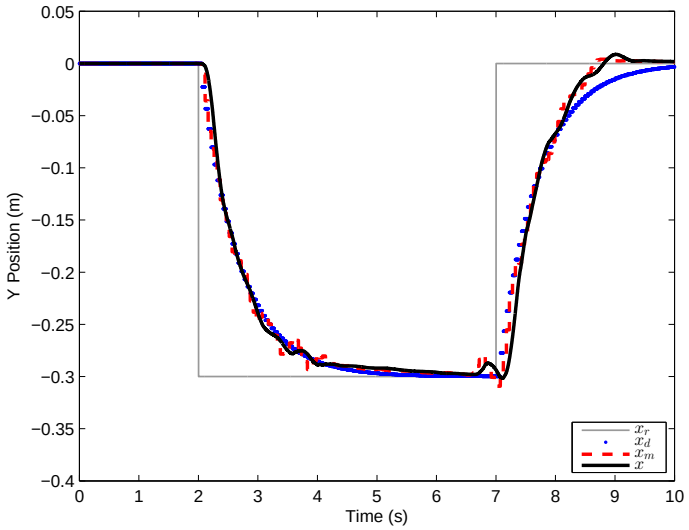
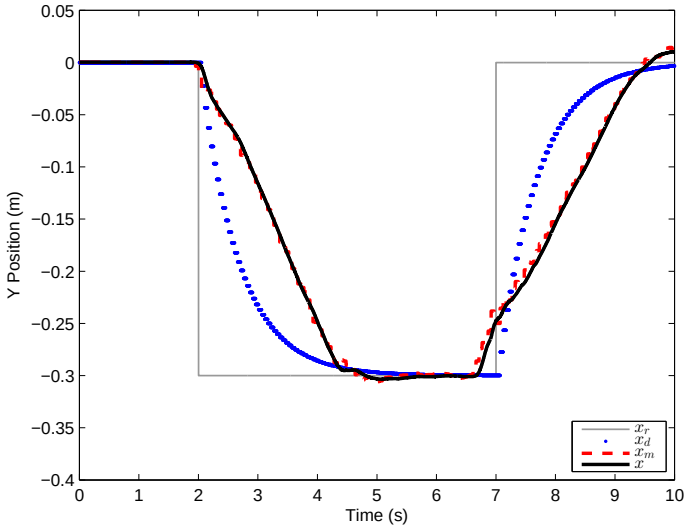


Figure 4.8. Outer-loop error plot for subjects S_1, S_2, and S_3.

Table 4.2 shows the experimental data. Each of the three subjects performed the experiment 6 times, 3 times for Set 1 and 3 times for Set 2. The mean μ and standard deviation σ per user per set are presented in Table 4.2 and Fig. 4.8. The outer-loop error d_e is calculated as $d_e = \sum_{k=1}^{N} \|e_k\|_2$ with $\|e\|_2$, the 2-norm of the error $\epsilon(k) = x_d(k) - x_m(k)$. The data show that this error is lower in Set 1 for all subjects where the RLS filter is enabled. This indicates that the outer-loop adaptation assists the subject perform the point-to-point task by adapting the robot admittance model.



(a) Outer-loop robot admittance model adaptation enabled.



(b) Outer-loop robot admittance model adaptation disabled.

Figure 4.9. Human experiment with actual robot motion vs fixed ARMA prescribed robot admittance model output on PR2 for subject S_3.

Here, the outer-loop Fig. 4.4 adapts the admittance model (4.33) of the inner-loop Fig. 3.15. Fig. 4.9a shows that the robot and human combined motion x follow the prescribed task reference model output x_d. Here x_r is the point-to-point reference task goal, x_d is the desired task reference trajectory, x_m is the output of the desired

robot admittance model, and x is the robot motion. Fig. 4.9b shows the performance when RLS adaptation of the robot admittance model is disabled and fixed ARMA parameters (4.36) are used. The performance is significantly degraded in comparison to Fig. 4.9a

This experiment shows that the outer-loop and inner-loop controllers combined can successfully make the combined human and robot system behave like a prescribed task reference model adapted to the human user.

4.5 Adaptive Physical HRI Scheme with Intent Estimation

In this section the human intent aware adaptive admittance control methodology is presented. First, the inner-loop and outer-loop controller formulations are summarized. Then the outer-loop adaptive admittance controller that includes human intent estimation and nominal task models is proposed. Finally, the inner-loop neuroadaptive controller is described.

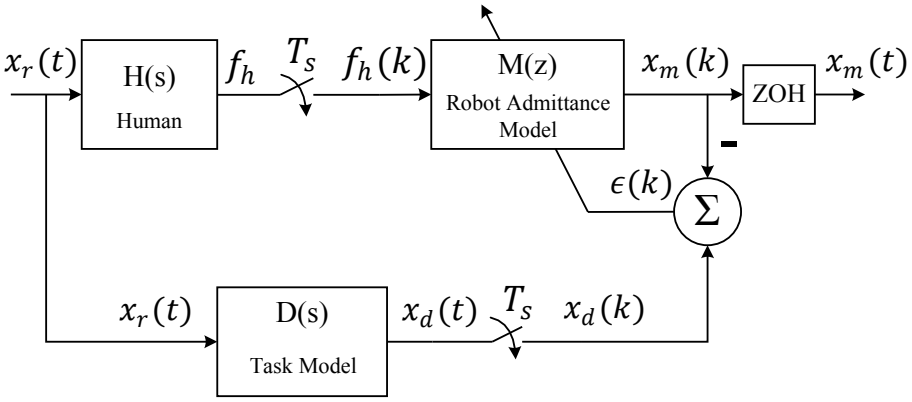


Figure 4.10. Prescribed robot admittance model adaptation using adaptive inverse filtering.

The proposed controller structure consists of: (a) an inner-loop controller utilizing online learning with neural networks to estimate and cancel the nonlinear robot dynamics; (b) an adaptive admittance model that can be programmed to follow known robot Cartesian trajectories x_i or estimated trajectories $\hat{x}_i$ resulting from a human intent estimator.

The method proposed consists of an inner-loop dynamic compensation controller with an outer-loop adaptive admittance controller. The inner-loop controller is an adaptive controller that utilizes online learning neural networks to estimate and cancel the nonlinear robot dynamics.

For the pHRI problem, the general dynamics equation in Cartesian space is written as follows [131]

$$\Lambda(q)\ddot{x} + \mu(q,\dot{q})\dot{x} + J^{\dagger T}F(\dot{q}) + g_x(q) = f_c + f_h \tag{4.40}$$

where $q \in \mathbb{R}^n$ are the joint positions, n is the DOF of the robot, $x = \binom{p_e}{\phi_e} \in \mathbb{R}^6$ is the Cartesian space pose, where $p_e \in \mathbb{R}^3$ is the task space position and $\phi_e \in \mathbb{R}^3$ the orientation. $\Lambda(q)$ is the Cartesian space inertia matrix, $\mu(q,\dot{q})$ is the Cartesian space Coriolis/centripetal vector, $g_x(q)$ is the Cartesian space gravity vector, and $F(\dot{q})$ is the joint space friction term. J is the Jacobian and $J^{\dagger} = J^T(JJ^T + k^2 I)^{-1}$ is the damped least-squares pseudoinverse of the Jacobian with damping factor k. The force applied by a human operator in Cartesian space is $f_h \in \mathbb{R}^6$. The Cartesian space control force $f_c \in \mathbb{R}^6$ will be computed via a neuroadaptive scheme and can be used to compute the joint control torque $\tau = J^T f_c$.

Assume the following robot admittance model

$$M_m\ddot{x}_m + D_m\dot{x}_m = f_h \tag{4.41}$$

where x_m is the admittance model output. The matrices M_m and D_m are the mass and damping matrices respectively. In this section we assume zero stiffness $K_m = 0$, and no virtual trajectory. Therefore the admittance model is a simple mass-damper system in Cartesian space.

In Section 4.5.1, an adaptive admittance controller with human intent is proposed to generate desired trajectories for the robot to follow. This is shown in Fig. 4.11. If the ideal intent trajectory x_i is known, as it is usually the case in rehabilitation exercises, then it is sent to the human subject as an audiovisual cue and to a robot task model $D(s)$ which generates x_d. An adaptive filter with human force f_h as an input and output x_m will then be employed to minimize the error between x_m and x_d.

If the intent trajectory is not known, a human intent estimator can estimate it from interaction forces and/or other sensor measurements. In Figure 4.11, the estimated human intent trajectory is denoted by $\hat{x}_i \approx x_i$. The human model can generally be defined as a transfer function $H(s)$ assumed to be unknown, and representing the dynamics of thinking and completing a trajectory tracking task by physically guiding the robot.

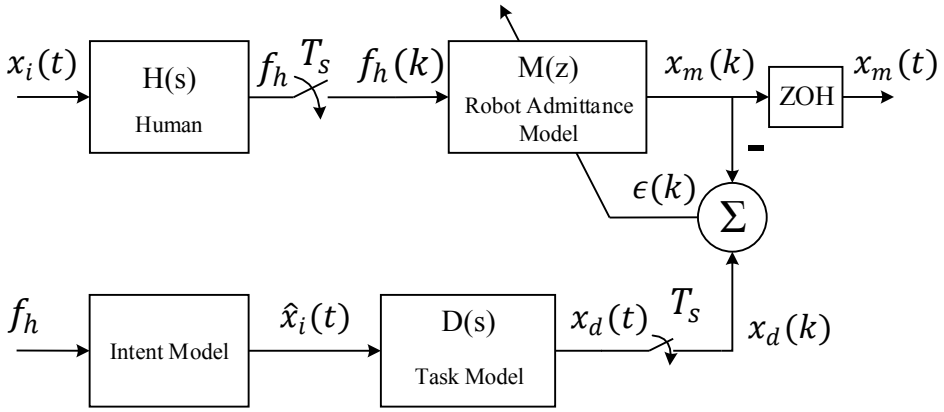


Figure 4.11. Human intent aware robot admittance model adaptation using adaptive inverse filtering.

4.5.1 Outer-Loop: Adaptive Admittance with Human Intent

The outer-loop adaptive admittance controller is inspired by the adaptive inverse control approach pioneered by Widrow et al. [125]. The objective is to tune the admittance model in (4.41), denoted here by the discrete transfer function $M(z)$, such that the overall human-robot transfer characteristics $H(z)M(z)$ equals the task model $D(z)$. This outer-loop design uses sampled versions of f_h and the desired model output $x_d(t)$ with a sampling period of T_s, namely $f_h(k) = f_h(kT_s)$ and $x_d(k) = x_d(kT_s)$.

Let $M(z)$ be given in ARMA form by

$$\begin{aligned} x_m(k) &= -a_1 x_m(k-1) \; ... - a_{n_\theta} x_m(k-n_\theta) \\ &+ b_0 f_h(k) + b_1 f_h(k-1) + ... + b_{m_\theta} f_h(k-m_\theta) \end{aligned} \tag{4.42}$$

where n_θ is the degree of the denominator of $M(z)$ and m_θ the degree of its numerator. Then $M(z)$ is implemented as

$$x_m(k) \equiv h^T(k)\theta \tag{4.43}$$

where the measured regression vector is

$$h^T(k) = [\,-x_m(k-1),\ \ldots\ , -x_m(k-n_\theta), \\ f_h(k), f_h(k-1),\ \ldots\ , f_h(k-m_\theta)] \tag{4.44}$$

and the ideal ARMA parameter vector is

$$\theta = [a_1,\ \ldots\ , a_{n_\theta}, b_0, b_1,\ \ldots\ , b_{m_\theta}] \tag{4.45}$$

The estimated ARMA parameter vector $\hat{\theta}(k)$ can be updated based on newly observed data $x_m(k+1)$, $f_h(k+1)$ by using a Recursive Least Squares (RLS) algorithm. This effectively converges to the Wiener solution by driving the error, $\epsilon(k) = x_d(k) - x_m(k)$, to zero which makes the combined human and prescribed robot admittance model behave like the task model by solving $H(z)M(z) = D(z)$. The derivatives $\dot{x}_m(t)$ and $\ddot{x}_m(t)$ are obtained by applying zero-order hold (ZOH) and backward difference to $x_m(k)$.

The above controller formulation assumes that the ideal human intent trajectory $x_i(t)$ is known by the robot. In reality this signal is not available a priori. If the human intent is not taken into account by the outer-loop controller, the goal position is fixed and communicated to the human. This makes it impossible for the human subject to change the motion of the robot at will, and applications are restricted to rehabilitation exercises, or following known trajectories. In this case, if the user changes the goal in his/her mind, the controller will have no information about the new intent $\hat{x}_i(t)$ and will fight against the user's movements.

Therefore, in a second formulation, we add a human intent estimator to generate approximations of x_i. In this section, a simple model was tested which converts the user applied force on the robot to a desired future position propagated by some time

interval. The following admittance model output is propagated into the future and it was assumed that this roughly approximates human intent $x_i(t)$

$$M_i \ddot{\hat{x}}_i = f_h, \qquad \hat{x}_i(t) = \iint_t^{t+\Delta t} \ddot{\hat{x}}_i(t) dt \tag{4.46}$$

where M_i is a fixed mass matrix, and f_h is the measured human force, and Δt is the propagation time. Thus, the human intent is obtained by double integration of the estimated intent acceleration $\ddot{\hat{x}}_i(t)$. This simple model has been utilized for human walking path predictions in social situations by Luber et al. [132]. In this work, the future motion paths of humans were predicted using environmental constraint forces and intent forces towards a known goal.

4.5.2 Experiments

Validation experiments were conducted on a PR2 robot which has an omnidirectional mobile base, two 7 DOF gravity compensated arms with parallel grippers, a pan tilt head, and several sensor modules. The controller was implemented using the real-time controller manager framework of the PR2 available via the Robot Operating System (ROS) [133]. The real-time loop on the PR2 runs at 1000 Hz and is implemented using the linux-rt kernel. The communication with the sensors and actuators is via an EtherCAT network. Interaction forces and torques are measured using an ATI Mini40 Force/Torque (FT) sensor attached between the gripper and forearm of the PR2.

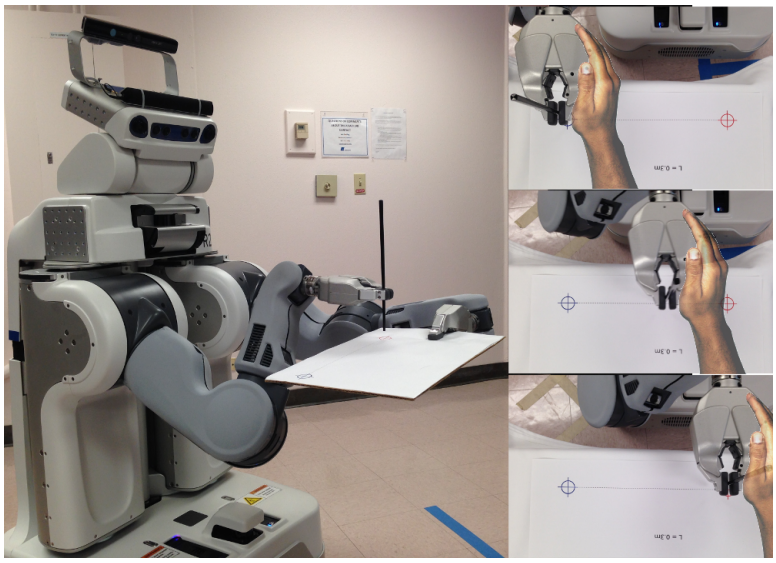


Figure 4.12. Experimental setup with the PR2 at the UTARI Living Lab.

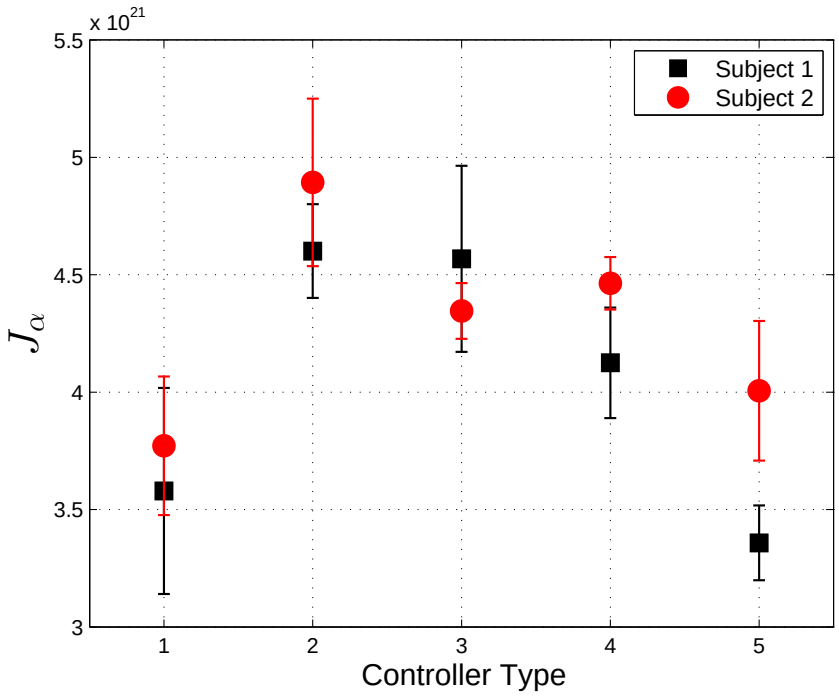


Figure 4.13. Error plot of J_α for controller Types 1 through 5.

4.5.2.1 Experimental Setup

The experiments performed involved a point-to-point motion task. This task corresponds to a step function input to the system and enables clear analysis of the system performance. The PR2 robot was setup as shown in Fig. 4.12, where the subjects were asked to sit in front of the robot and grasp the right gripper with their dominant hand. The start of the experiment was automatically indicated by the system via a text-to-speech program. The reference point-to-point motion was indicated by a "RED" or "BLUE" audiovisual cue to indicate the next movement point. Human factors studies [61] suggest that the human brain learns a closed-loop controller that behaves like a first order model with high bandwidth in closed-loop. Motivated by this a first-order transfer function is used as the task model

$$D(s) = \frac{a_d}{s + b_d} \tag{4.47}$$

where the model parameters depend on the specific task.

Experiments were conducted with the following five different types of controllers:

Type 1 *Adaptive admittance controller with human intent estimation*: In this experiment the admittance model in (4.43) is utilized. The admittance model in (4.46) is used to generate a time propagated signal corresponding to the human intent $\hat{x}_i(t)$, where $\Delta t = 0.1$s. The matrix M_i is the 6×6 identity matrix. The parameters of $D(s)$ in (4.47) are selected as $a_d = b_d = 1.5$.

Type 2 *Adaptive admittance controller with fixed task reference trajectory*: In this experiment the admittance model in (4.43) is utilized with a fixed task reference trajectory $x_i(t)$. The parameters of $D(s)$ are the same as for Type 1.

Type 3 *Admittance controller with fixed ARMA parameters*: In this experiment the ARMA parameter weights (4.45) were automatically tuned to a human subject and fixed as

$$\begin{aligned}\theta = [\ &-1.8775 \quad 1.2729 \quad -0.0578 \quad -0.3269 \\ &0.0018 \quad 0.0001 \quad -0.0013 \quad 0.0002\ ]\end{aligned} \tag{4.48}$$

Type 4 *Fixed mass-damper admittance controller*: In this experiment the fixed mass-damper admittance model in (4.41) is used with the parameters tuned for Subject 1 as $M_m = 20I_6$, and $D_m = 50I_6$, where I_6 is the 6×6 identity matrix. This model converts the user applied force f_h to Cartesian positions $x_m(t)$ and velocities $\dot{x}_m(t)$.

Type 5 *Direct task model controller*: In this experiment the output of the task reference model is sent directly to the inner-loop controller. The user applied force f_h is converted to Cartesian positions $x_d(t)$ via the intent estimator.

Three experimental trials were conducted for each type of controller with two male human subjects of ages 26 and 27. For all experiments, the inner-loop rate was 1000 Hz and the outer-loop rate was 20 Hz. The performance tuned inner-loop neuroadaptive controller parameters were $\Gamma = 20I_6$, $K_v = 5I_6$, $K_z = 0.001$, $Z_B = 100$, $F = 100I_6$, $G = 200I_6$, and $\kappa = 0.3$, where I_6 is the 6×6 identity matrix. A two-layer NN with 44 inputs (including the bias input), 10 hidden layer neurons, and 6 outputs was implemented. The sigmoid function was used as the activation function $\sigma(\xi)$. The weights $\hat{V}$ of the neural network were initialized to random values and the weights $\hat{W}$ were initialized to zero.

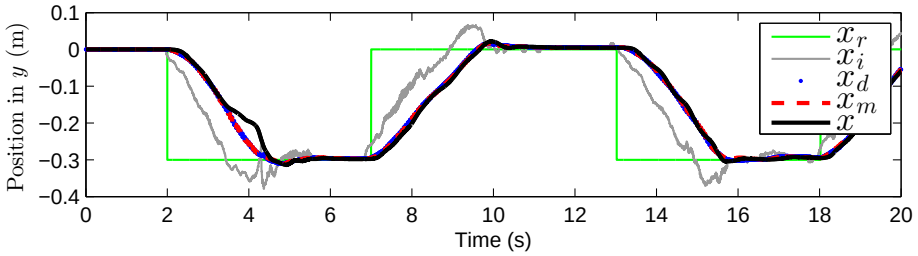
4.5.2.2 Performance Measure

To compare the performance of the different controllers a performance measure J_α is utilized. It is the dimensionless squared jerk of the gripper motion in Cartesian

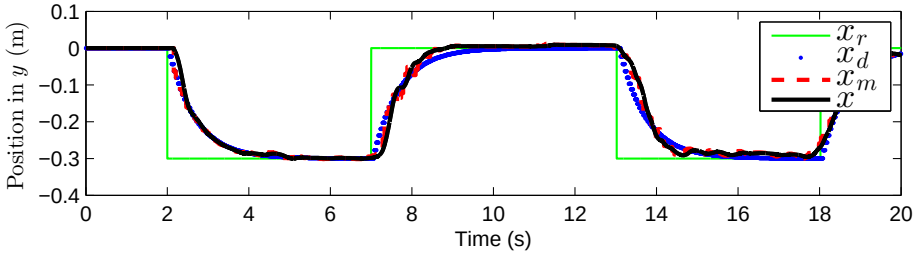
space $\dddot{x}(t)$. The work by Flash and Hogan [134] provides experimental evidence for a human motion model involving the minimization of jerk. Among the different jerk based performance measures proposed, the dimensionless squared jerk has been shown to be the most effective [134]. It is defined as

$$J_\alpha = \int_{t_1}^{t_2} \dddot{x}(t)^2 dt \; \frac{(t_2 - t_1)^5}{A^2} \tag{4.49}$$

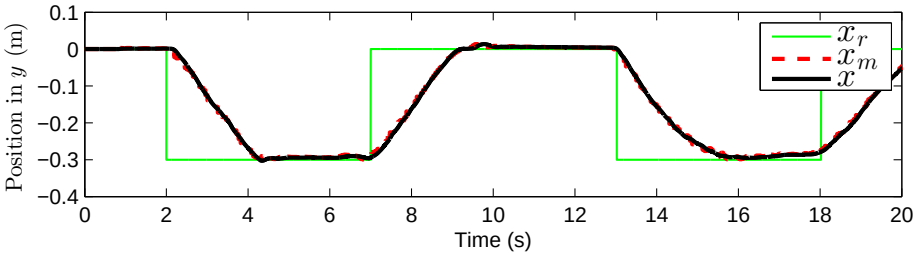
where A is the maximum amplitude of $x(t)$. Smaller values of J_α indicate better human-robot interaction performance.



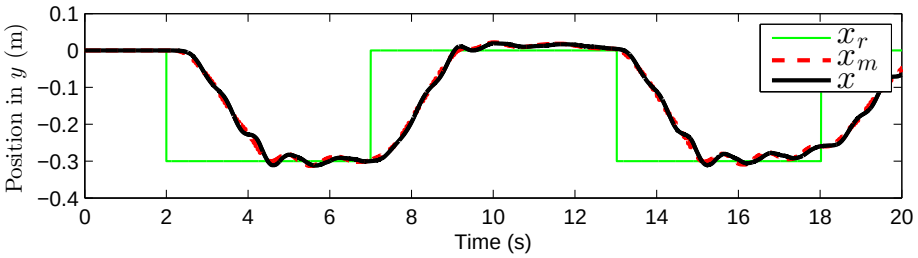
(a) Type 1: Adaptive admittance controller with human intent.



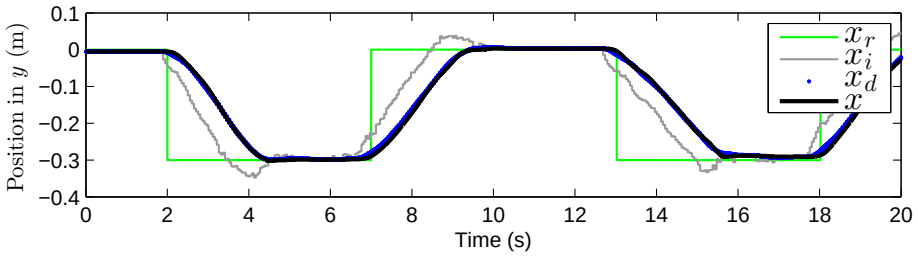
(b) Type 2: Adaptive admittance controller with fixed task reference.



(c) Type 3: Admittance controller with fixed ARMA parameters.

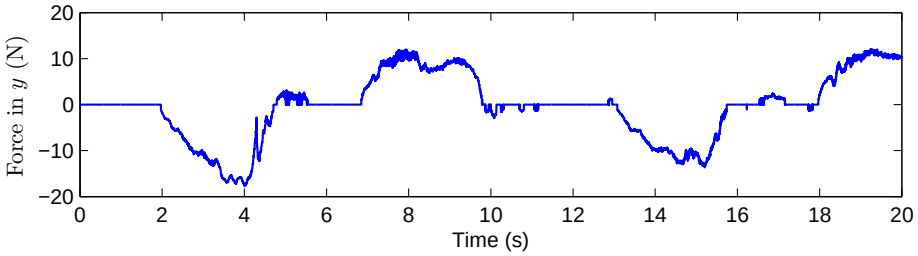


(d) Type 4: Fixed mass-damper admittance controller.

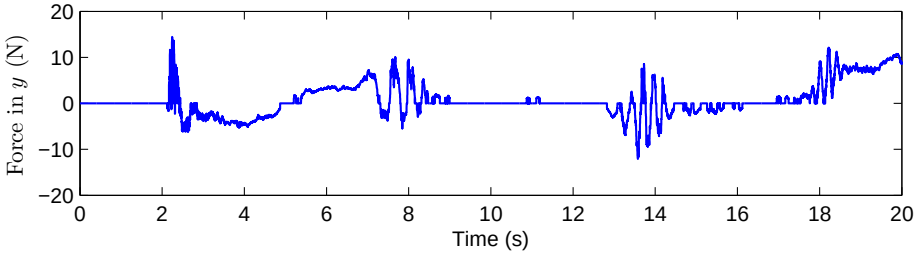


(e) Type 5: Direct task model controller.

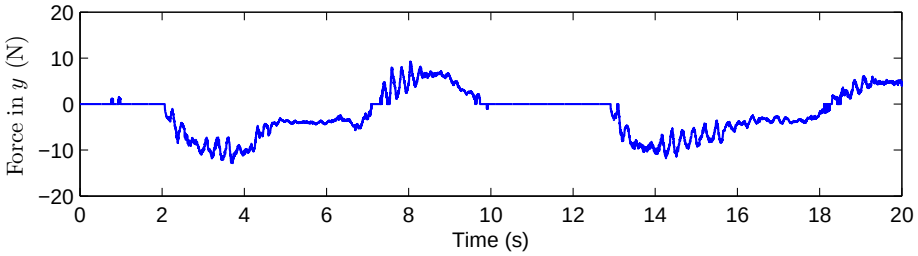
Figure 4.14. Postion in y (m) versus time (s).



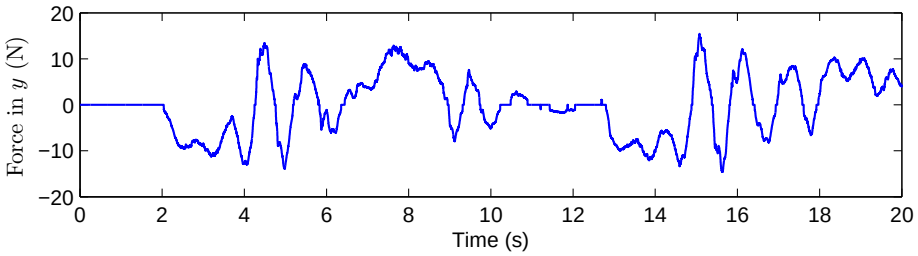
(a) Type 1: Adaptive admittance controller with human intent.



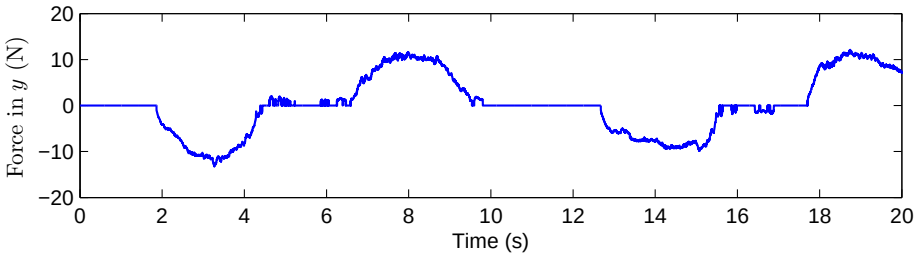
(b) Type 2: Adaptive admittance controller with fixed task reference.



(c) Type 3: Admittance controller with fixed ARMA parameters.



(d) Type 4: Fixed mass-damper admittance controller.



(e) Type 5: Direct task model controller.

Figure 4.15. Force in y (N) versus time (s).

4.5.2.3 Results

Table 4.3. Mean μ and standard deviation σ of J_α for controller types 1 through 5. In units of 10^{21}

	Subject 1		Subject 2	
	Mean μ	σ	Mean μ	σ
T_1	3.5791	0.4388	3.7717	0.2948
T_2	4.6010	0.1999	4.8935	0.3570
T_3	4.5680	0.3962	4.3461	0.1187
T_4	4.1249	0.2351	4.4638	0.1117
T_5	3.3580	0.1593	4.0056	0.2973

The best performance in terms of trajectory smoothness is achieved by the adaptive admittance controller with human intent estimation (Type 1) and the direct task model controller (Type 5). This is shown by the dimensionless squared jerk performance measure J_α in Fig. 4.13 and Table 4.3, and is consistent for both human subjects. The outer-loop in the Type 1 controller successfully tunes the admittance model based on the estimated human intent $\hat{x}_i(t)$, resulting in admittance model trajectories $x_m(t)$ that closely follow the task model trajectories $x_d(t)$. The inner-loop controller enables accurate tracking of the admittance model trajectories. This results in robot trajectories $x(t)$ that closely follow the admittance model trajectories $x_m(t)$ shown in Fig. 4.14a. The controller also exhibits minimal force jitter as seen in Fig. 4.15a, resulting in a smooth force profile which indicates good interaction performance. The direct task model controller (Type 5) controller does not include an adaptive component, but due to the structure chosen for this system, results in smooth trajectories.

The controller of Type 2 with a fixed trajectory reference $x_i(t)$ does not perform as well as the Type 1 and Type 5 controllers. This is indicated by higher values of J_α for Type 2. The minimization of the error between the task model trajectory $x_i(t)$ and the admittance model trajectory $x_m(t)$ tunes the outer-loop admittance model $M(z)$. This could result in a conflict between the robot trajectory and the human intent since priority is given to task model following. Evidence of this conflict is seen in the interaction force f_h oscillations in Fig. 4.15b. Also, Fig. 4.14b shows poor tracking performance of the task model $x_d(t)$, admittance model $x_m(t)$, and robot trajectories $x(t)$.

The controllers of Types 3 and 4 with fixed ARMA weights and mass-damper do not include a desired model trajectory $x_d(t)$ that could result in the controller competing with the human as in Type 2. These controllers perform marginally better than the Type 2 controller indicated by lower values of J_α. Although these controllers achieve relatively smooth position tracking as seen in Figs. 4.14c and 4.14d, the force profiles in Figs. 4.15c and 4.15d exhibit considerable jitter and oscillations.

The Type 3 controller utilizes the fixed ARMA model (4.48) that was automatically tuned to the human subject. The a priori tuning of the Type 3 controller ARMA weights is more efficient and precise than manually tuning a mass-damper system based on subjective human preferences. The manually tuned Type 4 controller still exhibits oscillations at steady state as shown in Fig. 4.14d.

Overall, the results provide evidence of the viability of the Type 1 adaptive admittance controller with human intent estimation in pHRI applications. The direct model reference controller which included no adaptation (Type 5) performed similar to the adaptive admittance controller (Type 1). The results for subject 2 showed better performance with the Type 1 controller while subject 1 favored the Type 5 controller. This requires further study by performing tests with different human

subjects, intent estimators, and task reference models. However, the Type 1 controller can adapt online in real-time to changes in human dynamics and intent. It can further be used to learn the optimal admittance using more complex intent estimator models. These complex models can then be replaced by the learnt admittance model further reducing final system complexity. The Type 2 controller can be utilized for automated rehabilitation systems where the human intent, i.e. the desired rehabilitation motion, is known. Here, the admittance model can be automatically tuned to the strength and ability of the patient. The Type 3 controller is a static tuned admittance model, such a controller can be used for instances where a one time admittance optimization is required, such as calibrating the system to different human operators.

4.5.3 Discussion

In this section a two-loop adaptive admittance controller framework is presented that includes human intent estimation. An inner neuroadaptive loop feedback linearizes the robot, and an outer model reference adaptive loop tunes the admittance model based on the intent of the human. The inner-loop neuroadaptive controller linearizes the robot dynamics and ensures consistent performance of the outer-loop across the entire robot workspace.

Experiments were conducted with five different types of admittance controllers. The results demonstrated the performance advantages of the adaptive admittance controller with human intent estimation. The proposed method resulted in lower jerk in the combined human-robot motion, does not need any offline tuning, is robust to changes in human dynamics, and also compensates for changing robot dynamics.

4.6 Tactile Robotic Skin Calibration

There has been considerable effort to calibrate tactile sensors to generate correct pressure of force values. This is necessary to perform useful applications with the sensor data. Calibration procedures are usually time-consuming, needs accurate instruments, and has to be repeated regularly due to sensor property changes. Thus calibrating large arrays of sensors is not only difficult, but also unmaintainable.

In the previous sections we assumed the force is measured directly as $f_h(t)$, however in the context of arrays of taxels as in robot skin such assumptions are invalid. To reformulate the adaptive framework proposed let us define the force detected per taxel as $f_{h_i}(t)$ for $i = 1, ...N$, where N is the total number of taxels.

Now we can redefine the ARMA model per taxel as $M(z)_i$ for $i = 1, ...N$. This allows us to generate a model that encodes not only the human preferences, it also enables the encoding of the sensor calibration information.

4.6.1 Experiments

In this section, we describe the experiments conducted to assess the effectiveness of the proposed method in overcoming sensor calibration issues.

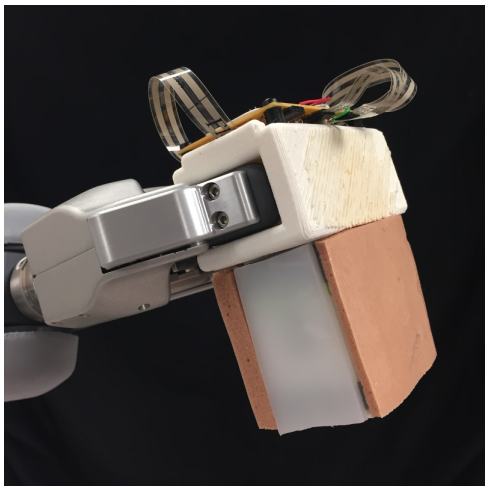


Figure 4.16. Tactile sensor experiment setup.

The experiment uses FlexiForce sensors mounted around 4 sides of a handle as seen in Fig. 4.16. Different materials of various thickness are placed over the sensors to simulate a situation involving robot skin.

The following are the thickness and type of the materials used:

1. Frubber 5 mm
2. Frubber 10 mm
3. Orthoflex 5 mm
4. Proflex 5 mm

A microcontroller reads the voltage drop across the FlexiForce sensor and converts it to a value between 0 and 1024. This value is filtered using a lowpass Butterworth filter with a 1000 Hz cutoff then divided by 100 to get the force data. This results in inaccurate and uncalibrated readings. The data is collected at a rate of 1000 Hz.

As before, the 7 DOF arms of the PR2 is used. Both controllers were implemented using the real-time control manager framework of the PR2. The real-time

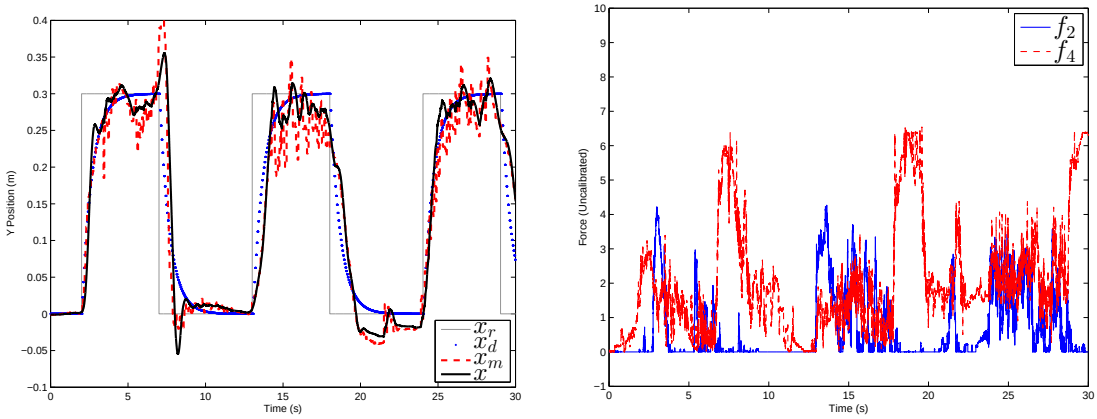
loop on the PR2 runs at 1000 Hz and communicates with the sensors and actuators on an EtherCAT network. ROS Groovy [133] was used for all the experiments.

Four different experiments were conducted with different controller configurations, they were:

1. Adaptive Admittance Controller.
2. Admittance Controller with Learnt ARMA parameters.
3. Mass Damper Admittance Controller with $M = 1$ and $D = 5$.
4. Mass Damper Admittance Controller with $M = 20$ and $D = 50$

4.6.2 Results

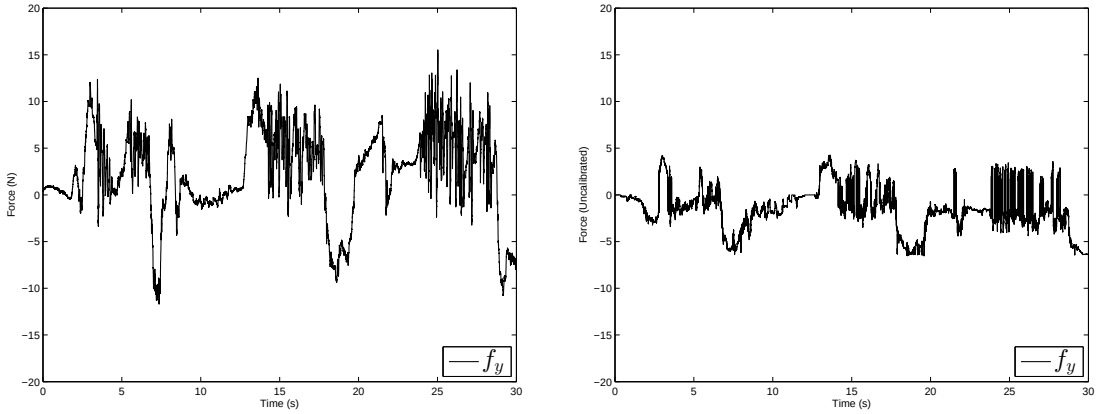
The results show that the readings of the force readings from the FlexiForce sensor are different from those from the calibrated Force/Torque sensor, see Figs. 4.18, 4.20, 4.22, and 4.24. The proposed adaptive admittance controller performs well as compared to using a fixed mass damper as seen in Figs. 4.17, 4.19, 4.21, and 4.23.



(a) Cartesian Trajectory Tracking (b) Flexiforce Output

Figure 4.17. ARMA with adaptive weights.

Figure 4.17 depicts the experimental results for the controller with the adaptive ARMA filter. This figure clearly shows that the controller makes the model trajectory x_m follow the task reference model x_d. Some oscillations are seen at the end points of the trajectory, this is caused due to the model switching at those points. From figures 4.17b and 4.18 the differences in the measured force from the FlexiForce sensors under the two different materials is seen. Although the force measured by the sensor is different the adaptive controller learns the different models per sensor.



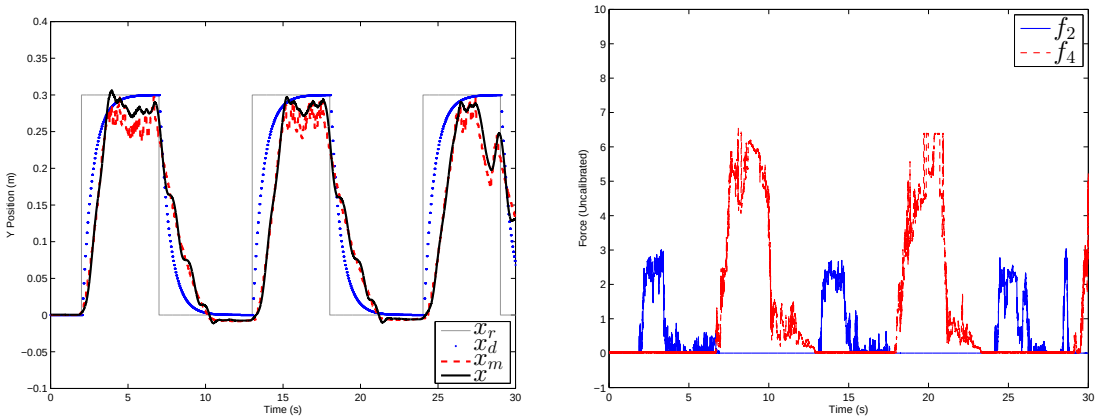
(a) FT Sensor output (b) Combined Flexiforce Output

Figure 4.18. Force output for ARMA with adaptive weights.

The ARMA weights obtained from the first experiment seen in figure 4.17 was used as the fixed weights in the second experiment as seen in figure 4.19. They were

$$flex_2 = [-0.6283 - 0.37570.1192 - 0.15120.0063 - 0.0017 - 0.00300.0026] \quad (4.50)$$

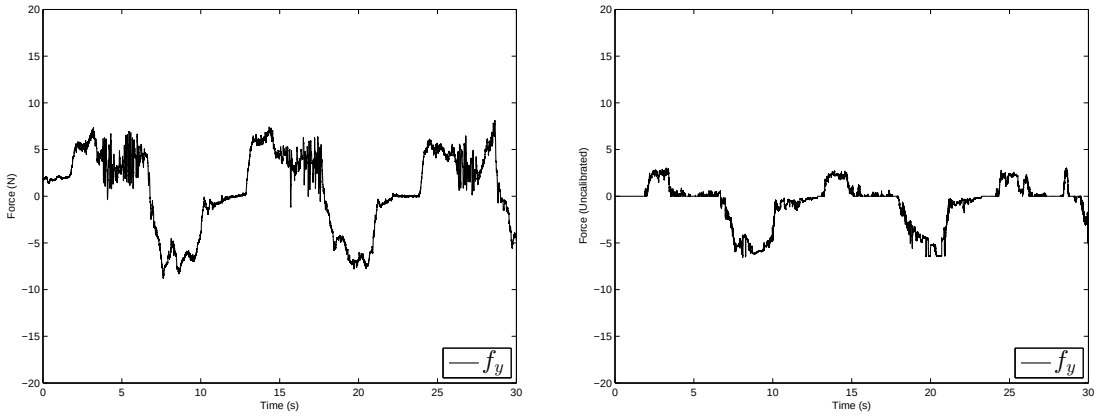
$$flex_4 = [-0.71850.1265 - 0.1395 - 0.2230 - 0.00090.0003 - 0.00010.0013] \quad (4.51)$$



(a) Cartesian Trajectory Tracking (b) Flexiforce Output

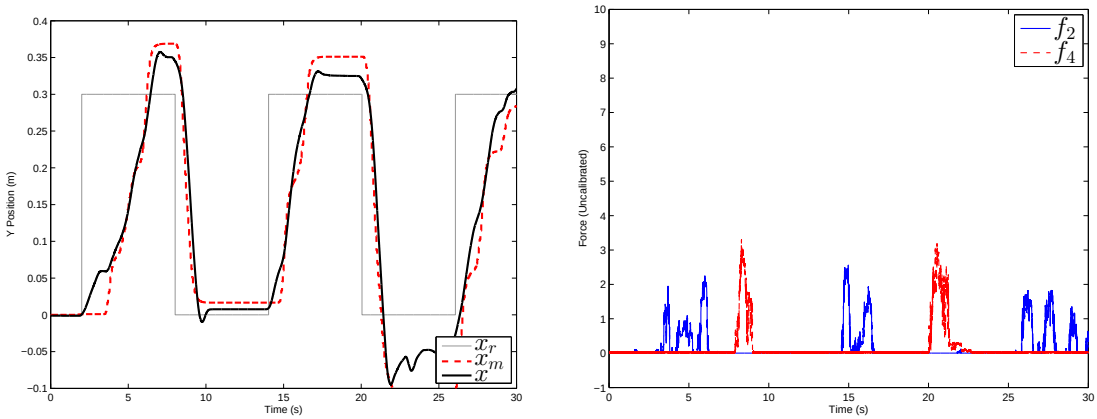
Figure 4.19. ARMA with fixed weights.

Figure 4.19 depicts the experimental results for the controller with the fixed ARMA filter weights 4.50 and 4.51. As in the previous experiment this figure clearly shows that the controller makes the model trajectory x_m follow the task reference model x_d. Since the model is already learnt and no further learning takes place the model switching does not cause as much oscillations. As in the previous experiment figures 4.19b and 4.20 clearly depict the vast differences in the measured force from the sensors.



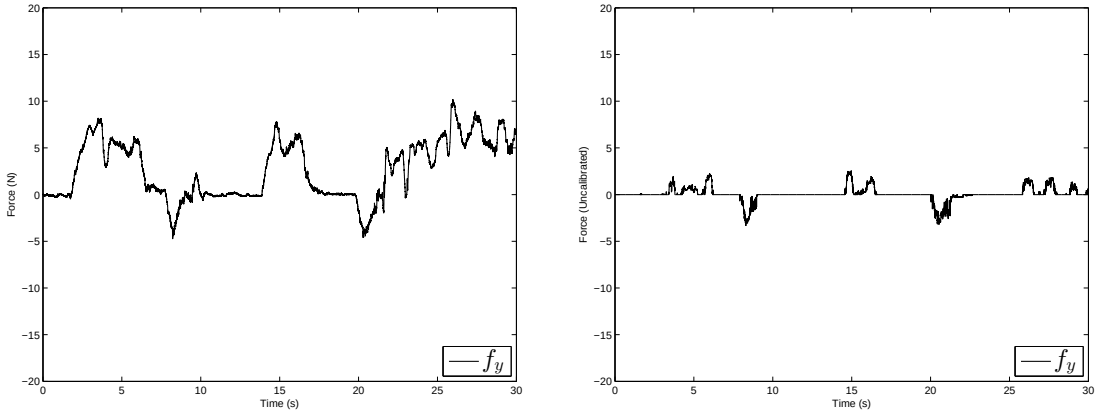
(a) FT Sensor output (b) Combined Flexiforce Output

Figure 4.20. Force output for ARMA with fixed weights.



(a) Cartesian Trajectory Tracking (b) Flexiforce Output

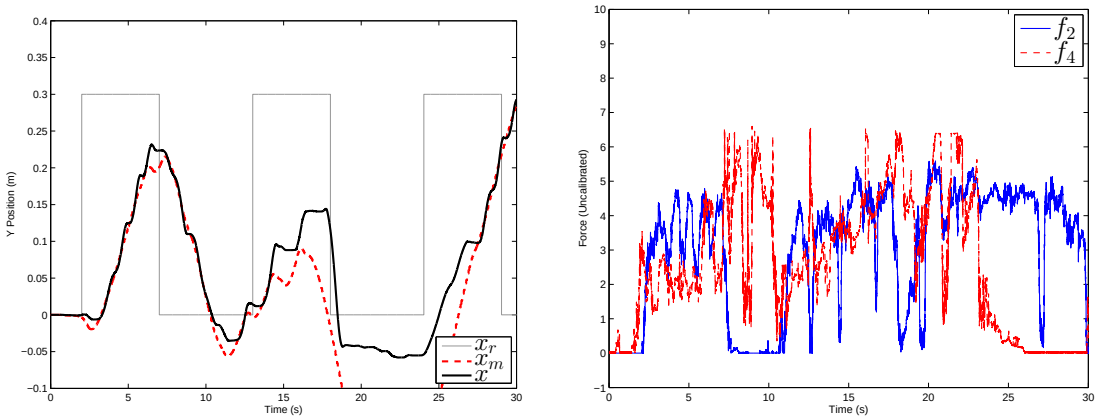
Figure 4.21. Fixed Mass Damper with $M = 1$ and $D = 5$.



(a) FT Sensor output (b) Combined Flexiforce Output

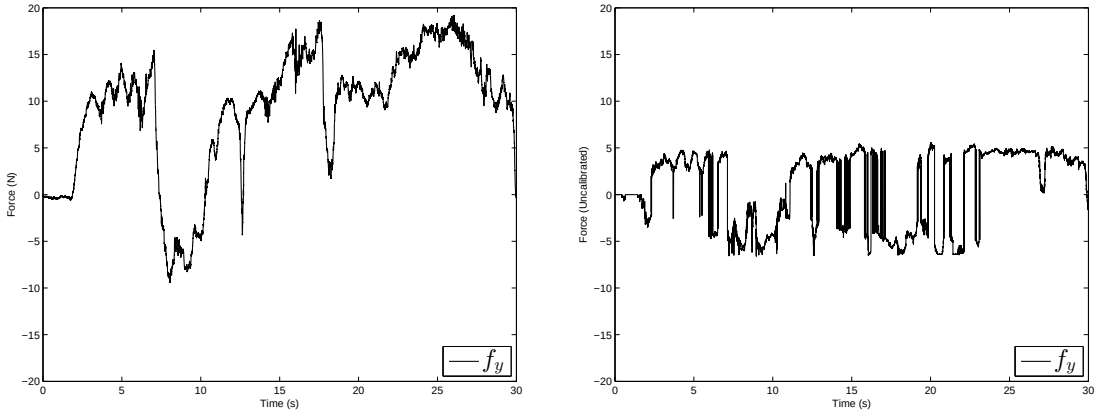
Figure 4.22. Force output for fixed Mass Damper with $M = 1$ and $D = 5$.

Figures 4.21 and 4.23 show the results of using the uncalibrated FlexiForce sensors with two different admittance models. Unlike in the first two experiments this results in significantly degraded performance as seen in figures 4.21, 4.22, 4.23, and 4.24.



(a) Cartesian Trajectory Tracking (b) Flexiforce Output

Figure 4.23. Fixed Mass Damper with $M = 20$ and $D = 50$.



(a) FT Sensor output (b) Combined Flexiforce Output

Figure 4.24. Force output for fixed Mass Damper with $M = 20$ and $D = 50$.

4.6.3 Discussion

In this section the adaptive admittance controller formulated in the previous sections was tested using a simple tactile sensor setup. A four sensor device was used with different types of material covering the sensors. This setup generated uncalibrated force data. The ability of the controller to effectively utilize uncalibrated force sensors was demonstrated experimentally.

4.7 Application: Robot Waiter

In this section, we study the interaction between a human and a robotic waiter operating in a dynamic environment with low, variable lighting, in which reliability, repeatability, and speed are crucial. Therefore, we rely on physical co-manipulation to allow a human to guide the robot through the environment. In general, physical Human-Robot Interaction (pHRI) requires coordination and control of the forces being applied by the human onto the robot.

The problem consists of two parts:

1. safe, stable contact with the human, and
2. identifying the human intent.

During the physical interaction, the robot should not exert any large forces on the human and have relative low joint velocities. The robot's limbs should move in a smooth, predictable manner and no oscillations should occur. In this section, the problem of safe and stable contact is addressed through compliance control [37, 39].

In a task the robot was required to follow along with a cart. The task definition allowed us to restrict the problem space such that the human pulling force applied through the cart was the input and the output was position regulation of the robot base with respect to the cart. Such a configuration enabled a natural cart-following behavior by the robot, and allowed the human operator to lead the robot to desired locations.

Figure 4.25. Personal Robot 2 (PR2) manipulating a cart at the UTA Research Institute (UTARI) Living Laboratory.

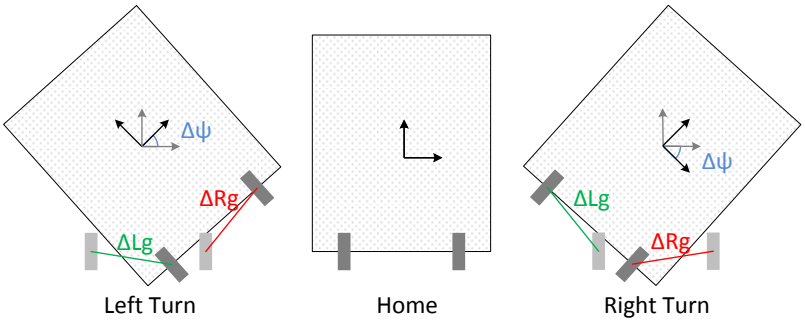


Figure 4.26. PR2 grippers location, and frames of the cart in top view.

4.7.1 Co-manipulation System

In this section, we describe the system used to follow human commands using a cart and the PR2 robot, consisting of a mobile base, two robotic arms, and a pan and tilt head unit (Fig. 4.25). Each arm is gravity compensated and offers 7 degrees of freedom (DOF). The arms extend up to 81 cm (32 inches) and the omnidirectional base can reach velocities up to 1.0 m/s (3.3 ft/s).

We first describe the controller used for compliant, stable interaction with the cart. Then, we describe the velocity based position controller that moves the robot to the home condition seen in Fig. 4.26, which leads to the cart-following behavior. A human will push/pull the cart or the robot, and the robot will initiate base and arm movements to follow guidance from the human.

4.7.1.1 Compliance Controller for the PR2 Arms

In classical explicit force control the objective is to maintain a desired interaction force f_d, using schemes such as proportional-integral-derivative controllers for regulation. This does not work in all interaction scenarios, especially not in cases where chattering occurs. It is well known that PID explicit force controllers can become unstable due to changes in contact environment [39, 68, 72, 73].

Impedance control was proposed by Hogan to achieve stable environmental contact [67]. It controls the dynamics of the physical interaction with the environment instead of achieving an explicit force objective. We use compliance control as a simplified version of this scheme [37, 39].

The general robot dynamic equation with actuator dynamics and external interaction with the environment is

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = \tau + \tau_e \tag{4.52}$$

where $q \in \mathbb{R}^n$ are the joint positions (n is the DOF), $M(q)$ is the inertia matrix, $V(q,\dot{q})$ is the Coriolis/centripetal vector, $G(q)$ is the gravity vector, and $F(\dot{q})$ is the friction term. We also add the disturbance torque $\tau_d \in \mathbb{R}^n$ and the environmental torque $\tau_e = J^T(q)f_e$, where f_e are interaction forces exerted on the robot in task space, $J^T(q)$ is the Jacobian and $\tau \in \mathbb{R}^n$ the control torque.

Neglecting friction and disturbance torques, the task space pose $x = \begin{pmatrix} p_e \\ \phi_e \end{pmatrix} \in \mathbb{R}^6$, where $p_e \in \mathbb{R}^3$ is the task space position and $\phi_e \in \mathbb{R}^3$ is the orientation. The dynamics in task space is

$$M_x(q)\ddot{x} + V_x(q,\dot{q})\dot{x} + G_x(q) = J^{-T}(q)\tau + f_e \tag{4.53}$$

where,

$M_x(q) = J^{-T}(q)M(q)J^{-1}(q)$,

$V_x(q,\dot{q}) = J^{-T}(q)V(q,\dot{q})J^{-1}(q) - M_x(q)\dot{J}(q)J^{-1}(q)$,

$G_x(q) = J^{-T}(q)G(q)$.

The desired impedance model to balance the interaction force f_e is chosen as

$$M_m(\ddot{x} - \ddot{x}_d) + D_m(\dot{x} - \dot{x}_d) + K_m(x - x_d) = f_e \tag{4.54}$$

where x_d is the desired trajectory, M_m is the inertia, D_m is the damping, and K_m is the stiffness of the impedance model.

Rearranging we get

$$\ddot{x} = \ddot{x}_d + M_m^{-1}(D_m(\dot{x}_d - \dot{x}) + K_m(x_d - x) + f_e) \tag{4.55}$$

Substituting (4.55) in (4.53) the control law is

$$\begin{aligned} \tau =& J^{-T}(q)(M_x(q)\ddot{x}_d + V_x(q,\dot{q})\dot{x} + G_x(q) \\ &+ M_x(q)M_m^{-1}(D_m(\dot{x}_d - \dot{x}) + K_m(x_d - x)) \\ &+ M_x(q)M_m^{-1}f_e - f_e) \end{aligned} \tag{4.56}$$

For the PR2 robot, the mass matrix and the Coriolis term are not readily available, while the gravity term can be neglected because of the robot's gravity compensated design. Therefore, replacing $M_m = M_x(q)$ in (4.54) and substituting into (4.53) the control torque leads to

$$\begin{aligned}\tau =& J^T(q)(M_x(q)\ddot{x}_d + V_x(q,\dot{q})\dot{x} + G_x(q) \\ &+ D_m(\dot{x}_d - \dot{x}) + K_m(x_d - x)).\end{aligned} \tag{4.57}$$

Assuming $\dot{x}_d = 0$ and for low joint velocities $\dot{q} \approx 0$ we obtain the compliance controller

$$\tau = J^T(q)(K_m(x_d - x) - D_m\dot{x} + G_x(q)). \tag{4.58}$$

Since the PR2 arms are gravity compensated, the gravity compensation term $G_x(q)$ can be removed. Replacing $\dot{x}$ with $J(q)\dot{q}$, the final desired torque is:

$$\tau = J^T(q)(K_m(x_d - x) - D_m J(q)\dot{q}). \tag{4.59}$$

This does not require interaction force measurements and avoids the need for an estimator to isolate the interaction forces from the PR2 gripper dynamics.

Because the controller in (4.59) only sets a desired gripper pose, a resorting torque

$$\tau_r = K_{P,r}\epsilon + K_{D,r}\dot{\epsilon} \tag{4.60}$$

was used to keep the arm joints close to their original home position q_d. The joint error ϵ is defined as the difference between the desired and current joint angles

$$\epsilon = q_d - q. \tag{4.61}$$

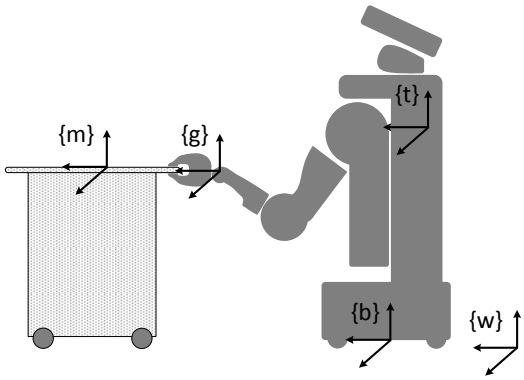


Figure 4.27. PR2 robot and cart transfer frames.

We consider the following reference frames as seen in Fig. 4.27: $\{w\}$ attached to the world, $\{b\}$ attached to the robot base, $\{t\}$ attached to the robot torso, $\{g_r\}$ attached to the right gripper, $\{g_l\}$ attached to the left gripper, and $\{m\}$ attached to the cart being used. The right and left gripper pose in task space is $x_r \in \mathbb{R}^6$ and $x_l \in \mathbb{R}^6$ in frame $\{t\}$. The pose of the robot base in frame $\{w\}$ is $x_b \in \mathbb{R}^6$. The generalized forces at the grippers are $f_r \in \mathbb{R}^6$ and $f_l \in \mathbb{R}^6$ in frames $\{g_r\}$ and $\{g_l\}$.

The cart, pulled by a human, will apply forces f_r and f_l on the grippers of the PR2. These interaction forces will cause changes in the task space states of the grippers $x_r = [r_x, r_y, r_z, r_\phi, r_\theta, r_\psi]$, $\dot{x}_r$, $\ddot{x}_r$, $x_l = [l_x, l_y, l_z, l_\phi, l_\theta, l_\psi]$, $\dot{x}_l$ and $\ddot{x}_l$ according to the dynamics of the defined impedance model (4.54) . The desired task space position of the compliance controller is set to some point in front of the PR2, whereas the reference velocity and acceleration are set to zero. When the cart is grasped, the grippers lose 3 DOF and the motion of the grippers will lie in the plane defined by the constraints due to the cart. The remaining DOF are translation in x, y and rotation about the z axis as seen in Fig. 4.26. The gains are set low in these directions for smooth interaction.

Let the robot Jacobian including the base motion (3DOF) be $J(q)$. Now the robot controller is

$$\tau = J^T(q) f_c + \left(I - J^T(q) J^{\dagger T}\right) \tau_r \tag{4.62}$$

where f_c is

$$f_c = \hat{W}^T \sigma(\hat{V}^T \varphi) + K_v r - v(t) - f_h \tag{4.63}$$

where f_h is the output of the force-torque sensor of the PR2.

4.7.2 Discussion

This type of system without adaptation was demonstrated at a live event called "Sky Ball XI", which is an annual fundraising event for US military veterans [135]. Specifically, the PR2 was used to serve wine to VIP guests seated at different tables. A cart with wine glasses and bottles had to be moved from table to table in a crowded, dynamic environment. The use of perception was limited due to uncontrolled lightning conditions that included many sources of potential interference with the robot's sensor. The proposed compliance and velocity controller performed well. Together with a human waiter, the PR2 successfully navigated through the crowded environment without any spills or other incidents.

4.8 Summary

In this chapter a new inner-loop/outer-loop controller is proposed to make pHRI robust, simple to implement, and more intuitive for humans to use. An inner-loop robot-specific controller was proposed which enables the user to interact with the robotic system so that it behaves like a prescribed robot impedance model. This for-

mulation allowed the nonlinear robot dynamics to behave like any prescribed admittance model. This tunable controller enabled the design of an outer-loop task-specific controller to take into account the human dynamics and adapt the prescribed robot admittance model for different users. Simulation results with a simulated human model and a robot model were presented to illustrate the robustness and stability properties of the controller. Finally, experimental results of the inner-loop and outer-loop controllers driving the interaction with the PR2 shows the feasibility of the proposed scheme.

Future work will include experiments of the outer-loop adaptation with a more diverse group of human subjects. The interaction forces will be measured using a full body robotic skin that is being developed. The effect of adapting to human abilities will be further expanded. The need for a reference signal to the task-specific model will be eliminated by using reinforcement learning techniques.

Future work will include testing with more human subjects, integrating different task models, testing new intent models, and performing more complex Cartesian tasks. Another extension of this work is the use of multi-sensory data from whole body robot skin for adaptive admittance control.

CHAPTER 5

Study of Human Motion

The motivation for studying human motion was driven by the need to specify $x_r(t)$ and human intent models that are representative to pHRI tasks. In addition, motion studies are also motivated by other HRI applications such as robotic therapy of Autism.

Autism is a neurological disorder that manifests as deficiencies in social, communication and learning. Individuals with Autism have difficulty understanding human intent and other relevant social interaction components. Understanding human motion can help design robots that can help in rehabilitation and human-robot interaction.

This chapter is organized as follows: Section 5.1 describes work conducted that uses Dynamic Time Warping (DTW) to compare motions and identify a motor marker for Autism. Then Section 5.2 describes a new system that can learn and generalize human motion using Dynamic Movement Primitives (DMPs). Finally, Section 5.3 provides a summary of the chapter.

5.1 Motion Analysis for Autism Diagnosis

Currently Autism diagnosis is mainly a subjective process with high irregularity and inconsistency. This lack of standard objective tests for autism prevents the use of consistent, continuous development tracking and therapy. Studies have shown that children with Autism respond positively to robot companions. This motivated

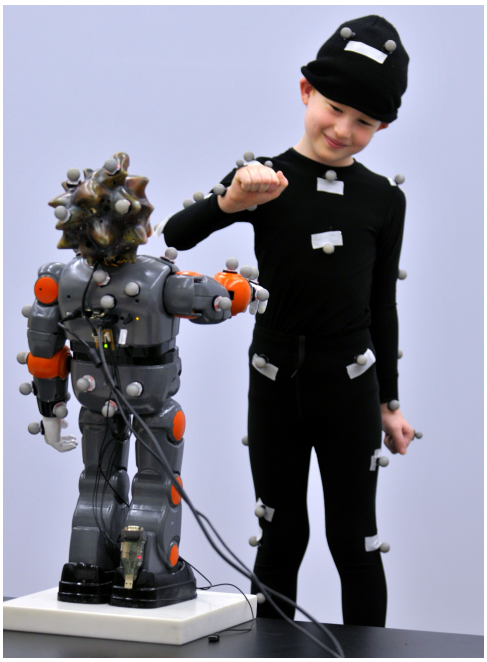


Figure 5.1. Zeno and Child during an Experiment - Photo credit: Fort Worth Star-Telegram/Max Faulkner.

the development of a humanoid social robot called Zeno for early Autism diagnosis, tracking and therapy [23].

According to Baer et al. [136] any behavior whose structure follows that of a model functionally and is close to the model temporally can be considered imitation. In this respect they are specific about what constitutes imitative behavior; the imitator should be following cues from the model. From this definition it is clear that imitation does not generate the exact behavior that is exhibited by a model. One of the points being that the structure or topography of the action is given more importance than the temporal similarity of the behavior. This means that we can assess the quality of imitation by ignoring some temporal differences in the observed behavior.

The contribution of this work, is the development of a robotic system which uses Dynamic Time Warping (DTW) as a tool for the diagnosis and treatment of Autism. DTW [110, 137] was used extensively in the speech processing community,

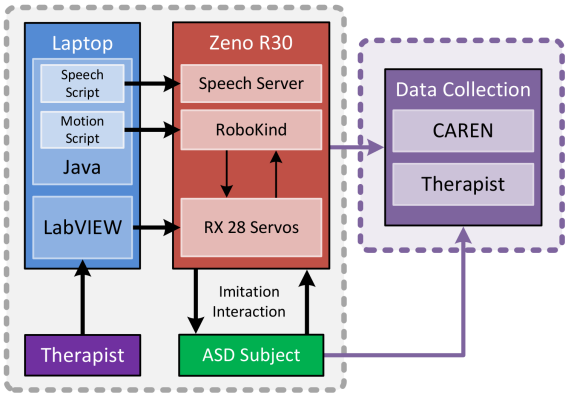


Figure 5.2. System diagram.

and is a dynamic programming algorithm, which gives a distance measure that is locally temporally invariant. It has also been used extensively by the data mining community recently [111]. Here, we use DTW as a similarity measure for comparing arm motions initiated by robot Zeno and imitated by children. Results show that the DTW similarity measure can serve as both a meaningful and objective measure for evaluating the HRI quality.

5.1.1 System Description

Project RoDiCA aims at developing a new motor cortex for the Zeno humanoid robot, as well as data collection and processing components to enable the early detection and treatment of ASD [23]. An overall system diagram of the HRI system including Zeno, the therapist, child and the associated data collection environments is shown in Fig. 5.2.

5.1.1.1 Humanoid robot

Zeno is a 2 foot tall articulated humanoid robot with an expressive human like face shown in Fig. 5.1. It has 9 degrees of freedom (DOF) in the upper body and

arms, an expressive face with 8 DOF, and a rigid lower body [23, 24]. The robot is capable of moving the upper body using a waist joint, and four joints each on the arms implemented using Dynamixel RX-28 servos. It has a 1.6 GHz Intel Atom Z530 processor onboard and is controlled by an external Dell XPS quad core laptop running LabVIEW [23].

Two different modes of interaction are implemented on Zeno. The first is called 'Dynamic Interaction,' which uses a Kinect sensor and LabVIEW to allow full teleoperation control of the arms and waist DOFs. This allows a therapist to interact with the child through the robot, or allows the child to control the robot directly. The second mode called 'Scripted Interaction,' is based on the Zeno RoboKind software which allows preprogrammed motions and conversations using text to speech software. In this section we look at the data captured during both scripted and dynamic interaction modes where the child imitates Zeno's motion.

5.1.1.2 Data collection and testing of human subjects

The data collection is performed using a motion capture system from Motion Analysis Corp, Santa Ana, CA, and CAREN, a virtual reality system from Motek Medical BV described in detail in previous work [23]. This system consists of cameras that capture motion at 120 Hz. In this work the human subject and the robot Zeno are both instrumented using 40-50 reflective markers.

For the present pilot study interaction experiments were conducted with Zeno facing the subject as seen in Fig. 5.1. A scripted routine of gestures was run on Zeno, which the child was prompted to imitate using Zeno's speech functionality. The following 42 second scripted routine from [23] was run on Zeno:

1 Wave hello with right arm [8.3s to 15.8s]

2 "Tummy rub" with right arm [16.8s to 25.3s]

3 Fist bump right with arm [25.7s to 30.3s]

4 Fist bump left with arm [31.0s to 34.5s]

5 Wave goodbye with left arm [34.5s to 42.0s]

For this phase of the analysis, we only used the time series of the joint angles of Zeno's and the subject's motion during a right handed wave. Similarity in motion between child and robot is measured using the joint angles, since Cartesian positions are more difficult to compare due to dimensional and pose differences e.g. robot may be much smaller than child, and rotated in space to face each other.

5.1.2 Dynamic Time Warping (DTW) as a Motion Marker

In this section, we describe how the data is processed, describe the DTW algorithm and propose the development of DTW as a tool to identify a motor marker for ASD.

5.1.2.1 Data representation and Inverse Kinematics

In this section, we record the imitative behavior using the four joint angles of the arm as depicted in Fig. 5.3. The angles are α and β in the shoulder joint and γ and θ in the elbow joint of Zeno. The data from the motion capture system is in the form of Cartesian joint positions of the shoulder $P_s = (x_s, y_s, z_s)$, elbow $P_e = (x_e, y_e, z_e)$ and the hand $P_h = (x_h, y_h, z_h)$ respectively. The joint angles are then calculated from these positions using the trigonometric equations below.

We define n_c the surface normal to the plane containing P_s, P_e and P_h, $n_c = V_{se} \times V_{eh}$, V_{se} is the vector from P_s to P_e, V_{eh} the vector from P_e to P_h, V_{eh} the vector from P_e to P_h and V_{es} the vector from P_e to P_s.

The following equations are used to obtain the joint angles:

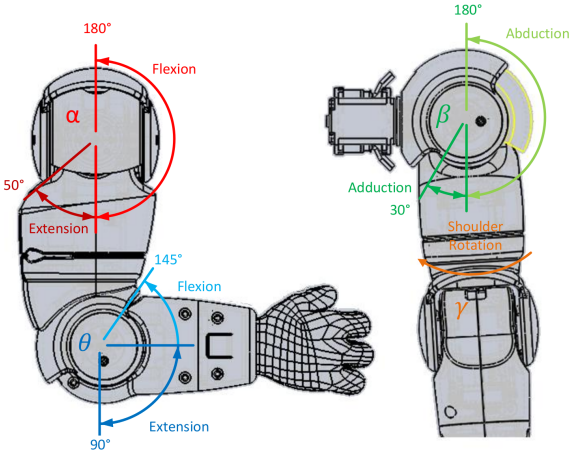


Figure 5.3. Zeno's Arm Angles.

$$\alpha = tan^{-1}\left(\frac{V_{se}(3)}{V_{se}(2)}\right), \tag{5.1}$$

$$\beta = cos^{-1}\left(\frac{V_{se}(1)}{|V_{se}|}\right), \tag{5.2}$$

$$\theta = cos^{-1}\left(\frac{V_{eh} \cdot V_{es}}{|V_{eh}||V_{es}|}\right), \tag{5.3}$$

Where $V_{se}(1)$, $V_{se}(2)$ and $V_{se}(3)$ are the x, y and z components of V_{se} respectively, and n_i is defined as:

$$n_i = R_{\alpha\beta}\begin{bmatrix}0\\0\\1\end{bmatrix} \tag{5.4}$$

$$R_{\alpha\beta} = \begin{bmatrix} cos(\beta) & -sin(\beta) & 0 \\ cos(\alpha)sin(\beta) & cos(\alpha)cos(\beta) & -sin(\alpha) \\ sin(\alpha)sin(\beta) & sin(\alpha)cos(\beta) & cos(\alpha) \end{bmatrix} \tag{5.5}$$

Since the imitation between Zeno and the human is mirrored, Zenos right arm is compared to each subjects left arm angle trajectories. This data is then preprocessed by z-normalization, suggested by various researchers as an important step to be performed before running any tests [111]. The z-normalization removes offsets and scaling issues in the data, see following equation:

$$\vec{s_z} = \frac{\vec{s} - \mu_s}{\sigma_s} \tag{5.6}$$

where μ_s and σ_s are the mean and standard deviation of the signal $\vec{s}$.

5.1.2.2 Motion Analysis using Dynamic Time Warping

There is a need in the Autism research community to obtain quantitative measurements of imitation quality. We propose using the DTW algorithm to obtain a similarity measure between time series joint angle signals. The strength of DTW is in its ability to compare the similarity between signals by ignoring time-delays and uneven time sampling. This situation is very relevant in the context of our problem, since the motion of child and robot experiences both these effects.

Given two signals that are time dependent $\vec{X} = \{x_1, x_2...x_n\}$ and $\vec{Y} = \{y_1, y_2...y_n\}$ where $n \in \mathbb{N}$. DTW finds the optimal distance between these two signals using a local distance measure d, we use the euclidean distance as the distance measure $d(\vec{X}, \vec{Y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$.

The DTW cost $\mathfrak{D}(n, n)$ can be calculated using the following recursion:

$$\begin{aligned} \mathfrak{D}(i,j) = d(s_i, r_j) + min\{&\mathfrak{D}(i-1, j-1), \\ &\mathfrak{D}(i-1, j), \\ &\mathfrak{D}(i, j-1)\} \end{aligned} \tag{5.7}$$

DTW is applied to each set of angle trajectories generating a DTW value for each of the four angles. A range of motion is then obtained in order to weigh each angle using equation 5.8, where $\vec{X}$ represents a column vector of unnormalized joint angles:

$$W = max(\vec{X}) - min(\vec{X}). \tag{5.8}$$

The combined DTW distance for all four angles is calculated by a weighted average as shown in 5.9.

$$A_w = \frac{W_\alpha D_\alpha + W_\beta D_\beta + W_\gamma D_\gamma + W_\theta D_\theta}{W_\alpha + W_\beta + W_\gamma + W_\theta}, \tag{5.9}$$

where,

W_* - weight per joint

D_* - calculated DTW distance for each joint

The pseudo-code used for implementing our version of DTW is shown in Algorithm 1.

Algorithm 1 DTW Cost

1: **procedure** DTWCOST(S,R)
2: $m \leftarrow$row(S)
3: $n \leftarrow$row(R)
4: $\mathfrak{D} \leftarrow \text{zeros}(n+1, m+1)$
5: $\mathfrak{C} \leftarrow 0$
6: $\mathfrak{D}(1,1) \leftarrow 0$
7: **for** $i \leftarrow 2, m$ **do**
8: $\mathfrak{D}(i,1) \leftarrow \mathfrak{D}(i\text{-}1,1) + \text{d}(\text{S}(i),\text{R}(1))$
9: **end for**
10: **for** $j \leftarrow 2, n$ **do**
11: $\mathfrak{D}(1,j) \leftarrow \mathfrak{D}(1,j\text{-}1) + \text{d}(\text{S}(1),\text{R}(j))$
12: **end for**
13: **for** $i \leftarrow 2, n$ **do**
14: **for** $j \leftarrow 2, n$ **do**
15: $\mathfrak{C} \leftarrow \text{d}(\text{S}(i),\text{R}(j))$
16: $\delta \leftarrow \min(\mathfrak{D}(i\text{-}1,j),\mathfrak{D}(i,j\text{-}1),\mathfrak{D}(i\text{-}1,j\text{-}1))$
17: $\mathfrak{D}(i,j) \leftarrow \mathfrak{C} + \delta$
18: **end for**
19: **end for**
20: **return** $\mathfrak{D}(m,n)$
21: **end procedure**

5.1.3 Results

A set of clinical experiments with several children were performed to test the robot, capture the motion, and to compare the motion data using DTW. During experiments, children were directed by the robot to follow along performing several hand gestures, such as wave with both hands, tummy rub, fist bump, etc. Each motion was performed three times. The best one was picked for analysis.

Table 5.1. DTW distance per joint

Subject	**Age**	**Gender**	**Type**	α	β	γ	θ	β θ **Combined**
3	6	Male	Control	0.4421	0.6730	0.2775	0.1145	0.3075
2	6	Male	ASD	1.2128	1.2128	1.1359	0.7232	0.8913
12	9	Male	Control	0.6436	0.0792	0.5253	0.3073	0.2247
4	9	Male	ASD	0.3387	0.6738	0.6899	0.9251	0.7331
13	11	Male	Control	0.6164	0.2267	0.6409	0.2854	0.2666
6	11	Male	ASD	0.4473	0.1881	0.4319	0.3228	0.2757
11	12	Male	Control	1.1141	0.2583	0.5157	0.1159	0.1636
7	12	Male	ASD	0.2350	0.6023	0.4192	0.2091	0.4102

Wave gestures were compared for four different age groups (6, 9, 11, 12) as seen in Table 5.1, they consist of pairs of control and ASD children recruted from the Dallas Autism Treatment Center. Each age group consists of a control subject and an ASD subject. Representative joint angle trajectories for each DOF are shown in Fig. 5.4. The angle trajectories of subject 12 gives insight into the significance of each joint in performing each action. Here the elbow angle θ has the best match during the imitation, this is because the elbow performs a periodic motion when executing a wave, so it is easy to script and imitate. β which is the other main angle used in

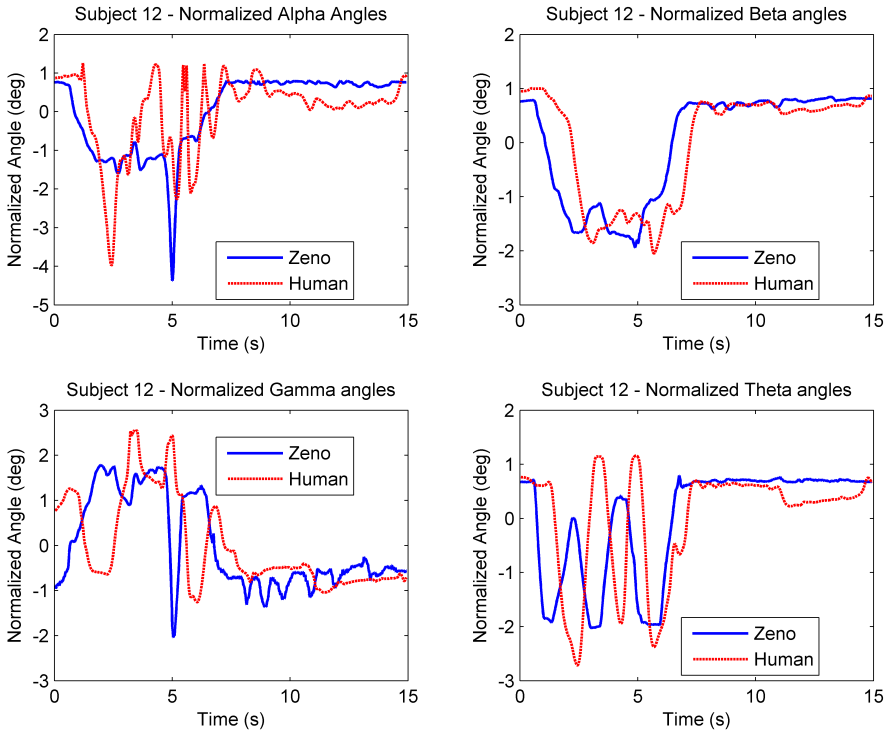


Figure 5.4. Normalized Joint Angles.

a hand wave is also fairly similar. The difference seen in Fig. 5.4 for the normalized angles varies more in the human trajectory because it is difficult for humans to keep their shoulder in the exact place over a period of time like the robot. The computation of the DTW for the different subjects performing the wave motion is shown in Table 5.1 and summarized in Fig. 5.5. The weights for α and γ in Equation 5.9 are set to zero to calculate a weighted average using only β and θ, this is because β and θ angles are more representative for the motions performed in this study.

By looking at this data we can see that the combined average for the control subjects are all lower than for the ASD subjects. This shows that DTW in combination with weighting based on range of motion for the most important angles, β and θ, for a hand wave shows a promising method of comparing human to robot imitation.

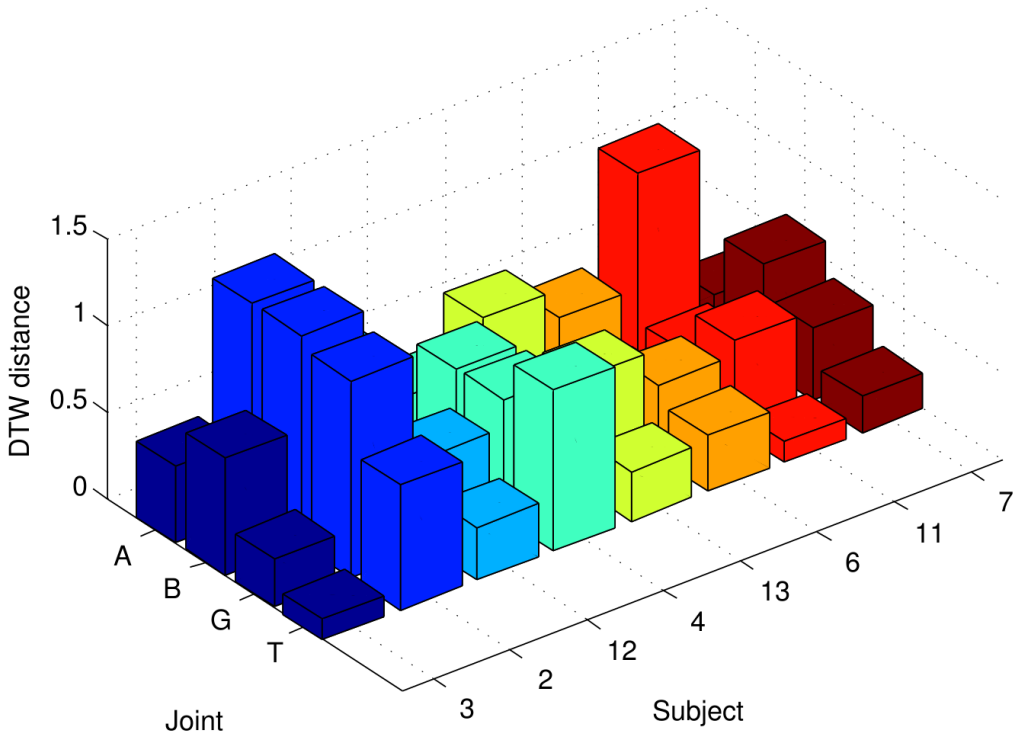


Figure 5.5. DTW values.

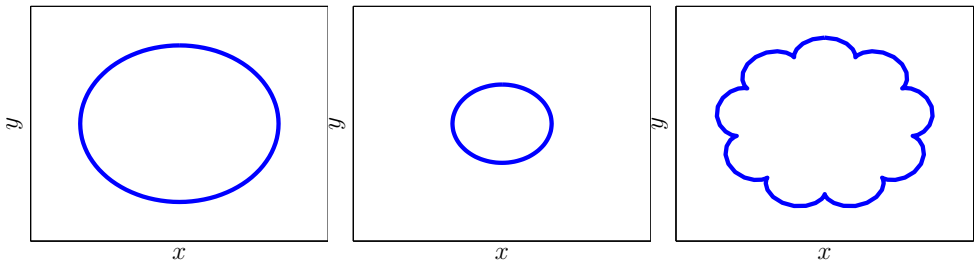
Also, notice by looking at Table 5.1 the DTW values for β and θ also are consistent for each age group with the exception of β for the group of eleven year olds. This discrepancy is simple due to a better imitation by the ASD subject for β, but as mentioned before β will always have variation due to human nature. However, this inconsistency helps show that the value of the overall performance is not affected by one angle, but instead provides a consistent value based on both angles used. This demonstrates that DTW is a good method to use in motion comparison analysis.

These preliminary results show that the DTW similarity measure can serve as both a meaningful and objective measure for evaluating the quality of imitation behavior. With age all children improve their imitation behavior, but the children with ASD consistently perform worse than their age-match controls. The combined

weighted joint average for the wave motion has a higher value in all ASD children, indicating that their imitation of Zenos waving motion is less accurate.

5.2 Human Motion Generalization using Dynamic Movement Primitives

In this work we are interested in changing fundamental properties of gestures without changing their overall structure and meaning. To achieve this efficient encoding of gestures is needed. Signals are traditionally defined by duration δ, amplitude α, and frequency ν. This basic description of signals form the basic structure of motion as well. To adapt motion the three components have to be tunable.



(a) A = 1.0, f=1 Hz (b) A = 0.5, f=1 Hz (c) f=1 Hz + 10 Hz

Figure 5.6. Change in amplitude and frequency.

5.2.1 Dynamic Movement Primitives (DMP)

Dynamic Movement Primitives (DMP) have emerged as a tool to encode robot motions as dynamical systems. These systems are desirable due to the robustness properties of nonlinear systems. DMPs have been used successfully for force dependant tasks [138].

The DMP method was first introduced by Ijspeert et al. [117]. The DMP system consists of three parts the canonical system, the modulation function, and a stable

converging dynamic system. As described in [119], the modified version of the DMP equations are used

$$\tau\ddot{x}(t_f) = K(x(t_f) - x(t)) - D\dot{x} + (x(t_f) - x(0))f(s) \tag{5.10}$$

where $\ddot{x}(t_f)$, $\dot{x}(t_f)$, and $x(t_f)$ are the acceleration, velocity, and position respectively. K is the spring constant, D is the damping coefficient, τ is a time scaling factor. $f(s)$ is a nonlinear function called the modulation function and is defined as

$$f(s) = \frac{\sum_i \psi_i(s) w_i}{\sum_i \psi_i(s)} s \tag{5.11}$$

where $\psi_i(s)$ are the bases functions of $f(s)$. The canonical function which generates the input s of (5.11) is

$$\tau\dot{s} = -\alpha s \tag{5.12}$$

where α determines the structure of s, the input to the modulation function.

Since DMPs enable encoding nonlinear motion data, they can be used to predict the motion of a human given a sufficiently large database of motion patterns. A local distance measure such as DTW can be used to predict the motion pattern of the user. The distance measure should be low in computation cost and online.

5.2.2 Trajectory Adaptation for Robot Differential Teaching

In this section the metrics used to compare the duration δ, amplitude α, and frequency ν are defined. The design utilizes the properties of the DMP framework to define scalable parameters for human motion. The following sections illustrate the different properties exhibited by the system.

5.2.2.1 Duration adaptation

The duration of a gesture can be obtained easily by recording the start time t_s and finish time t_f of the gesture. Then the duration is $\delta = t_f - t_s$. This parameter

can be compared with the duration of the reference gesture δ_r to obtain the duration difference $\Delta\delta = \delta - \delta_r$. This is used in the adaptive mechanism to determine duration of the DMP gesture.

$$\tau = \tau + \xi_\delta \Delta\delta \tag{5.13}$$

where ξ_δ is a fraction of the difference between the ideal motion and the human motion.

5.2.2.2 Amplitude adaptation

The amplitude of a gesture can be obtained by resampling the gestures and calculating the 2-norm. Then the amplitude difference is α. This parameter can be compared with the amplitude of the reference gesture α_r to obtain the duration difference $\Delta\alpha = \alpha - \alpha_r$. This is used in the adaptive mechanism to determine duration of the DMP gesture.

$$A = A + \xi_\alpha \Delta\alpha \tag{5.14}$$

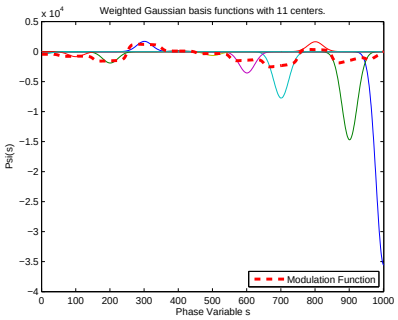
where ξ_α is a fraction of the difference between the ideal motion and the human motion.

5.2.2.3 Frequency adaptation

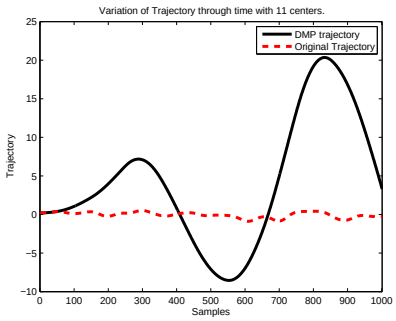
Given the frequency of a gesture ν. This parameter can be compared with the frequency of the reference gesture ν_r to obtain the frequency difference $\Delta\nu = \nu - \nu_r$. This is used in the adaptive mechanism to determine frequency of the DMP gesture.

$$f = f + \xi_\nu \Delta\nu \tag{5.15}$$

where ξ_ν is a fraction of the difference between the ideal motion and the human motion.

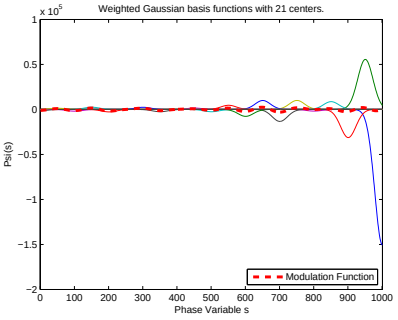


(a) Learnt modulation function.

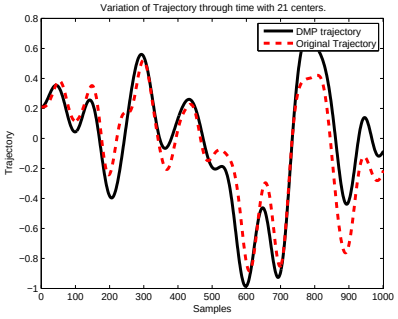


(b) Learnt motion.

Figure 5.7. DMP with 11 basis functions.

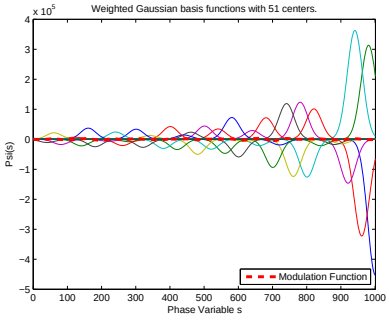


(a) Learnt modulation function.

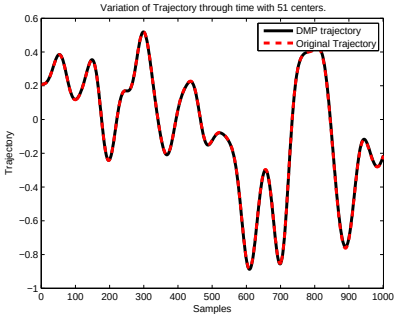


(b) Learnt motion.

Figure 5.8. DMP with 21 basis functions.



(a) Learnt modulation function.



(b) Learnt motion.

Figure 5.9. DMP with 51 basis functions.

5.3 Summary

The contribution of this chapter is the extension of the adaptive admittance control methods proposed to utilize DMP to encode $x_r(t)$. Another contribution is the development of a robotic system which uses Dynamic Time Warping (DTW) as a tool for the diagnosis and treatment of Autism. This chapter describes the development of a robotic system capable of measuring the quality of imitation interaction between a humanoid robot and a human subject. A humanoid robot called Zeno, is used to perform gestures which the human subject imitates. The similarity of the gestures performed is measured using DTW, and used to objectively analyze the quality of the imitation interaction between the human and the robot. The hope is that this type of system will enable consistent objective measurement of the quality of imitation, and can be used to obtain information about the condition and possible improvement of children with ASD.

From the graphs of the wave motion it is clear that the robot motion should be derived from nominal human motion trajectories. The initial data analysis shows that DTW can be a good tool for comparing imitation interaction since it allows the comparison of temporally inexact imitative motion. However, it is not clear what robot motions to select, and what DOFs to compare using DTW in order to best affect ASD treatment.

CHAPTER 6

Software Developed

6.1 SkinSim

SkinSim is a multi-modal skin simulation for Gazebo.

Figure 6.1. SkinSim Logo.

As more and more robots inhabit our living space physical contact between humans and robots become inevitable and even desirable, especially for cooperative tasks. Assuring safe robot operation around humans has become a major area of research in Physical Human-Robot Interaction (pHRI). The change from structured industrial human living spaces with significant uncertainties, has made the development of multimodal robotic skin a priority [5].

Since robot hardware is expensive and hard to maintain, realistic simulation can become a major tool to support research and development of robotic systems. This will add both flexibility and reproducibility to experiments. In recent years various robotic simulators have been introduced including Microsoft Robotic Studio [139], Webots [140], MORSE [141], Stage [142], OpenHRP3 [143], V-REP [144], OpenGRASP [145], and Gazebo [146]. Gazebo has achieved increased adoption by the robotics community and was the government furnished simulator for DARPAs Virtual Robotics Challenge (VRC) [147].

Our simulation environment, SkinSim is a new and unique suite of software that address challenges of design, integration and use of multi-modal robot skin and garments as physical interfaces between robots and humans. Built on top of ROS/Gazebo, SkinSim provides a simulation platform for designing robust and safe robot behaviors based on sensor feedback. This can be used to test methods for optimal placement, data management and selection of sensors. In addition to pressure sensors, SkinSim supports other types of distributed MEMS skin sensors including thermal, accelerometer, and IR sensors.

6.1.1 System Design

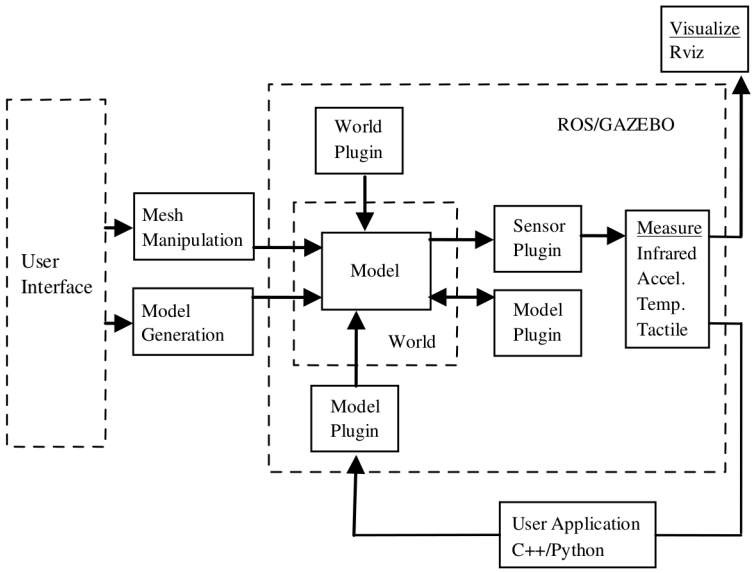


Figure 6.2. Data flow diagram of SkinSim.

SkinSim is based on the Gazebo robot simulation software. It uses the ODE physics engine [120] for dynamic simulation and also includes contact detection to determine interaction forces. It also has support for Bullet [121], Simbody [122],

and DART [123] physics which include soft body simulation capabilities. A block diagram of SkinSim is shown in Fig. 6.2. It represents the steps involved from identifying a robot to generating a fully sensorized simulation model with controller plugins. SkinSim consists of three modules: mesh manipulation, model generation, and user interface (UI). The mesh manipulation module extracts robot meshes and identifies poses for sensor placement using the Point Cloud library (PCL). The skin model generation module assembles skin sensor patches given skin density, patch size, skin model structure, tactile model structure, and tactile data communication design as inputs. The skin patches can be generated in two ways: as standalone Simulation Description Format (SDF) files with embedded model and sensor plugins or as a modification of an existing robot Unified Robot Description Format (URDF)/SDF model using the mesh manipulation package. SkinSim modules are designed to run independently of a user interface for batch execution and optimization of pHRI. A helpful wizard called Robot Tailor serves as a front end UI for the mesh manipulation and model generation modules (Fig. 6.3). The Robot Tailor is a UI for designing smart multi-modal skin based garments or coverings for robots. This was motivated by the planning configuration description wizard of MoveIt! [148].

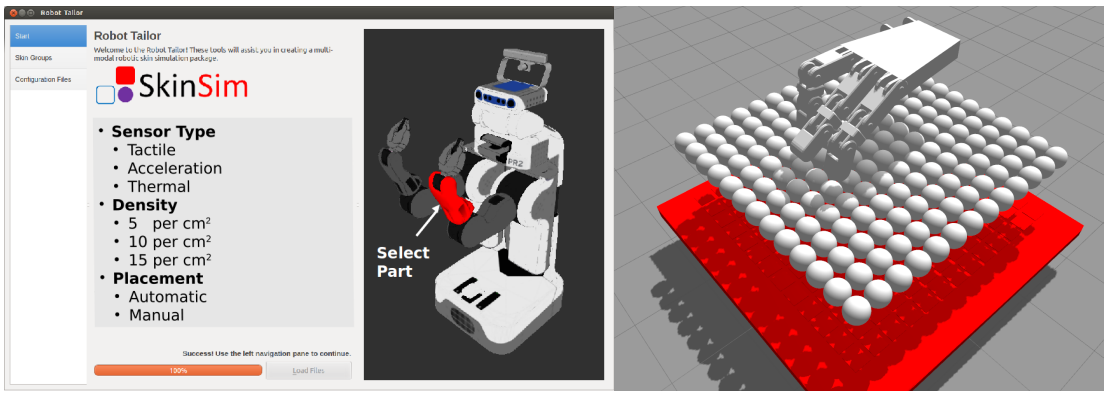
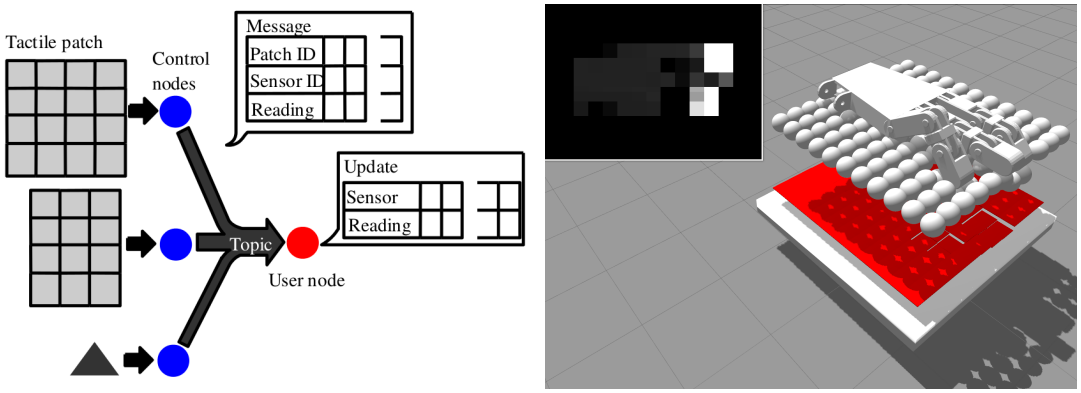


Figure 6.3. SkinSim user interface and tactile sensor array.

Fig. 6.3 shows the Robot Tailor UI and a typical robot skin patch generated by it. Users can control the robot via model plugins and can initiate interactions with it. During the interaction, data is collected from each sensor via a sensor plugin or model plugin and made available to the user. The user is able to visualize the data via the ROS Robot Visualizer (Rviz) [149].

6.1.2 Data Management and Visualization



(a) ROS topics handle sensor data fusion and data management

(b) Tactile image of a robot hand

Figure 6.4. Data Management and Visualization.

Each tactile sensor provides contact information including the sensor ID, simulation time, and sensed force or any other output depending on the type of the sensor. A node is assigned to each tactile array patch to collect this data, see Fig. 6.4a. Various data encoding and transmission methods can be implemented in this node to reflect those employed in real sensors. For example, one scheme could publish data a row at a time for a given tactile array as in [150]. Another could publish the entire tactile data at a given rate as employed in Tekscan [151]. Users will be able to regulate the rate at which data is published.

Each n by m tactile sensor patch encodes the force reading into an n by m 8 bit monochrome image. Each pixel of this image can have values ranging from 0 to 255 representing the normalized force sensed by individual sensors. RViz is used to view the tactile image. Fig. 6.4b depicts a tactile image created by pushing a robot hand against a sensorized robot skin in SkinSim.

SkinSim is still in the development stage. It is available at

www.bitbucket.org/nextgensystems/skinsim

and has received some interest from the robotics community.

Future simulator development plans will involve the design and integration of new sensor models such as infrared sensors, thermal sensors, and accelerometers. We believe that this multi-modal robot skin model with tactile sensing, infrared sensing, temperature sensing and more generally complete simulation of human robot interaction will be of great benefit to the robotics community. This system could be used to perform regression testing to determine optimal sensor design and placement criteria. Such a system will also be a useful tool for testing out new robotic algorithms for pHRI.

6.2 SkinLearn

SkinLearn, is a multi-modal skin based Human-Robot Interaction learning, estimation, and control system. The work in adaptive physical human-robot interaction, intent estimation, and tactile awareness will be implemented as libraries in SkinLearn.

Figure 6.5. SkinLearn Logo.

6.2.1 Intelligent Control and Estimation Library (ICE)

The Intelligent Control and Estimation Library (ICE) will form the basis of the SkinLearn framework. It will be based on Dr. Lewis's Optimal Estimation, Optimal Control, Robot Manipulator Control, and Neural Network controller books.

It will contain: basic PID, adaptive, robust, LQR, LQG, Neural Network based, and reinforcement learning based controllers. This will also include filtering algorithms and will contain: low pass/high pass filters, alpha beta filters, Kalman filters, Extended Kalman Filters, Unscented Kalman filters, Particle filters. This will provide tested base functionality as C++ classes to be used by other applications as object instances.

SkinLearn is still in the development stage. It will be available at

www.bitbucket.org/nextgensystems/skinlearn

in the near future.

CHAPTER 7

Conclusion and Future Work

7.1 Conclusion

To enable close human-robot interaction and co-operation controllers motivated by human factors studies and the human neurophysiological system are developed. A two-loop control architecture which overcomes the challenges associated with traditional trajectory based controllers and allow the human to be included in the human-robot team is proposed. First, an inner-loop is used to control the usually nonlinear dynamics of a robot. This was based on neuroadaptive inverse dynamics based control. Then, an outer-loop is used to make the robot system combined with the human operator behave like a prescribed task model. This structure decouples the design of the controllers and removes the requirement for trajectory following.

7.1.1 Neuroadaptive Inverse Dynamic Controller

An inner-loop robot controller that causes a robot to behave like a prescribed admittance model was developed. No task information is needed by this controller. This is in contrast to most standard work in robot impedance and neural network control which is based on trajectory following [40, 41, 67]. Since neural network function approximation techniques [40] are used, this is called model reference neuroadaptive control.

7.1.2 Adaptive Admittance Controller

An outer-loop task-specific controller was developed. This adapts the parameters of a prescribed robot admittance model so that the robot system assists the human to achieve task-specific objectives. This separation of robot-specific design and task-specific design follows human factors studies [60,61] and is facilitated by the fact that the admittance model parameters appear nowhere in the control design of the inner-loop neuroadaptive admittance controller.

7.1.3 Human Motion Study

Motivated by research in robot assisted Autism therapy. Methods were proposed for measuring human motion similarity using Dynamic Time Warping (DTW) and generalizing human motion was proposed based on Dynamic Movement Primitives (DMP). These methods will enable robots to better understand the human motion, intention, and react accordingly.

7.1.4 Software Developed

Tools to test PHRI control methods - SkinSim and SkinLearn were described. SkinSim, a new simulator framework for multi-modal, multi-resolution robot skin, aimed at solving complex design problems was proposed. SkinSim is implemented using the ROS and Gazebo simulation infrastructures which are supported by the Open Source Robotics Foundation, and therefore will be shared with the community. SkinLearn, is a multi-modal skin based Human-Robot Interaction learning, estimation, and control system. The work in adaptive physical human-robot interaction, intent estimation, and tactile awareness will be implemented as libraries in SkinLearn.

7.2 Future Work

7.2.1 Neuroadaptive Inverse Dynamic Controller

Future work will involve further testing the neuroadaptive controllers with different weight update laws, neural network size, and activation functions.

7.2.2 Adaptive Admittance Controller

Future work will include testing with more human subjects, integrating different task models, testing new intent models, and performing more complex Cartesian tasks. Another extension of this work is the use of multi-sensory data from whole body robot skin for adaptive admittance control. The work on adaptive admittance will be further expanded to include multi-contact scenarios, high density tactile elements. This work will also be implemented and tested on custom designed tactile skin arrays.

7.2.3 Human Motion Study

In the future the DTW algorithm will be evaluated in a larger cross-longitudinal study focussed on Autism diagnosis. The motions generated by the Zeno robot will be made more natural and human like. Data collection, natural human motion generation, and analysis will be contained in the newly developed motor cortex of the Zeno robot. The imitation script will be modified to increase its ability to engage the children with autism. The analysis of imitation will be extended to facial gestures.

7.2.4 Software Developed

In the future the developed software will be further expanded and standardized. This will enable better sharing of the algorithms and help make this research reproducible.

REFERENCES

[1] R. Dahiya, "Electronic Skin *," in *XVIII AISEM Annual Conference*, 2015, pp. 3–6.

[2] M. W. Strohmayr, H. Wörn, and G. Hirzinger, "The DLR Artificial Skin Part I : Uniting Sensitivity and Robustness," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 1012–1018.

[3] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–73, Feb. 2013.

[4] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical humanrobot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, Mar. 2008.

[5] B. D. Argall and A. G. Billard, "A survey of Tactile Human-Robot Interactions," *Robotics and Autonomous Systems*, vol. 58, no. 10, pp. 1159–1176, 2010.

[6] J. Rajruangrabin and D. O. Popa, "Enhancement of Manipulator Interactivity through Compliant Skin and Extended Kalman Filtering," in *Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on*, Sept. 2007, pp. 1111–1116.

[7] O. Khatib, K. Yokoi, O. Brock, K. Chang, and A. Casal, "Robots in Human Environments," in *in Proceedings of the 1st Workshop on Robot Motion and Control*. Ieee, 1999, pp. 213–221.

[8] R. T. Consortium, "A Roadmap for U . S . Robotics," Tech. Rep., 2013.

[9] H. Modares, I. Ranatunga, F. L. Lewis, and D. Popa, "Optimized assistive human-robot interaction using reinforcement learning," *IEEE Transactions on Cybernetics*, 2015.

[10] I. Ranatunga, S. M. Tousif, D. O. Popa, and F. L. Lewis, "Adaptive Admittance Control for Human-Robot Interaction Using," *IEEE Transactions on Control Systems Technology*, vol. 00, no. 0, pp. 1–10, 2015.

[11] B. AlQaudi, I. Ranatunga, S. M. Tousif, F. L. Lewis, and D. O. Popa, "Model Reference Adaptive Impedance Control for Physical Human-Robot Interaction," *Autonomous Robots*, 2015.

[12] I. Ranatunga, I. Wijayasinghe, N. Balakrishnan, N. Bugnariu, and D. O. Popa, "Human-Robot Gesture Analysis for Diagnosis of Autism," *International Journal of Social Robotics*, 2015.

[13] I. Ranatunga, S. Cremer, D. O. Popa, and F. L. Lewis, "Tactile Calibration Independent Adaptive Admittance Control," *IEEE Trans. Robot.*, 2015.

[14] I. Ranatunga, S. Cremer, D. O. Popa, and F. L. Lewis, "Intent Aware Adaptive Admittance Control for Physical Human-Robot Interaction," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, Seattle, Washington, USA, 2015.

[15] G. M. Atmeh, I. Ranatunga, D. Popa, K. Subbarao, F. Lewis, and P. Rowe, "Implementation of an Adaptive, Model Free, Learning Controller on the Atlas Robot *," in *American Control Conference (ACC), 2014*, 2014.

[16] S. Cremer, I. Ranatunga, and D. O. Popa, "Robotic Waiter with Physical Co-manipulation Capabilities," in *IEEE Int. Conf. Autom. Sci. Eng.* IEEE, 2014.

[17] A. Habib, I. Ranatunga, K. Shook, and D. O. Popa, "SkinSim : A Simulation Environment for Multimodal Robot Skin," in *IEEE Int. Conf. Autom. Sci. Eng.* IEEE, 2014.

[18] N. Bugnariu, C. Young, K. Rockenbach, R. M. Patterson, C. Garver, I. Ranatunga, M. Beltran, N. Torres-Arenas, and D. Popa, "Human-robot interaction as a tool to evaluate and quantify motor imitation behavior in children with Autism Spectrum Disorders," in *2013 International Conference on Virtual Rehabilitation (ICVR)*. IEEE, Aug. 2013, pp. 57–62.

[19] J. Sanford, I. Ranatunga, and D. Popa, "Physical human-robot interaction with a mobile manipulator through pressure sensitive robot skin," in *Proc. 6th Int. Conf. PErvasive Technol. Relat. to Assist. Environ. - PETRA '13*. New York, New York, USA: ACM Press, 2013, pp. 1–6.

[20] I. Ranatunga, M. Beltran, N. A. Torres, N. Bugnariu, R. M. Patterson, C. Garver, and D. O. Popa, "Human-Robot Upper Body Gesture Imitation Analysis for Autism Spectrum Disorders," in *International Conference on Social Robotics*, ser. Lecture Notes in Computer Science, G. Herrmann, M. Pearson, A. Lenz, P. Bremner, A. Spiers, and U. Leonards, Eds., vol. 8239, no. Cdc. Springer International Publishing, 2013, pp. 218–228.

[21] J. Rajruangrabin, I. Ranatunga, and D. O. Popa, "Adaptive Interface Mapping for Intuitive Teleoperation of Multi-DOF Robots," in *Advances in Autonomous Robotics*, ser. Lecture Notes in Computer Science, G. Herrmann, M. Studley, M. Pearson, A. Conn, C. Melhuish, M. Witkowski, J.-H. Kim, and P. Vadakkepat, Eds., vol. 7429. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 49–60.

[22] I. Ranatunga and D. O. Popa, "Collision-free trajectory generation on assistive robot Neptune," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '12*. ACM, 2012.

[23] I. Ranatunga, N. A. Torres, R. Patterson, N. Bugnariu, M. Stevenson, D. O. Popa, and F. Worth, "RoDiCA: a Human-Robot Interaction System for Treatment of Childhood Autism Spectrum Disorders," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '12. New York, NY, USA: ACM, 2012.

[24] N. A. Torres, N. Clark, I. Ranatunga, and D. Popa, "Implementation of Interactive Arm Playback Behaviors of Social Robot Zeno for Autism Spectrum Disorder Therapy," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '12. New York, NY, USA: ACM, 2012, p. 1.

[25] P. Kanajar, I. Ranatunga, J. Rajruangrabin, D. O. Popa, and F. Makedon, "Neptune : Assistive Robotic System for Children with Motor Impairments," in *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '11. New York, NY, USA: ACM, 2011, pp. 1–6.

[26] I. Ranatunga, J. Rajruangrabin, D. O. Popa, and F. Makedon, "Enhanced therapeutic interactivity using social robot Zeno," *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '11*, p. 1, 2011.

[27] I. Ranatunga, S. Cremer, F. L. Lewis, and D. O. Popa, "Neuroadaptive Control for Safe Robots in Human Environments: A Case Study," in *IEEE International Conference on Automation Science and Engineering*. IEEE Press, 2015.

[28] I. Ranatunga, N. Balakrishnan, and D. O. Popa, "Trajectory Adaptation for Robot Differential Teaching," in *Proceedings of the 8th International Conference on PErvasive Technologies Related to Assistive Environments*. New York, NY, USA: ACM, 2015.

[29] G. M. Atmeh, I. Ranatunga, D. O. Popa, and K. Subbarao, "Indoor Quadrotor State Estimation using Visual Markers Categories and Subject Descriptors," in *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '14*. ACM, 2014.

[30] B. Komati, M. R. Pac, I. Ranatunga, C. Clevy, D. O. Popa, and P. Lutz, "Explicit force control vs impedance control for micromanipulation," in *Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. ASME, 2013.

[31] C. McMurrough, I. Ranatunga, A. Papangelis, D. O. Popa, and F. Makedon, "A development and evaluation platform for non-tactile power wheelchair controls," in *Proc. 6th Int. Conf. PErvasive Technol. Relat. to Assist. Environ. - PETRA '13*. New York, New York, USA: ACM Press, 2013, pp. 1–4.

[32] A. Papangelis, I. Ranatunga, D. O. Popa, and F. Makedon, "An assistive object manipulation system," in *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '13*, 2013, pp. 1–4.

[33] H. Ding, M. Schipper, and B. Matthias, "Optimized Task Distribution for Industrial Assembly in Mixed Human-Robot Environments Case Study on IO Module Assembly," in *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, 2014, pp. 19–24.

[34] M. S. Erden and A. Billard, "End-point Impedance Measurements at Human Hand during Interactive Manual Welding with Robot," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, pp. 126–133.

[35] B. Llorens-Bonilla and H. H. Asada, "A Robot on the Shoulder : Coordinated Human - Wearable Robot Control using Coloured Petri Nets and Partial Least

Squares Predictions," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, pp. 119–125.

[36] J. J. Craig, *Introduction To Robotics: Mechanics And Control*. Prentice Hall, 2008.

[37] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2009.

[38] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley, 2005.

[39] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot Manipulator Control: Theory and Practice*, ser. Control Engineering. Marcel Dekker, 2004.

[40] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control Of Robot Manipulators And Non-Linear Systems*, ser. Taylor & Francis systems and control book series. Taylor & Francis, 1999.

[41] S. S. Ge, T. H. Lee, and C. J. Harris, *Adaptive neural network control of robotic manipulators*. World Scientific.

[42] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive Processing*, vol. 12, no. 4, pp. 319–340, 2011.

[43] O. Sigaud, C. Salaün, and V. Padois, "On-line Regression Algorithms for Learning Mechanical Models of Robots: A Survey," *Robot. Auton. Syst.*, vol. 59, no. 12, pp. 1115–1129, Dec. 2011.

[44] P. J. Werbos, "Neural networks for control and system identification," in *Decision and Control, 1989., Proceedings of the 28th IEEE Conference on*, 1989, pp. 260–265 vol.1.

[45] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *Neural Networks, IEEE Transactions on*, vol. 1, no. 1, pp. 4–27, 1990.

[46] D. A. White and D. A. Sofge, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptative Approaches*, ser. Vnr Computer Library. Van Nostrand Reinhold, 1992.

[47] J. Si, A. G. Barto, and W. B. Powell, *Handbook of learning and approximate dynamic programming*, ser. IEEE Press Series on Computational Intelligence. IEEE Press, 2004.

[48] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," *Neural Networks, IEEE Transactions on*, vol. 6, no. 3, pp. 703–715, 1995.

[49] F.-C. Chen and H. K. Khalil, "Adaptive control of nonlinear systems using neural networks," *International Journal of Control*, vol. 55, no. 6, pp. 1299–1317, 1992.

[50] F. L. Lewis, A. Yegildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance." *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 7, no. 2, pp. 388–99, Jan. 1996.

[51] S. Jagannathan and F. L. Lewis, "Multilayer discrete-time neural-net controller with guaranteed performance," *Neural Networks, IEEE Transactions on*, vol. 7, no. 1, pp. 107–130, 1996.

[52] M. M. Polycarpou, "Stable adaptive neural control scheme for nonlinear systems," *Automatic Control, IEEE Transactions on*, vol. 41, no. 3, pp. 447–451, 1996.

[53] G. A. Rovithakis and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 24, no. 3, pp. 400–412, 1994.

[54] A. S. Poznyak, W. Yu, E. N. Sanchez, and J. P. Perez, "Nonlinear adaptive trajectory tracking using dynamic neural networks," *Neural Networks, IEEE Transactions on*, vol. 10, no. 6, pp. 1402–1411, 1999.

[55] G. A. Rovithakis, "Performance of a neural adaptive tracking controller for multi-input nonlinear dynamical systems in the presence of additive and multiplicative external disturbances," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 30, no. 6, pp. 720–730, 2000.

[56] R. Zbikowski and A. Dzielinski, *Neural Adaptive Control Technology*, ser. World Scientific series in robotics and intelligent systems, J. Kalkkuhl and K. J. Hunt, Eds. World Scientific, 1996.

[57] Y. H. Kim and F. L. Lewis, *High-Level Feedback Control with Neural Networks*, ser. World Scientific series in robotics and intelligent systems. River Edge, NJ, 1998.

[58] F. L. Lewis, J. Campos, and R. Selmic, *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities*, ser. Frontiers in Applied Mathematics. Society of Industrial and Applied Mathematics Press, Philadelphia, PA, 2002.

[59] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable Adaptive Neural Network Control*, ser. The International Series on Asian Studies in Computer and Information Science. Kluwer, Boston, MA, 2001.

[60] S. Franklin, D. M. Wolpert, and D. W. Franklin, "Visuomotor feedback gains upregulate during the learning of novel dynamics," *Journal of Neurophysiology*, no. 2, pp. 467–478, 2012.

[61] S. Suzuki and K. Furuta, "Adaptive Impedance Control to Enhance Human Skill on a Haptic Interface System," *Journal of Control Science and Engineering*, vol. 2012, no. 2, 2012.

[62] C. C. Kemp, A. Edsinger, and E. Torres-Jara, "Challenges for robot manipulation in human environments [Grand Challenges of Robotics]," *Robotics Automation Magazine, IEEE*, vol. 14, no. 1, pp. 20–29, Mar. 2007.

[63] L. Baudouin, N. Perrin, T. Moulard, F. Lamiraux, O. Stasse, and E. Yoshida, "Real-time replanning using 3D environment for humanoid robot," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, 2011, pp. 584–589.

[64] R. B. Rusu, A. Holzbach, R. Diankov, G. Bradski, and M. Beetz, "Perception for mobile manipulation and grasping using active stereo," in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, 2009, pp. 632–638.

[65] J. Bohren, R. B. Rusu, E. G. Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mosenlechner, W. Meeussen, and S. Holzer, "Towards autonomous robotic butlers: Lessons learned with the PR2," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 5568–5575.

[66] J. E. Colgate and N. Hogan, "Robust control of dynamically interacting systems," *International Journal of Control*, vol. 48, no. 1, pp. 65–88, 1988.

[67] N. Hogan, "Impedance control - An approach to manipulation. I - Theory. II - Implementation. III - Applications," *Transaction of the ASME, Journal of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 1–24, 1985.

[68] S. K. Singh and D. O. Popa, "An analysis of some fundamental problems in adaptive control of force and impedance behavior: theory and experiments," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 6, pp. 912–921, Dec. 1995.

[69] H. Kazerooni, T. Sheridan, and P. Houpt, "Robust compliant motion for manipulators, part I: The fundamental concepts of compliant motion," *Robotics and Automation, IEEE Journal of*, vol. 2, no. 2, pp. 83–92, June 1986.

[70] F. Ficuciello, A. Romano, L. Villani, and B. Siciliano, "Cartesian Impedance Control of Redundant Manipulators for Human-Robot Co-Manipulation," in *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*, 2014, pp. 2120–2125.

[71] C. Ott, R. Mukherjee, and Y. Nakamura, "Unified Impedance and Admittance Control," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 554–561.

[72] R. Volpe and P. Khosla, "A theoretical and experimental investigation of explicit force control strategies for manipulators," *Automatic Control, IEEE Transactions on*, vol. 38, no. 11, pp. 1634–1650, 1993.

[73] S. D. Eppinger and W. P. Seering, "On dynamic models of robot force control," in *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol. 3, 1986, pp. 29–34.

[74] J. J. Craig and M. Raibert, "A systematic method of hybrid position/force control of a manipulator," in *Computer Software and Applications Conference, 1979. Proceedings. COMPSAC 79. The IEEE Computer Society's Third International*, 1979, pp. 446–451.

[75] T. Tsuji and Y. Tanaka, "Tracking control properties of human-robotic systems based on impedance control," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, no. 4, pp. 523–535, July 2005.

[76] A. Morbi, M. Ahmadi, A. D. C. Chan, and R. Langlois, "Stability-Guaranteed Assist-as-Needed Controller for Powered Orthoses," *Control Systems Technology, IEEE Transactions on*, vol. 22, no. 2, pp. 745–752, Mar. 2014.

[77] P. Marayong, G. D. Hager, and A. M. Okamura, "Control methods for guidance virtual fixtures in compliant human-machine interfaces," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept. 2008, pp. 1166–1172.

[78] W. Yu, J. Rosen, and X. Li, "PID admittance control for an upper limb exoskeleton," in *American Control Conference (ACC), 2011*, June 2011, pp. 1124–1129.

[79] A. Albu-Schäffer and G. Hirzinger, "Cartesian impedance control techniques for torque controlled light-weight robots," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 1, 2002, pp. 657–663 vol.1.

[80] L. Huang, S. S. Ge, and T. H. Lee, "An adaptive impedance control scheme for constrained robots," *International Journal of Computers, Systems and Signals*, vol. 5, no. 2, 2004.

[81] S. Jung, T. C. Hsia, and R. G. Bonitz, "Force tracking impedance control of robot manipulators under unknown environment," *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 3, pp. 474–483, May 2004.

[82] H. Wang, K. H. Low, and M. Y. Wang, "Combined Impedance/Direct Control of Robot Manipulators," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Oct. 2006, pp. 3605–3610.

[83] E. Gribovskaya, A. Kheddar, and A. Billard, "Motion learning and adaptive impedance for robot control during physical interaction with humans," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, May 2011, pp. 4326–4332.

[84] Y. Li and S. S. Ge, "Impedance Learning for Robots Interacting With Unknown Environments," *Control Systems Technology, IEEE Transactions on*, vol. 22, no. 99, pp. 1422–1432, 2013.

[85] C. Mitsantisuk, K. Ohishi, and S. Katsura, "Variable mechanical stiffness control based on human stiffness estimation," in *2011 IEEE International Conference on Mechatronics*, Apr. 2011, pp. 731–736.

[86] F. Dimeas and N. Aspragathos, "Fuzzy Learning Variable Admittance Control for Human-Robot Cooperation," in *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*, 2014, pp. 4770–4775.

[87] A. Lecours, B. Mayer-St-Onge, and C. Gosselin, "Variable admittance control of a four-degree-of-freedom intelligent assist device," in *2012 IEEE International Conference on Robotics and Automation*, no. 2, May 2012, pp. 3903–3908.

[88] S. Haddadin, A. Albu-Schäffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical Human-Robot Interaction," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on.* IEEE, Sept. 2008, pp. 3356–3363.

[89] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "The role of the robot mass and velocity in physical human-robot interaction - Part I: Non-constrained blunt impacts," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, pp. 1331–1338.

[90] S. Haddadin, A. Albu-Schäffer, M. Frommberger, and G. Hirzinger, "The role of the robot mass and velocity in physical human-robot interaction - Part II: Constrained blunt impacts," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, no. 011838. IEEE, May 2008, pp. 1339–1345.

[91] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Soft-Tissue Injury in Robotics," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 3426–3433.

[92] K. A. Wyrobek, E. H. Berger, H. M. Van der Loos, and J. K. Salisbury, "Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, May 2008, pp. 2165–2170.

[93] L. Maiolo, A. Pecora, F. Maita, A. Minotti, E. Zampetti, S. Pantalei, A. Macagnano, A. Bearzotti, D. Ricci, and G. Fortunato, "Flexible sensing systems based on polysilicon thin film transistors technology," *Sensors and Actuators, B: Chemical*, vol. 179, pp. 114–124, 2013.

[94] P. Mittendorfer, E. Dean, and G. Cheng, "Automatic Robot Kinematic Modeling with a Modular Artificial Skin," in *IEEE-RAS International Conference on Humanoid Robots*, 2014.

[95] P. Maiolino, M. Maggiali, G. Cannata, G. Metta, and L. Natale, "A flexible and robust large scale capacitive tactile system for robots," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3910–3917, 2013.

[96] M. Ohka, Y. Mitsuya, Y. Matsunaga, and S. Takeuchi, "Sensing characteristics of an optical three-axis tactile sensor under combined loading," *Robotica*, vol. 22, no. 2, pp. 213–221, 2004.

[97] N. F. Lepora, U. Martinez-Hernandez, M. Evans, L. Natale, G. Metta, and T. J. Prescott, "Embodied hyperacuity for robot touch," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 10–11.

[98] D. M. Vogt, Y. Menguc, Y.-L. Park, M. Wehner, R. Kramer, C. Majidi, L. Jentoft, Y. Tenzer, R. Howe, and R. J. Wood, "Progress in Soft, Flexible, and Stretchable Sensing Systems," in *IEEE International Conference on*, 2013, pp. 0–1.

[99] C. Breazeal and B. Scassellati, "Challenges in building robots that imitate people," pp. 363–390, 2002.

[100] P. Niu, P. Chapman, R. Riemer, and X. Zhang, "Evaluation of motions and actuation methods for biomechanical energy harvesting," *PESC Record - IEEE Annual Power Electronics Specialists Conference*, vol. 3, pp. 2100–2106, 2004.

[101] E. Murphy-chutorian, S. Member, and M. M. Trivedi, "Head Pose Estimation in Computer Vision: A Survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 4, pp. 607–626, 2009.

[102] D. H. Geschwind, "Advances in Autism," *Annual Review of Medicine*, vol. 60, no. 1, pp. 367–380, 2009.

[103] C. for Disease Control and Prevention, "Autism and Developmental Disabilities Monitoring (ADDM) Network," Sept. 2012.

[104] B. Scassellati, H. Admoni, and M. Mataric, "Robots for Use in Autism Research," *Annual Review of Biomedical Engineering*, vol. 14, no. 1, pp. 275–294, 2012.

[105] A. Tapus, A. Peca, A. Aly, C. Pop, L. Jisa, S. Pintea, A. S. Rusu, and D. O. David, "Children with autism social engagement in interaction with Nao, an imitative robot." *Interaction Studies*, vol. 13, no. 3, pp. 315–347, 2012.

[106] N. Rinehart, J. Bradshaw, A. Brereton, and B. Tonge, "Movement Preparation in High-Functioning Autism and Asperger Disorder: A Serial Choice Reaction Time Task Involving Motor Reprogramming," *Journal of Autism and Developmental Disorders*, vol. 31, no. 1, pp. 79–88, 2001.

[107] D. J. Ricks and M. B. Colton, "Trends and considerations in robot-assisted autism therapy," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 4354–4359.

[108] B. Robins, K. Dautenhahn, and J. Dubowski, "Does appearance matter in the interaction of children with autism with a humanoid robot?" *Interaction Studies*, vol. 7, no. 3, pp. 509–542, 2006.

[109] J. Fasola and M. J. Mataric, "Using Socially Assistive Human-Robot Interaction to Motivate Physical Exercise for Older Adults," *Proceedings of the IEEE*, vol. 100, no. 8, pp. 2512–2526, 2012.

[110] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 26, no. 1, pp. 43–49, Feb. 1978.

[111] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, 2005.

[112] S. Kim, C. H. Kim, and J. H. Park, "Human-like Arm Motion Generation for Humanoid Robots Using Motion Capture Database," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Oct. 2006, pp. 3486–3491.

[113] C. H. Kim, S. Kim, S. Ra, and B.-J. You, "Regenerating human-like arm motions of humanoid robots for a movable object," in *SICE, 2007 Annual Conference*, Sept. 2007, pp. 1081–1086.

[114] S. Gaertner, M. Do, T. Asfour, R. Dillmann, C. Simonidis, and W. Seemann, "Generation of Human-like Motion for Humanoid Robots Based on Marker-based Motion Capture Data," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, June 2010, pp. 1–8.

[115] S. Calinon, F. Guenter, and A. Billard, "On Learning, Representing, and Generalizing a Task in a Humanoid Robot," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 2, pp. 286–298, Apr. 2007.

[116] S. H. Lee, I. H. Suh, S. Calinon, and R. Johansson, "Learning basis skills by autonomous segmentation of humanoid motion trajectories," in *Humanoid*

Robots (Humanoids), 2012 12th IEEE-RAS International Conference on, Nov. 2012, pp. 112–119.

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

[118] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 2587–2592.

[119] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 763–768.

[120] R. Smith and Others, "Open dynamics engine," 2005.

[121] E. Coumans, "Bullet physics engine," 2010.

[122] M. A. Sherman, A. Seth, and S. L. Delp, "Simbody: multibody dynamics for biomedical research," *Procedia* $\{IUTAM\}$, vol. 2, no. 0, pp. 241–261, 2011.

[123] B. MacIntyre, M. Gandy, S. Dow, and J. D. Bolter, "DART: a toolkit for rapid design exploration of augmented reality experiences," in *Proceedings of the 17th annual ACM symposium on User interface software and technology*. ACM, 2004, pp. 197–206.

[124] N. Hogan and S. P. Buerger, "Impedance and interaction control," in *Robot. Autom. Handb.*, 1st ed., T. R. Kurfess, Ed. CRC Press, 2005, ch. 19.

[125] B. Widrow and E. Walach, *Adaptive Inverse Control, Reissue Edition: A Signal Processing Approach*, ser. IEEE Press series on power engineering. Wiley, 2008.

[126] F. L. Lewis, L. Xie, and D. Popa, *Optimal and robust estimation: with an introduction to stochastic control theory*, 2nd ed., ser. Automation and control engineering. CRC Press, 2008.

[127] F. L. Lewis, *Applied Optimal Control and Estimation: Digital Design and Implementation*, ser. Digital signal processing series. Prentice Hall, 1992.

[128] F. L. Lewis, *Optimal Estimation*. New York: John Wiley & Sons, 1986.

[129] S. S. Haykin, *Adaptive filter theory*, ser. Prentice-Hall information and system sciences series. Prentice Hall, 2002.

[130] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*. CRC Press, 2004.

[131] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 1, pp. 43–53, Feb. 1987.

[132] Matthias Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 464–469.

[133] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[134] N. Hogan and D. Sternad, "Sensitivity of smoothness measures to movement duration, amplitude, and arrests." *Journal of Motor Behavior*, vol. 41, no. 6, pp. 529–34, Nov. 2009.

[135] "Skyball XI," \url{http://www.skyballinfo.com}.

[136] D. M. Baer, R. F. Peterson, and J. A. Sherman, "The development of imitation by reinforcing behavioral similarity to a model," *Journal of the Experimental Analysis of Behavior*, vol. 10, no. 5, p. 405, 1967.

[137] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 23, no. 1, pp. 67–72, Feb. 1975.

[138] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, "Learning Force Control Policies for Compliant Manipulation," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 4639–4644.

[139] J. Jackson, "Microsoft robotics studio: A technical introduction," *IEEE Robotics & Automation Magazine*, vol. 14, no. 4, pp. 82–87, Dec. 2007.

[140] O. Michel, "Webots: Professional Mobile Robot Simulation," in *International Journal of Advanced Robotic Systems*, 2004, pp. 39–42.

[141] G. Echeverria, N. Lassabe, A. Degroote, and S. Lemaignan, "Modular open robots simulation engine: MORSE," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, May 2011, pp. 46–51.

[142] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems," in *In Proceedings of the 11th International Conference on Advanced Robotics*, 2003, pp. 317–323.

[143] F. Kanehiro, H. Hirukawa, and S. Kajita, "OpenHRP: Open Architecture Humanoid Robotics Platform," *The International Journal of Robotics Research*, vol. 23, no. 2, pp. 155–165, 2004.

[144] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Nov. 2013, pp. 1321–1326.

[145] B. León, S. Ulbrich, R. Diankov, G. Puche, M. Przybylski, A. Morales, T. Asfour, S. Moisio, J. Bohg, J. Kuffner, and R. Dillmann, "OpenGRASP: A Toolkit for Robot Grasping Simulation," in *Simulation, Modeling, and Programming for Autonomous Robots*, N. Ando, S. Balakirsky, T. Hemker, M. Reggiani, and O. von Stryk, Eds. Springer Berlin Heidelberg, 2010, pp. 109–120.

[146] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3. IEEE, pp. 2149–2154.

[147] G. Pratt and J. Manzo, "The DARPA Robotics Challenge [Competitions]," *IEEE Robotics & Automation Magazine*, vol. 20, no. 2, pp. 10–12, June 2013.

[148] D. Coleman, I. A. Sucan, S. Chitta, and N. Correll, "Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study." in *CoRR*, 2014.

[149] D. Gossow, A. Leeper, D. Hershberger, and M. Ciocarlie, "Interactive Markers: 3-D User Interfaces for ROS Applications [ROS Topics]," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 14–15, Dec. 2011.

[150] J. Meyer, P. Lukowicz, and G. Troster, "Textile Pressure Sensor for Muscle Activity and Motion Detection," in *2006 10th IEEE International Symposium on Wearable Computers*. IEEE, Oct. 2006, pp. 69–72.

[151] B. Herold, M. Geyer, and C. J. Studman, "Fruit contact pressure distributions equipment," *Computers and Electronics in Agriculture*, vol. 32, no. 3, pp. 167–179, 2001.

BIOGRAPHICAL STATEMENT

Isura Ranatunga was born in Galle, Sri Lanka, in 1987. He received his Ph.D. degree in Electrical Engineering with a focus on robotics and automation from The University of Texas at Arlington, in 2015. He received his B.Sc. degree in Electrical Engineering from The University of Texas at Arlington, in 2010.

He worked as a Graduate Research Assistant with the University of Texas at Arlington Research Institute, Fort Worth, TX, USA and the Next Generation Systems Research Group. His current research interests include force control, physical human-robot interaction, bipedal walking, adaptive robot control, and autonomous navigation.