

University of Louisville

## ThinkIR: The University of Louisville's Institutional Repository

---

Electronic Theses and Dissertations

---

8-2019

# Adaptive physical human-robot interaction (PHRI) with a robotic nursing assistant.

Sumit Kumar Das

*University of Louisville*

Follow this and additional works at: <https://ir.library.louisville.edu/etd>



Part of the [Controls and Control Theory Commons](#)

---

### Recommended Citation

Das, Sumit Kumar, "Adaptive physical human-robot interaction (PHRI) with a robotic nursing assistant." (2019). *Electronic Theses and Dissertations*. Paper 3268.

<https://doi.org/10.18297/etd/3268>

This Doctoral Dissertation is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact [thinkir@louisville.edu](mailto:thinkir@louisville.edu).

ADAPTIVE PHYSICAL HUMAN-ROBOT INTERACTION (PHRI) WITH A  
ROBOTIC NURSING ASSISTANT

By

Sumit Kumar Das  
M.S., Electrical Engineering,  
University of Texas, Arlington, TX

A Dissertation  
Submitted to the Faculty of the  
J.B. Speed School of Engineering of the University of Louisville  
in Partial Fulfillment of the Requirements  
for the Degree of

Doctor of Philosophy  
in Electrical Engineering

Department of Electrical and Computer Engineering  
University of Louisville  
Louisville, Kentucky

August 2019

Copyright 2019 by Sumit Kumar Das

All rights reserved





ADAPTIVE PHYSICAL HUMAN-ROBOT INTERACTION (PHRI) WITH A  
ROBOTIC NURSING ASSISTANT

By

Sumit Kumar Das  
M.S., Electrical Engineering,  
University of Texas, Arlington, TX

A Dissertation Approved On

July 19, 2019

by the following Dissertation Committee:

---

Prof. Dan O. Popa, Ph.D., Dissertation Director

---

Prof. M Cynthia Logsdon, Ph.D., WHNP-BC, FAAN

---

Prof. Tamer Inanc, Ph.D.

---

Prof. Olfa Nasraoui, Ph.D.

---

Prof. John Naber, Ph.D.

## DEDICATION

In memory of Dolagobinda Mohanty and Bibhudhendra Mohanty.

## ACKNOWLEDGEMENTS

I would like to thank Dr. Dan Popa for guiding and advising me during my PhD studies. I am very grateful for the constant support and opportunities he provided. I had an enriching experience working at the Next Generation Systems lab. I would also like to thank Dr. M Cynthia Logsdon, Dr. Tamer Inanc, Dr. Olfa Nasraoui and Dr. John Naber for taking the time out of their busy schedule to serve on my thesis defense committee.

I would like to thank and acknowledge Dr. Isura Ranatunga, Dr. Sven Cremer and Dr. Indika Wijayasinghe who mentored me and helped me develop an aptitude for research work. I have always learned from our discussions on robotics. I would also like to thank my lab-mates Christopher Robinson, Abubakar Shamasudeen, Jordan Klotz, Ruoshi Zhang, Zhong Yang, Danming Wei, Maxwell Carter, Emily Legg and Dr. Mohammad Nasser Saadatzi for their support.

I am always indebted to my parents, Sunil Kumar Das and Namita Das, for their unconditional love, sacrifice and unwavering support. They have always believed in my abilities and encouraged me to push my boundaries and to not be complacent in my comfort zone. Their teachings and advises have always guided me during difficult times. I consider myself lucky to have parents like them. I would like to thank my brother Sambit Kumar Das for his love and support.

I am thankful to have the support of my wife, Ankita Sahu, who stood by me and encouraged to pursue my dreams without hesitation. She is always the voice of reason whenever I am overwhelmed and keeps me level-headed. Thank you for being my collaborator, proofreader, therapist and above all my friend. Our discussions on wide range of topics have led to many projects requiring us to constantly better ourselves and hone our skills. Thank you for always motivating me do more than I thought I could accomplish.

I am grateful to have a friend like Biswa Prakash Sarangi who has always been by my side since our school days. He understood me better than anybody and never hesitated to tell me the harsh truth whenever I was wrong. He is the one person who is always available whenever I need him, and I will always cherish our friendship.

## ABSTRACT

### ADAPTIVE PHYSICAL HUMAN-ROBOT INTERACTION (PHRI) WITH A ROBOTIC NURSING ASSISTANT

Sumit Kumar Das

July 19, 2019

Recently, more and more robots are being investigated for future applications in health-care. For instance, in nursing assistance, seamless Human-Robot Interaction (HRI) is very important for sharing workspaces and workloads between medical staff, patients, and robots. In this thesis we introduce a novel robot - the Adaptive Robot Nursing Assistant (ARNA) and its underlying components. ARNA has been designed specifically to assist nurses with day-to-day tasks such as walking patients, pick-and-place item retrieval, and routine patient health monitoring.

An adaptive HRI in nursing applications creates a positive user experience, increase nurse productivity and task completion rates, as reported by experimentation with human subjects. ARNA has been designed to include interface devices such as tablets, force sensors, pressure-sensitive robot skins, LIDAR and RGBD camera. These interfaces are combined with adaptive controllers and estimators within a proposed framework that contains multiple innovations. A research study was conducted on methods of deploying an ideal Human-Machine Interface (HMI), in this case a tablet-based interface. Initial study points to the fact that a traded control level of autonomy is ideal for tele-operating ARNA by a patient. The proposed method of using the HMI devices makes the performance of a robot similar for both skilled and un-skilled workers.

A neuro-adaptive controller (NAC), which contains several neural-networks to estimate and compensate for system non-linearities, was implemented on the ARNA robot. By linearizing the system, a cross-over usability condition is met through which humans find it more intuitive to learn to use the robot in any location of its workspace. A novel Base-Sensor Assisted Physical Interaction (BAPI) controller is introduced in this thesis, which utilizes a force-torque sensor at the base of the ARNA robot manipulator to detect full body collisions, and make interaction safer. Finally, a human-intent estimator (HIE) is proposed to estimate human intent while the robot and user are physically collaborating during certain tasks such as adaptive walking. A NAC with HIE module was validated on a PR2 robot through user studies. Its implementation on the ARNA robot platform can be easily accomplished as the controller is model-free and can learn robot dynamics online.

A new framework, Directive Observer and Lead Assistant (DOLA), is proposed for ARNA which enables the user to interact with the robot in two modes: physically, by direct push-guiding, and remotely, through a tablet interface. In both cases, the human is being “observed” by the robot, then guided and/or advised during interaction. If the user has trouble completing the given tasks, the robot adapts their repertoire to lead users toward completing goals. The proposed framework incorporates interface devices as well as adaptive control systems in order to facilitate a higher performance interaction between the user and the robot than was previously possible.

The ARNA robot was deployed and tested in a hospital environment at the School of Nursing of the University of Louisville. The user-experience tests were conducted with the help of healthcare professionals where several metrics including completion time, rate and level of user satisfaction were collected to shed light on the performance of various components of the proposed framework. The results indicate an overall positive response towards the use of such assistive robot in the healthcare environment. The analysis of these gathered data is included in this document.

To summarize, this research study makes the following contributions:

- Conducting user experience studies with the ARNA robot in patient sitter and walker

scenarios to evaluate both physical and non-physical human-machine interfaces.

- Evaluation and Validation of Human Intent Estimator (HIE) and Neuro-Adaptive Controller (NAC).
- Proposing the novel Base-Sensor Assisted Physical Interaction (BAPI) controller.
- Building simulation models for packaged tactile sensors and validating the models with experimental data.
- Description of Directive Observer and Lead Assistance (DOLA) framework for ARNA using adaptive interfaces.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
ABSTRACT	vi
LIST OF TABLES	xiii
LIST OF FIGURES	xiv
INTRODUCTION	1
1.1 Motivation	3
1.2 Contributions	7
1.3 Thesis Outline	10
1.4 List of Publications	10
DIRECTIVE OBSERVER AND LEAD ASSISTANCE (DOLA) FRAMEWORK	11
2.1 Components	13
2.1.1 Task Model	14
2.1.2 Human-Machine Interfaces (HMI)	14
2.1.3 Network	15
2.1.4 Controllers	16
2.1.5 Sensors	16
2.1.6 Robot Hardware	17
2.2 Architecture of DOLA framework	17
ADAPTIVE ROBOT NURSING ASSISTANT	20
3.1 Robot Architecture	20
3.1.1 Components	22
3.2 Robot Functions	24
3.2.1 Sitter Scenario	24



3.2.2 Walker Scenario	26
3.3 Discussion	29
SENSORS AND HUMAN-MACHINE INTERFACES	30
4.1 Tactile Sensors	30
4.1.1 Sensor Design, Fabrication and Experimental Loading	32
4.1.2 Fabrication and Packaging	33
4.1.3 Sensor Package Modeling	37
4.1.4 Results and Comparison	41
4.1.5 Discussion	45
4.2 Tablet Interface	46
4.2.1 System Architecture	49
4.2.2 Application Interface	55
4.2.3 Experiments and Discussion	57
NEURO-ADAPTIVE CONTROLLER (NAC) AND HUMAN INTENT ESTIMATION MODULE (HIE)	61
5.1 Introduction	61
5.2 Outer Loop	63
5.2.1 Human Transfer Function	63
5.2.2 Human Intent Estimator	65
5.3 Inner Loop	67
5.3.1 Prescribed Robot Error Dynamics	68
5.3.2 Inner-loop Error Dynamics	69
5.4 Experimental Validation	70
5.5 Discussion	73
BASE-SENSOR ASSISTED PHYSICAL INTERACTION (BAPI)	76
6.1 Whole-body Collision Detection using BAPI	76
6.1.1 Introduction	76

6.1.2	Base-Sensor Assisted Physical Interaction (BAPI)	79
6.1.3	Experimental Testbed and NN Training	85
6.1.4	Results and Discussion	87
6.2	Joint Torque Estimation using BAPI	91
6.2.1	Introduction	91
6.2.2	Controller Formulation	94
6.2.3	Experimental Implementation	101
6.2.4	Results and Discussion	104
6.2.5	Conclusions	106
	ARNA USER STUDIES	107
7.1	Introduction	107
7.2	Experiment Setup	109
7.2.1	Sitter Scenario	109
7.2.2	Walker Scenario	111
7.2.3	Questionnaire	113
7.3	Results and Discussion	114
	CONCLUSION AND FUTURE WORK	120
8.1	Conclusion	120
8.2	Future Work	122
	REFERENCES	123
	LIST OF PUBLICATIONS	134
A.1	Published Papers	134
A.2	Conditionally Accepted	135
A.3	Papers Under Review	136
A.4	Papers in Preparation	136
A.5	Patent Pending	136

QUESTIONNAIRES	137
CURRICULUM VITAE	142

## LIST OF TABLES

4.1	Dimensions of model entities	39
4.2	Physical Properties of P10 Silicone Base, Gold ID structure and Kapton sheets	39
4.3	Percent Error between Simulated and Experimental Data for change in re- sistance	43
4.4	Average rating of the interface by the users	60
6.1	Experimental Results	89
6.2	Experimental Results	104
6.3	Experimental Results	106
7.1	Phase 1 Walker Experiment Result	115
7.2	Phase 1 Walker Questionnaire Result	118
7.3	Phase 1 Walker Regression Result	118
B.1	Walker Scenario Questionnaire	137
B.2	Technology Acceptance Questionnaire	138
B.3	Human-Robot Interaction Questionnaire	139
B.4	Human-Robot Interaction Questionnaire	140

## LIST OF FIGURES

2.1	DOLA framework incorporating the entire autonomy range	12
2.2	DOLA framework	13
3.1	ARNA Robot	21
3.2	ARNA Sensor Suit	23
3.3	Digital IR Thermometer	25
3.4	ARNA walker task	27
3.5	ARNA walker with patient carrying task	27
4.1	Interdigitated Structure [1]	33
4.2	Sensor Package [1]	34
4.3	Experiment Components [1]	36
4.4	Geometric Entities [2]	38
4.5	Longitudinal and Transverse Stress [2]	40
4.6	Final and Meshed model [2]	41
4.7	Displacement Vs Force [2]	42
4.8	Plots comparing simulation and experimental results for Resistance ( $\Omega$ ) [2]	44
4.9	Plots comparing simulation and experimental results for Displacement [2]	45
4.10	Issues related to pHRI. Adapted from [3]	48
4.11	System Architecture depicting different hardware and software components [4]	50
4.12	Diagram of our proposed Human Machine Interface Framework [4]	52
4.13	Speech Interface output during conversation [4]	54
4.14	External camera sensors installed on the Baxter for user identification [4]	55
4.15	Tablet application screen [4]	56
4.16	Markers attached to objects [4]	57

4.17	Time of completion comparison between users [4]	59
5.1	NAC with HIE and human-in-loop operation [5]	62
5.2	Human intent estimator [5]	66
5.3	Inner Loop [5]	67
5.4	Experimental Setup [5]	71
5.5	Grid layout for trajectory following. [5]	73
5.6	Trace of user while following the trajectory with PED disabled (left) and PED enabled (right) [5]	74
5.7	Performance measure plots [5]	74
5.8	Controller performance for different number of hidden layer neurons (PED enabled with reference trajectory $\hat{x}_d$ ). [5]	75
6.1	6-DOF Robot manipulator constructed using torque-controlled Dynamixel motors MX-106, with an FTS at base [6]	80
6.2	BAPI controller block diagram. The upper portion represents the NAC non-linear compensator, while the BFE NN in the lower block estimates the robot interaction force at the base [6]	82
6.3	NN Validation Error Distribution [6]	86
6.4	NN Operation Error Distribution [6]	86
6.5	User interaction forces at Joint 1 ( $P_1$ ) at 20s, 40s, 60s, 80s and 100s [6]	88
6.6	Low Interaction Forces at Joint 6 ( $P_6$ ) [6]	90
6.7	Sensor data (forces and torques) from BFTS during collision [6]	90
6.8	Sensor data (forces and torques) from BFTS during no collision [6]	91
6.9	Proposed controller block diagram. The NAC (top) learns the robot dynamics, while the BFE (bottom) estimates the joint torques based on wrench experienced at the base of the robot. [7]	95
6.10	6-DOF Robot manipulator coordinate system [7]	96
6.11	Segmenting NAC NN to extract dynamics terms. [7]	98
6.12	6-DOF robot manipulator placed on a base force-torque sensor. [7]	102

6.13	Actual vs Estimated torque value at Joint 4 (top), and Joint 2 (bottom). [7]	103
6.14	Trajectory Tracking at Joint 4 (top) and Join 2 (bottom). [7]	105
7.1	Technology Acceptance Model (TAM) proposed by Davis [8]	109
7.2	ARNA at School of Nursing, University of Louisville, KY	110
7.3	Sitter Task Experiment	111
7.4	Walker Task Experiment	112
7.5	Sitter Average Completion Time	114
7.6	Sitter Average Pick Time	115
7.7	Walker Average Completion Time	116
7.8	Walker Average Speed	116
7.9	Walker Error in Trajectory Following	117
7.10	Walker Average Torque	117

## CHAPTER 1

### INTRODUCTION

With an increase in the inclusion of robots in a workspace shared by human users, the issue of human-robot interaction and user safety needs to be looked into closely. Safe physical Human-Robot Interaction (pHRI) is imperative especially in the case of interaction with industrial robots. For safe interaction, the safety standards of ISO 10218-1 and ISO 10218-2 has to be followed, as outlined by Robotic Industries Association [9] [10]. As discussed by Bicchi et. al., the safety concerns of pHRI should go hand in hand with the performance of the robot as well [11]. These objectives are achieved through implementation of sensor arrays, adaptive control system, intuitive interfaces etc. The work presented in this document explores ways to make pHRI with a nursing assistant robot safe, usable and efficient.

Medical robotics has been around since 1990s and has been helping health-care professionals perform their tasks with ease [12]. Some examples of such tasks are minimally invasive surgery (MIS) [13] and computer-integrated surgery (CIS) [14]. In an article by Okamura et. al, the application areas for health-care robots are identified as “surgical and interventional, robotic replacement of lost function, robot-assisted rehabilitation, behavioral therapy, personalized care and health promotion” [15]. In my research a novel nursing assistant robot, Adaptive Robot Nursing Assistant or ARNA, is introduced which concentrates on robotic replacement of lost function, robot-assisted rehabilitation and personalized care.

To facilitate pHRI with an assistive robot such as ARNA, an intuitive and adaptive interface is required. The interface can be in the form of sensors to provide input to the robot-system. This information is useful in interpreting the user commands or intentions on which the robot should act upon. Some examples of these sensors are tactile, ultrasonic,



infrared, cameras and force-torque (FT) sensors. Along with sensors, computing devices such as a hand-held tablet can also serve as an intuitive interface device for a user. All these devices, when combined, form the sensory part of the overall framework governing the robot.

A Neuro-Adaptive Controller (NAC) is also investigated as part of this research work. This neural-net based controller estimates and linearizes the non-linear dynamics of the robot. Learning occurs on-line and no previous training is required for the neural-network. This controller was introduced by Lewis et. al [16] [17]. Along with the NAC, a Human Intent Estimator (HIE) module was introduced to estimate the user intent during pHRI. Further refinement of the NAC includes the formulation and investigation of a Base-Sensor Assisted Physical Interaction (BAPI) controller [6]. The BAPI controller has a FT sensor attached to the base of a manipulator which allows full-body sensing without the use of tactile sensors spread throughout the robot. These controllers are integrated into the framework design for ARNA.

Kapoor et. al. have discussed the various application framework in the context of health-care robots [18]. Broadly these frameworks can be divided into three categories - *Complete Software Control, Supervisory Control and API-based control* [18]. Similarly, Wang et. al. have discussed a human-in-the-loop based framework for minimally invasive surgery [19]. These frameworks bear resemblance to the shared control scheme as described by Murphy [20]. The shared control framework allows the user to constantly supervise and guide the robot in order to accomplish an intended task. The API based control [18] discussed by Kapoor et. al. is similar to the traded control [20] scenario discussed by Murphy. The traded control scenario allows the user to decide the high-level tasks and allows the robot to autonomously complete the commanded tasks. In all these frameworks, the human user has to monitor the robot during its operation. The Directive Observer and Lead Assistance (DOLA) framework, developed as a part of our research work in this thesis, observes the user while adapting to the dynamic directives from them. The framework also incorporates the lead assistance module which help the user complete a common goal. This

framework is implemented for both physical as well as non-physical human-robot interaction.

Robot Operating System (ROS) was used for the development and control of all the components of ARNA which are described in this document. ROS is an open source development platform which can be easily used to deploy controllers, acquire sensor data and control a robot [21]. ROS is now being used prolifically in many standardized robot systems. ROS provides access to previously built packaged solutions to develop routine modules, which allows for faster development and prototyping of newer modules for a robot.

The research work also presents the qualitative and quantitative results from user experiments conducted on ARNA as well as the individual components of the robots. The future plans and user studies related to ARNA are discussed in this document, including various use-cases. The results from these experiments were analyzed and tabulated. These experiments provide context to the work developed and compare them to the existing systems in place.

## 1.1 Motivation

Collaborative robots or co-robots share the same work-space as the human users and often work in tandem to accomplish a common task. This collaboration between a human user and a robot requires various components and system checks, such as safety checks, interfaces, sensors, controls etc., to be in place to ensure smooth operation. Co-robots are not only restricted to manufacturing assembly lines [22] but have also transcended to service robots [23], crisis management [24], construction [25] and health-care [26–28]. Co-robots enhance the quality the tasks to be performed in a repeatable fashion, which is why it is imperative to study and develop systems to better facilitate physical human-robot interaction (pHRI).

Investigation of safe HRI has been conducted by many researchers in order to find parameters of operation [29–31]. Lu et. al [32], [33] have conducted studies to make robot manipulator detect and stop the operation in cases of human-robot collision. It was also found that a human user can tolerate a robot collision force of 50N [34]. This establishes

the base-line of operation for co-robots and the parameters of collision detection for safer operation. In case of robot-assisted surgeries, where pHRI is the key, studies suggest that the robot needs to be back-drivable for safe operation [35–37]. Also, according to Bicchi et. al, due to repeated manual process and material-handling, the workers are at a risk of developing *work-related musculoskeletal disorders (WMSD)* [11]. WMSD costs industries thousands of dollars in health-care and lost time [11]. This provides a strong motivation for the use of co-robots in workplaces where the robots can supplement the user in performing the intensive task loads. This is also true in case of health-care environment where some tasks can be taken over by co-robots to assist or supplement the medical personnel with their tasks.

In order to interact with a robot, a human-machine interface is required. An HMI gathers user data values through sensors and devices and interprets them into the tasks to be performed by the robot. A user can either perform gesture, verbal or non-verbal, or use computing devices in order to issue a command. A gesture set was proposed by Barattini et. al. which can be used to communicate as well as command an industrial co-robot [38]. A set of 19 gestures were proposed to be tracked through visual monitoring of the user [38]. Computing devices such as mobile phones and tablets have also been used to interface users with the robots. The integration of these computing devices has been studied by Yepes et. al. [39] and Aroca et. al. [40]. Using these devices allows mobile applications to be developed and deployed in order to communicate with a robot remotely over internet or a common network. Studies have argued that android application developed and deployed on these devices can harness the data from the complex on-board sensors integrated in them [39–41]. This allows for cost-effective implementation of HMI [42].

One of the common sensors used for HRI is the force-torque sensor (FTS). Generally, an FTS is attached to end-effector of a robot manipulator. During collaborative tasks, the FTS senses the user forces and helps the robot to adapt and move towards the intended direction. Along with FTS, tactile sensors are also another way of sensing forces exerted on a robot. E-skin has been developed which are embedded with tactile sensors and can

be wrapped around a robot. These sensors then provide force feedback data to the robot controller. This data helps determine collision events as well as user intended direction of motion. This can only be possible if the tactile sensors are well calibrated and are accurate enough to detect the applied forces.

To design an accurate and usable tactile sensor, a software model needs to be created to measure the effect of components on the sensitivity of the sensor. But modelling only the sensors numbs the effect of the skin that they are embedded in which in turn will lead to misinterpretation of the sensor data. This is because the skin also dissipates the applied forces and if this is not modelled, the misinterpretation of forces will lead to wrong control outputs. In order to mitigate this issue, a sensor package, which includes the sensor and the surrounding skin material, was modelled and experimentally validated in this thesis. This will assist in future E-skin designs which might include multi-modal sensors such as accelerometer, thermal sensors etc. A finite element analysis (FEA) model of this sensor package can be created to verify or estimate the effect of factors such as skin material and thickness on the sensor readings. This in turn will help in designing the control algorithms which will use these sensor data.

Robot control algorithms are implemented to control the robot activity such that the error between the desired task and the task being performed by the robot can be driven to zero. The most common controller in use is the PID controller which is heavily used during trajectory following tasks of a robot. During pHRI, such a rigid controller would not be able to adapt to the user intentions. Use of neural networks in a control system, as proposed by Lewis et. al [17] [16], facilitates the estimation of non-linear terms. A Neuro-Adaptive Controller (NAC) is a combination of two cascading loops, in which the inner loop estimates the system dynamics and non-linear terms using a neural network and the outer loop generates a desired trajectory by estimating the user intent or a given task model. A NAC can be used in a co-robot during pHRI to estimate and adapt to user intent while driving the robot.

Along with NAC, whole-body collision detection and avoidance is also required in the

case of robot manipulators. This is usually achieved through the use of tactile skin covering a robot. The controller then has to have the knowledge of location of each tactile sensor so that the location of the forces being applied can be interpreted from sensor data. Another method of detecting whole-body collision is to implement a force-torque sensor at the base of the robot manipulator. This allows the FTS to sense any collision or forces applied on the robot, thus eliminating the need of multiple tactile sensors. This method of collision detection would be more advantageous when implemented on a co-robot to facilitate a safe pHRI.

To develop a framework for robot operation, the roles and requirements of the robot has to be specified. A framework allows the rules and their precedence to be implemented during robot operation. Currently, there are several frameworks governing the teleoperation of a robot which define how a robot is controlled in different situations. Types of operation of the robots, as defined by Murphy, can be broadly classified into autonomous, shared and traded control [20]. In contrast, the novel Directive Observer and Lead Assistance (DOLA) framework monitors the patient during both physical and non-physical HRI. This framework assists in completion of tasks by monitoring and adapting to user needs rather than being actively driven by a human user.

Booth has mentioned 4 types of robots, which are being developed by Toyota, and their functions in the field of nursing [26]. These robot types are *walk and train assist*, *independent walk assist*, *balance train assist* and *patient transfer assist* [26]. Another type of nursing robot concentrates on elderly care and assisted-living scenarios. Robots described by Wu et. al. follow the protocol of alerting health-care professionals and relatives in an event of emergency [28]. Such usage of robots in the field of health-care provides assistance to medical professionals. These ideas encapsulate the motivation for development of a novel framework and controllers for a nursing assistant robot.

## 1.2 Contributions

In this thesis we propose, a structured Directive Observer and Lead Assistance (DOLA) framework for the nursing robot to include protocols for both physical and non-physical HRI. Along with defining the framework, we also formulate and improve the constituting individual modules. Safety concerns are addressed for pHRI mode of operation. For non-physical HRI, command-based modules are developed. This proposed framework is detailed in Chapter 2

Neuro-Adaptive Controller (NAC) and Human Intent Estimator (HIE) is planned to be improved upon and implemented on the ARNA robot. The HIE module will be used during pHRI mode, where it will estimate and adapt to the user’s intent. The NAC will form the underlying controller for the robot which will adapt to the system’s non-linearities and ensure that the given task is followed with minimal error.

We formulate model-free full body collision detection for robotic manipulator and implement it on a 6-DOF robot arm by placing a force-torque sensor at its base. We also address the issue of decoupling the wrenches experienced due to system dynamics and collision. Using this configuration, we develop a Base-Sensor Assisted Physical Interaction (BAPI) controller.

We propose and implement a Human-Machine Interface (HMI) on a tablet to communicate with the robot. The HMI uses a traded control based interaction which allows the user to decide high level tasks without having to constantly monitor the robot’s motion during task completion. The HMI is capable of accepting voice commands as well as on-screen button commands to initiate an intended task. This HMI is a part of the non-physical HRI mode of the proposed DOLA framework.

We develop a tactile sensor package simulation module to help future manufacturing of optimal sensors. The FEA analysis of this sensor package is done using COMSOL software and the results are validated using experimental data. A Robot Operating System (ROS) based simulation environment called SkinSim is also improved to account for simulation of

large tactile skin patches integrated with robots and their control systems.

We propose use cases for testing the robot and validating its systems in a healthcare environment through in-depth investigation. These use cases are to be evaluated for the patient walker and patient sitter scenarios. The planned studies explore the operational capability of the robot. The use cases encapsulate the situations that the robot may face and test its system’s effectiveness. Proper metrics are also defined for standardized tabulation and comparison of results during the course of the experiments.

The contributions of this thesis are summarized below:

1. The DOLA Framework

- The DOLA framework defines the roles and behavior of the underlying modules and components of an assistive robot. The framework adapts to different levels of autonomy as described by Beer et. al. [43]. The various levels of autonomy dictate the robot’s behavior during any HRI process. The introduction of an Observer block allows the robot to continuously monitor and estimate the user intent. The use of neuro-adaptive controllers facilitates a model-free smooth and safe HRI.

2. Evaluation and validation of HIE and NAC

- The structure and functionality of NAC has been proposed by Ranatunga et. al. and Lewis et. al. [16,17,44–46]. It has been further improved upon by addition of HIE and prescribed error dynamics (PED) by Cremer et. al. [5]. This controller contains two loops - an inner loop, which is responsible for trajectory tracking and generating required torque signals, and an outer loop, which is responsible for defining the desired trajectory based on HRI and HIE. These loops contain neural networks which assist with learning the robot and human dynamics and makes the controller model-free. The controller was deployed and experimentally validated on the PR2 robot platform through user studies.

3. The BAPI controller

- The BAPI Controller uses a single force-torque (FT) sensor attached at the base of a robot arm to detect collisions with a human user. The collisions can be detected through the entire robot body due to the fact that the FT sensor at the base, or the Base Force-Torque Sensor (BFTS), can detect all the forces at the base caused due to collisions as well as the movement of the robot. The key contribution here is to decouple the system dynamic forces at the base, caused by the movement of the robot, and the forces due to collision with the human-user.

#### 4. Human-Machine Interfaces (HMIs)

- A robot needs an interface to communicate with the human user. For each level of autonomy, there is a different HMI to be used. The performance of a user in issuing commands or controlling the robot may also depend on the level of expertise of the user. A research study by the author and team, found that using a traded control level of autonomy allows both expert and inexperienced users to perform similarly. This also sets the level of autonomy that can be used during the robot functions as the user will be a patient and may not be an expert in controlling the robot. This level of autonomy was achieved through the use of a tablet-based interface.

#### 5. Modeling tactile sensor packages

- A tactile sensor package consists of a sensor and an artificial skin encompassing it. The behavior of this sensor package is different from the sensor itself [1]. The artificial skin can be modeled as an extra layer of mass-spring-damper system which takes into account the elasticity of the skin as well as the spread of applied forces. The sensor package needs to be accurately characterized in order to find the relation between the sensor output and applied forces. Post characterization, these sensor packages can be used on robots to provide feedback to controllers as well as act as HMI during collaborative tasks between the user and the robot.



### **1.3 Thesis Outline**

Chapter 2 introduces the proposed novel DOLA framework and chapter 3 describes the nursing assistant robot used for deploying all the developed modules. In chapter 4, the work on sensors and human-machine interfaces are presented. Chapter 5 illustrates the work on adaptive controllers and chapter 6 contains the formulation of the BAPI controller. Chapter 7 contains the user studies conducted with the ARNA robot. This chapter describes the experiment setup and the protocols used to test ARNA in a hospital environment. Chapter 8 presents our conclusions and discusses future work for the research.

### **1.4 List of Publications**

The list of publications is enumerated in Appendix A

## CHAPTER 2

### DIRECTIVE OBSERVER AND LEAD ASSISTANCE (DOLA) FRAMEWORK

The quality of Human-Robot Interaction is intertwined with the level of automation of the robot as well as the nature of task being performed. A single task may require different levels of assistance from the robot for different users. Also, the safety measures [9] have to be ensured during all interactions. A novel framework, Directive Observer and Lead Assistant or DOLA, is introduced in this chapter which includes the required flexibility in automation as well as adapts to user specifications.

A study by Beer et. al. defines a 10-level taxonomy for the “*Level of Robot Automation (LORA)*” in the context of human-robot interaction [43]. They also define a framework consisting of 5 guidelines for choosing the level of autonomy which depends on the tasks, environment and the level of tasks that the robot can perform. Huang et. al. also have explored the idea of classifying the measure of autonomy in a robot [47]. This study focused on defining three dimensions which were required to access the level of autonomy in a robot. Another such article by Thurun explores the components for building a framework for HRI operations [48]. Components such as the classification of the robot, level of autonomy and the various human-machine interfaces were explored in the article [48]. A task specific framework was proposed by Wang et. al. for robotically-assisted minimally invasive surgery [19]. This proposed framework contains four components for performing a vision-enabled surgery. Other studies such as by Kapoor et. al. have focused on defining the software modules and framework for computer integrated surgery [18].

There seems to be a requirement for a holistic framework which encapsulates the HRI process, taking into account the safety concerns and standards during HRI with varying level of autonomy. Different users will have different specification during any HRI process,

such as difference in speed, strength and accuracy between users, which will affect the outcome of the task completion. This calls for agility in a framework to ensure that the tasks are completed as accurately as possible while adapting to user parameters. An efficient framework should also be able to adapt to wide range of autonomy during HRI. This provides the user with choices for the kind of HRI they are comfortable with.

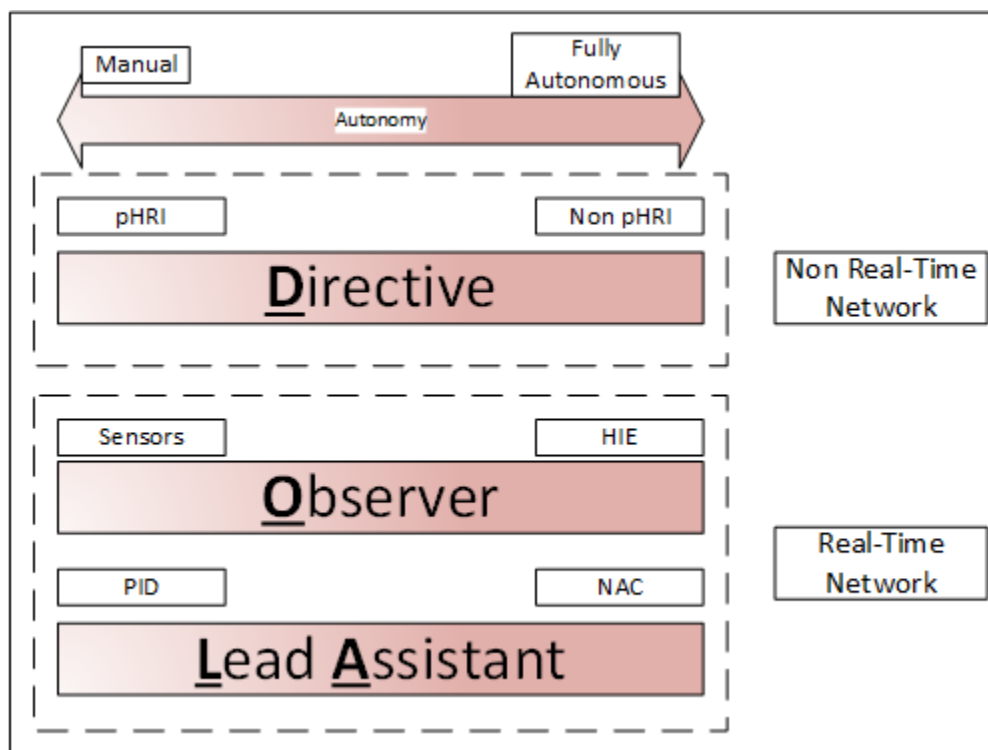


Figure 2.1: DOLA framework incorporating the entire autonomy range

Figure 2.1 shows the three modules of the framework incorporating different levels of autonomy as discussed by Beer et. al. [43]. The DOLA framework contains three functional blocks:

- Directive: The Directive block defines the user instructions or directives to be followed during any HRI process. The details of the tasks to be performed are stored in this functional block. The data from the human-machine interfaces are included in this as the user commands for the HRI.
- Observer: During HRI, the Observer block has to monitor/observe the user param-

eters and try and estimate the user intentions. This serves two purposes, safety and adaptability. Due to constant monitoring in real-time, this block would be able to mitigate any collision and with human-intent estimation (HIE), the user intent would be known for the robot to modify its behavior for ease in usability.

- Lead Assistant: This block contains the controllers for the robot. A novel neuro-adaptive controller (NAC) is proposed to be used which contains neural networks and is model-free. The use of neural networks in the controller allows it to modify the output based the user parameters as determined by the Observer block.

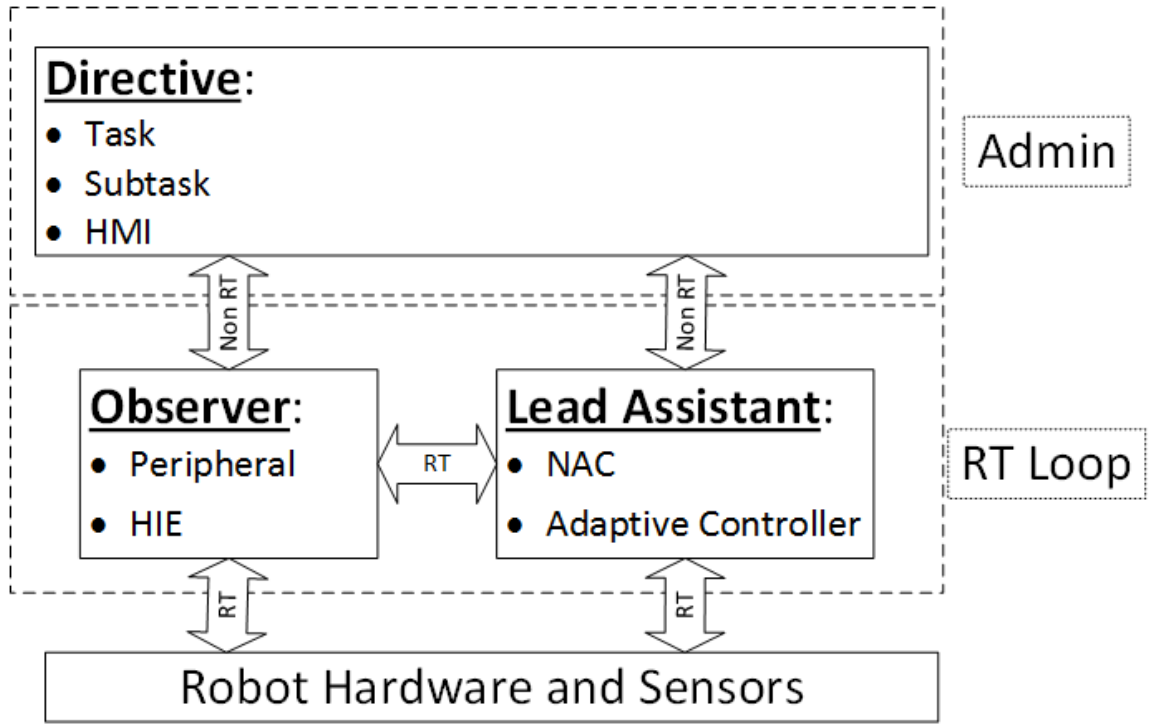


Figure 2.2: DOLA framework

## 2.1 Components

Before describing the architecture of the framework, it is important to describe the components included in the framework. These components make up the three major block of the proposed framework and understanding these components would clarify their placement

in the framework. Apart from their placement, their usage has to be looked at from the perspective of the level of autonomy required for any given case. The required level of autonomy would decide the need for each constituting component and sub-components.

In general, any interaction (pHRI or non-pHRI) with a robot system requires the components described in the following subsections. These components are an extension from article by Thrun [48].

### **2.1.1 Task Model**

This defines the task model or the tasks that is to be performed during the interaction process. Due to the complexity of HRI and the difference of expertise between users, a sub-task model is also included in the task model. The sub-task model along with the task model can define the autonomy of the robot during HRI. The advantage of having a sub-task module is that for the same task model different levels of autonomy can be defined without having the need to define multiple similar task models.

The scope of task and sub-task models are defined by an admin which also reflects the capabilities of the robot hardware in question. In case of pick and place tasks, a robotic arm with varying degrees of freedom can be used. But the scope of sub-tasks is limited by the range and function of the robotic arm. These limitations are known to an expert or the admin of the system, therefore the admin is responsible for design of such task models which can be achieved by the particular robot system.

Thus, this component acts as the overall governing mechanism during HRI. The task models can be changed by the user through an interface on-the-fly, and the robot has to be able to adapt to the change in directives.

### **2.1.2 Human-Machine Interfaces (HMI)**

The human-machine interface as the name suggests, facilitates communication between a user and a robot through some type of interface. There are various HMIs that are typically used during HRI such as tablet, joystick, gestures etc. The HMI is the conduit

through which the high-level decision taken by the user passes on to the control systems of a robot.

In the instances of pHRI, sensors like force-torque sensors and tactile sensors act as HMI. These sensors collect the user-data such as direction and intensity of user forces which are then used by the robot system to adapt to these parameters, making the HRI process intuitive and effective. Another such example is the tele-operation of a robot where typically a joystick or a tablet is used as an interface which collects user data. Examples of other interfaces that are used during HRI are voice and gesture interfaces.

From the perspective of autonomy, the same interface can have different roles during different levels of intended autonomy. For example, during tele-operation a tablet can be used for collecting continuous user-data to manipulate a robot system, but during traded control approach, the voice commands can be used to set the goal for the robot system and the autonomous control of the robot ensures that the task is completed. This highlights the changing roles of the same component under different task-models.

### **2.1.3 Network**

Network selection and implementation has a huge impact on how data is transferred and with what precedence. The network selection, in the context of proposed framework, can be divided into two major parts - real-time (RT) and non real-time (non-RT) networks

The RT network makes sure that the important data and functions are given higher precedence. In a RT network the timings of data transfer are deterministic which guarantees zero data packet-loss. Such networks are typically used in the low-level control systems. In contrast the non-RT networks facilitate data-transfer in a non-deterministic way. This type of network is typically used in robot systems to accomplish non-essential tasks such as communicating directives from users to the robot. The RT network on the other hand can be used for collision avoidance which is essential for safe pHRI.

A complete autonomous system requires the use of a RT network to sense and operate in an environment shared by human user. In between the autonomous tasks, if new user

directives are to be communicated, it has to be done through non-RT network as the robot would continuously require environment data through RT network to avoid collision. On the other hand, during tele-operation, the user data from the interfaces or sensors need to be transferred using RT network continuously in order to drive the robot safely. Any delay in updating the user data can cause collision. This emphasizes how the use of RT and non-RT networks are based on the selected task model / autonomy.

#### **2.1.4 Controllers**

The controllers are the brain of a robot system. The proposed framework incorporates adaptive controllers. The novelty in using adaptive controllers is that it is agile enough to adjust to different users and their parameters. The proposed framework contains a neuro-adaptive controller or NAC which utilizes neural networks to calculate control output. This controller is also model-free, which means it is not dependent on the robot system and can be ported to other robots easily.

A human-intent estimator or HIE module is also used to estimate user intent and change the control outputs accordingly. The HIE module allows the robot to work with the user and react to the user's grip strength or speed to keep up with them. The advantage of using model-free control systems is that the framework can be implemented on any robot system without changing or modifying it every time.

#### **2.1.5 Sensors**

Sensors are an integral part of any robot. They gather environment data and are used as feedback data or commands to the control system. The sensors can be broadly classified into peripheral sensors and HMI sensors. The peripheral sensors, such as Lidar, camera, IR-sensor, ultrasonic sensors etc., are used for sensing the environment surrounding the robot. The robot is able to make sense of the workspace and completing the task at hand using these sensor data.

The HMI sensors, such as force-torque sensors and tactile sensors, are used as pe-

ripheral sensors as well as HMI depending on specifics of a task model. This brings us to the relation between different levels of autonomy and the sensors on-board. During autonomous operation, all sensors act as peripheral sensors, informing the controllers of any obstacles to avoid or if any collision took place. Whereas during pHRI, the HMI sensors act as interfaces and collect user data for intent estimation or driving the robot.

This shows the need of having flexibility in using sensors as per the task model selected by the user.

### **2.1.6 Robot Hardware**

Robot hardware informs the kinematic and dynamic model of the robot. According to Thrun, robots can be classified into industrial robot, professional service robot and personal service robot based on the environment the robots are intended to be used in as well as the tasks they are expected to perform [48]. The type of robot selected for a specific use, informs the task models that it will be able to perform. An industrial robot will not be well equipped for service in a home environment and a service robot won't be able to lift the heavy objects in an industrial task.

These classification and abilities of the robot are used to formulate the task model as well as sub-task models.

## **2.2 Architecture of DOLA framework**

Figure 2.2 shows the architecture of the proposed framework. It illustrates the three blocks in an object-oriented fashion. The framework employs a state-machine based architecture where the RT loop is continuously running and changes its state based on incoming user directive.

The Directive class contains three functions representing the components discussed in the previous section - tasks, subtasks and HMI. The task and sub-task function are populated/maintained by an admin who is the expert user of the robot system. These are then presented to an end-user who would choose the type of task set to perform. The chosen



task and sub-task modules calculate and sets the level of autonomy intended by the user. This selects the choices for the type of HMI to be used during the task. The Directive class is capable to receiving user instructions at any given time and these instructions are passed down to subsequent systems through a non-RT network.

The Observer class is responsible for continuous peripheral monitoring and estimating user intent during robot operation. This is one of the novelties of the proposed framework as this class is responsible for safe-HRI. This class is contained in a RT loop which allows it to run continuously even in the absence of any input from Directive class. Another feature of this class is that it contains the peripheral function which gathers data from peripheral sensors. This allows it to override any ongoing operation and send emergency signal through RT network to the robot to halt in case of impending collision. This illustrates the idea that even when the robot is void of any task goals and operations, the Observer class is still active and is able to take necessary actions in case of emergencies.

During robot operation, the Observer class also uses the HIE function to predict human-intent. This is later used in the Lead Assistant class. This HIE function imparts the intuitiveness that a robot system needs during pHRI. This facilitates the system's ease of use.

The Lead Assistant class contains the controller functions such as NAC and BAPI. These are neural-network based controllers and their formulation is discussed in detail in Chapter 5 and 6. The use of such controllers in this framework makes it adaptable to user's mannerisms. This class communicates with the robot hardware through a RT network and drives the robot towards a given goal.

The Observer and the Lead Assistant class are encapsulated in a RT loop as a part of the state machine. Any user directive is received from the Directive class including the HMI definition. The Observer class monitors the peripheral data from sensors and estimates user intent using the HIE function. The controllers in the Lead Assistant class receives these information from the Observer module as well as Directive module. The task information from Directive module guides the controller goals. Along with the Lead Assistant module

the Observer module also has the authority to issue a halt command in real-time to the robot in case of obstacles and collisions during the tasks.

The uniqueness of the proposed framework can be summarized as below:

- The continuous operation of the Observer module ensures the safety guidelines [9]. This ensure a safe pHRI and enables the system to be deployed in the same environment as users.
- This is a holistic framework that can be deployed on any service robot with the current intent of deploying the framework on a nursing assistant robot.
- The introduction of Observer block imparts versatility to the entire system as it both monitors and tailors robot performance in accordance to user operation.
- Another advantage of using this framework is the flexibility of choosing different levels of autonomy during any HRI process.
- The neural-network based controllers make the framework model-free and can be implemented on any robot system.

## CHAPTER 3

### ADAPTIVE ROBOT NURSING ASSISTANT

Healthcare robotics is expected to grow in demand in near future. A recent report from Association of American Medical Colleges (AAMC) projects a shortfall of about 21,000 to 55,000 primary care physicians [49]. It has also been observed that Advanced Practice Registered Nurses (APRNs) and Physician Assistants (PAs) are able to provide the same healthcare services as a physician, thus assisting with some of the shortages in the physician workforce [49–53]. With these projected increase in the work load, use of robotic assistants is an effective way to lower the burden of the care givers.

These assistive robots fall in the category of service robots which may be required to have physical contact with users. This chapter describes such a robot named Adaptive Robot Nursing Assistant or ARNA that was built for the purpose of assisting nurses in a health-care environment. The ARNA robot is designed to assist nurses with routine tasks which will ease their workload and automate routine tasks. The subsequent sections describe the build and the functional capabilities of the robot.

#### 3.1 Robot Architecture

Figure 3.1 illustrates the ARNA robot which is an omni-directional mobile robot equipped with a 6-DOF robotic arm. The robot’s dimensions are 45in  $\times$  45in  $\times$  4ft. It contains a wide range of sensors to detect and perceive the environment around the robot. HMIs are also defined for this robot to help users interact with the robot easily.

The robot needs to navigate the small hospital room as well as in hospital halls which is populated at times. This requires extra precaution for safe interaction with the robot, especially when a patient is interacting with it. For pick and place tasks, the robot

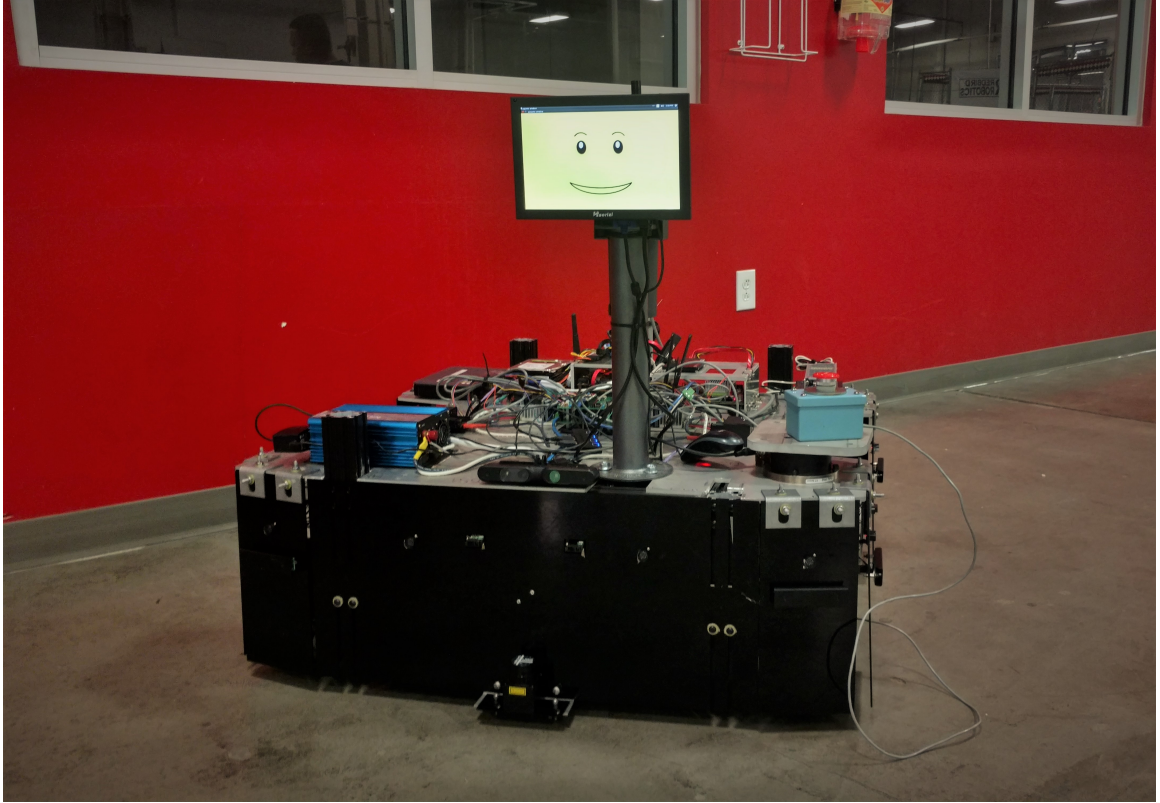


Figure 3.1: ARNA Robot

has to perform the tasks in a cluttered and unstructured environment which is far from the structured environment where industrial robots operate.

The robot uses Robot Operating System or ROS to operate and run. ROS is an open-source framework which allows researchers to use modules that has been already developed by others and utilize them in a new custom scenario. The advantage of using ROS is that, it provides many generic libraries that can be easily utilized for development purposes. ROS also supports distributed computing which is a huge advantage while operating a robot with multiple computing platforms. For these reasons ROS was chosen to be the underlying software structure upon which the DOLA framework, described in Chapter 2, can be implemented.

### 3.1.1 Components

As the robot is required to operate in unstructured and unpredictable environment, it needs to be equipped with appropriate components and sensors. These components are installed on the robot to serve their purpose based on the tasks that the robot is required to perform. The functions that the robot is expected to perform are detailed in section 3.2. This section describes in detail the installed components on the robot:

**Navigation** The robot's autonomous movement requires navigation through sensing. For navigation, ROS offers simultaneous localization and mapping (SLAM) packages which use GMapping to sense and build the map of the environment the robot is working in. SLAM requires a LiDAR and an RGB-D camera for operation. ARNA robot is equipped with a Hokuyo LiDAR which serves this purpose.

A sensor suite consisting of ultrasonic sensors and bump sensors are also installed on ARNA as shown in fig.3.2. These sensors provide additional 360° information about obstacles that may not be visible to the LiDAR which is attached to the front side of the robot. Information about the sensor suite is further detailed in subsequent paragraphs.

**Pick and Place** For pick and place tasks the robotic arm needs to reach into its workspace and pick objects using the gripping end-effector. A vision sensor or RGB-D camera is required to be used in such cases. The ROS based Object Recognition Kitchen (ORK) uses an RGB-D camera to build a 3D map of the environments and plans the path for the manipulator to avoid obstacles and identify the requested object. An RGB-D camera is a camera equipped with depth sensing along with capturing normal 2D pictures. The robot uses an Asus Xtion PRO camera for this purpose.

**Environment Sensing** Along with the above-mentioned sensors, there is a need for an additional redundant sensor system for early warning and emergency stopping system. Figure 3.2 shows the sensor suit that has been installed on the robot. The robot is equipped with 12 ultrasonic sensors distributed around the robot for sensing any approaching obsta-

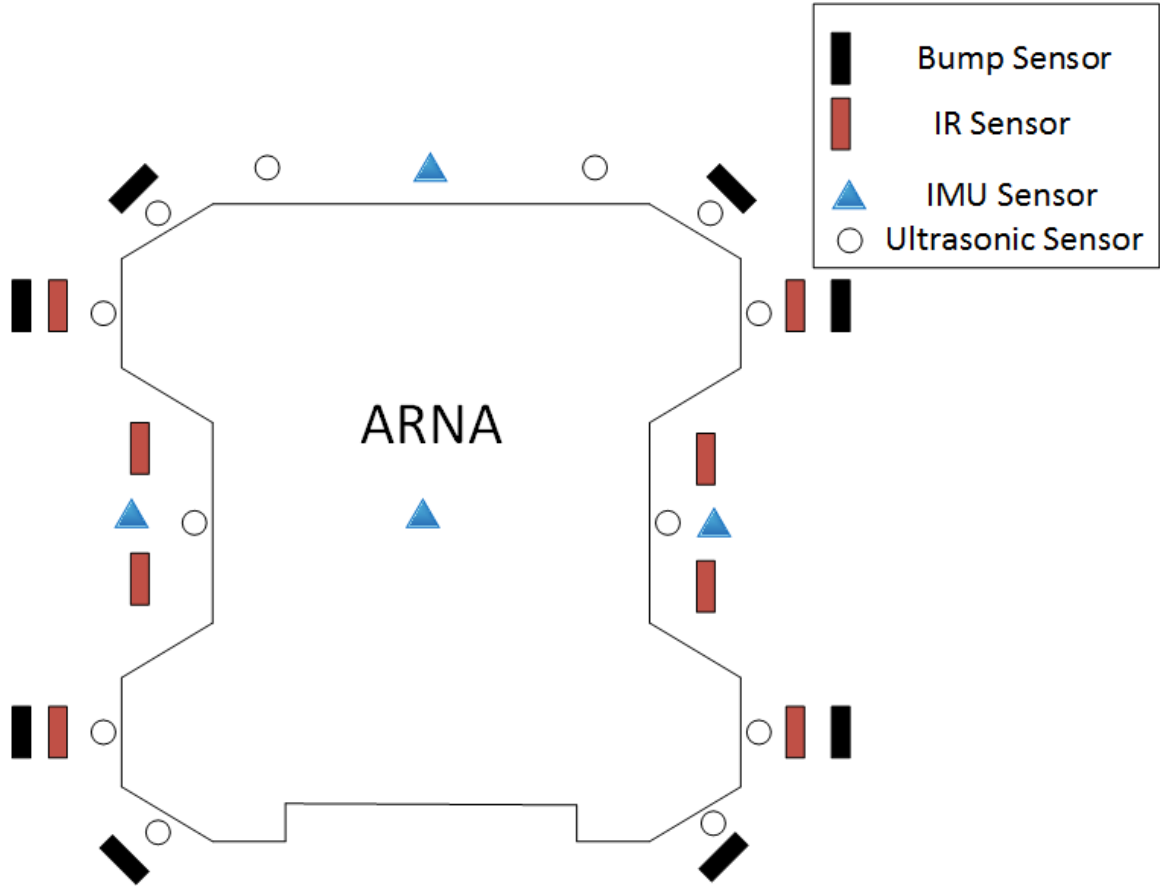


Figure 3.2: ARNA Sensor Suit

cles. The IR sensors act as level sensors which informs the robot of an incline plane and thus requires more power to drive.

The bump sensors act as emergency halting system for the robot. In the eventuality that any obstacle was missed during the sensing process, a soft collision with the bump sensor will immediately halt all operations on the robot.

**Detecting Human Activity** In order to detect human activity with the robot, sensors are installed to facilitate the interaction. A force-torque sensor, Axia80 model from ATI-IA USA, is affixed under the handlebar which senses the user interaction forces. Another force-torque sensor, Delta model by ATI-IA USA, is installed under the 6-DOF arm of the robot. This is used as a base force-torque sensor which is described in Chapter 6. This configuration is used for detecting collisions and user interaction forces on the arm.

**HMI** Three devices are used as HMI to facilitate user interaction with the robot. A joystick is used for tele-operation purposes, the force-torque sensor equipped under the handlebar as described earlier and an android tablet is used to develop and deploy apps to connect to the robot and issues commands through on-screen buttons and voice commands.

### **3.2 Robot Functions**

A study conducted by Cremer et. al. identified two areas of robot function as a nursing assistant, namely - walker and sitter [54]. The following subsections describe these tasks in detail which are also requirements for the ARNA robot.

#### **3.2.1 Sitter Scenario**

The sitter scenario describes the routine tasks that are performed while a patient is resting in a hospital room. This is known as a sitter task because as of now hospitals hire people to sit in the same room as the patient and monitor their vitals. They are responsible for alerting the nursing staff in case of emergencies. The personnel themselves cannot administer any medications as they are not certified for it. The required tasks come under the purview of non-physical HRI as the robot is not supposed to be in contact with the patient during this scenario.

These tasks were identified as sitter task and experiments were conducted through the use of the ARNA robot. The challenges of the robot in such a scenario is to move around in the hospital room while avoiding the medical equipment. The hospital rooms are small and cluttered which poses the problem of autonomous navigation in an unstructured environment. This also includes detection of people other than the patient in the room.

Monitoring the patient and detecting if they are getting off of the bed would require maintaining continuous visual contact with the patient. This can be achieved through the use of RGB-D camera as the depth sensing would be useful in skeletal tracking. Upon detection, the robot has to alert the nursing staff and issue verbal commands to the patient. Thus, the position and orientation of the camera has to be always pointed towards the

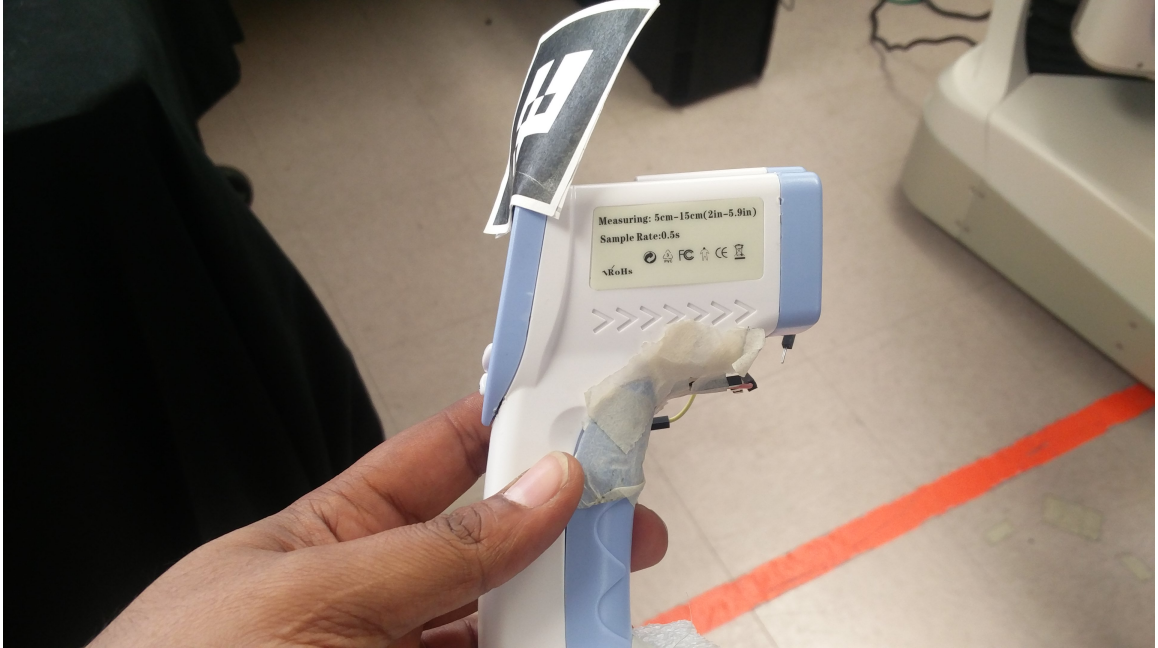


Figure 3.3: Digital IR Thermometer

patient while performing any other tasks.

While positioned in the room, patients might require objects that are far from their reach. The ARNA robot has to fetch these objects and safely place them near the patient. These commands can be issued through a tablet which acts as an HMI. The commands can be issued verbally as well as through on-screen buttons. Upon the issuance of the command, the ARNA robot has to navigate the room and locate the requested item. Then a pick-and-place task is performed while manipulating the mobile base to get the requested object in the work-space of the robotic arm.

While a patient is resting in a room, periodic measurements of their vitals are needed to track the progress of their health. As these tasks are non-physical based interactions, the choice of sensors should be such that the ARNA robot can take readings without directly touching the patient. Digital and non-intrusive medical equipment were chosen to equip the ARNA robot. For example, an IR based digital thermometer, as shown in figure 3.3, can be used to take temperature readings of a patient from a safe distance. This thermometer was also modified to be integrated with the robot electronics in order to trigger the readings autonomously.



To summarize, the list of tasks expected from ARNA in a sitter scenario are as follows [54]:

- Visual monitoring of patient and alerting nurses in case the patient is getting out of the bed or about to fall
- Taking vital readings periodically.
- Fetching items requested by the patient in the room.
- Ability to have a conversation with the patient.

### **3.2.2 Walker Scenario**

The walker scenario as the name suggests, involves assisting patients during walking. Apart from providing mechanical support to the patient during walking, the ARNA robot also has to carry IV poles and other necessary equipment. When the user is tired, the robot should be able to carry the person to a desired location. This is a pHRI based interaction where the patient and the robot are physically collaborating to perform a task.

During walking, the robot has to support the forces applied by a patient as well as sense the direction of forces. The direction of the applied forces would help in estimating the user's intended direction of movement. A force-torque (FT) sensor is used at the handle of the robot to sense these user forces. The choice of a specific FT sensor depends on two key aspects - sensor resolution and over-load capacity. The FT sensor's resolution decides the lowest amount of force that can be detected. Thus, the sensor resolution affects the sensitivity of the robot to forces applied by a user. The over-load capacity suggests the maximum forces that the sensor can undergo before braking. The over-load capacity in this case is the maximum force a patient can apply when exhausted. There is an inverse relationship between the resolution and the over-load capacity of FT sensor. With large over-load capacity, the FT sensors have lower resolution. The selection of the FT sensor for ARNA robot was done keeping in mind these variables.

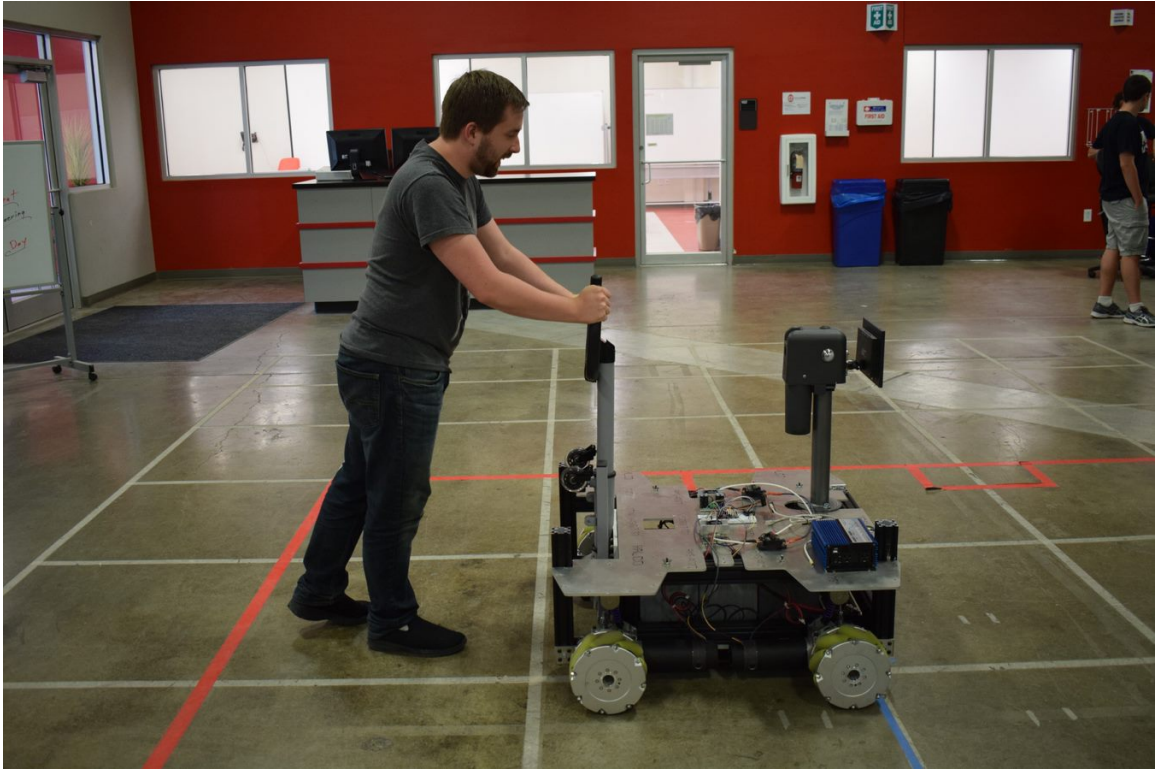


Figure 3.4: ARNA walker task



Figure 3.5: ARNA walker with patient carrying task

Figure 3.4 shows the walker task in progress. The varying levels of autonomy would decide the behavior of the sensor and the use of its data. For tele-operation based tasks, the FT sensor acts as a joystick and sends the direction of forces applied by the user and the robot moves in the resultant direction. During assisted walking, the FT sensor data is used by HIE to estimate the human intent as well as keep track of the desired path. The NAC uses this calculated information to adjust the speed and direction of the robot on-the-fly. Human users behave differently from each other from the perspective of their gait and speed. These behaviors also change for the same user between when they are tired and when they are well-rested. The NAC and HIE modules are capable of adapting to these changes in user variables and adjust course accordingly.

Another function of the robot is to carry the user to a location when the user is too tired to walk. As seen in figure 3.5, the ARNA robot is equipped with footrests and caster wheels for the user to hop on. The robot would then be able to carry the user to a desired location or it can be driving using the tele-operation mode. It can be noted that this change in autonomy can be achieved due to the DOLA framework discussed in chapter 2. This carrying task entails that both the footrest and the FT sensor should be able to support the entire weight of the user, should the user decide to lean on them to rest.

While walking, the patient may also require additional medical equipment such as IV poles or oxygen cylinders. The location of these equipment with respect to the patient has to be maintained at a certain distance. The robotic arm on board helps with this task. The required equipment, once put on a mobile platform, can be grabbed by the arm and then it would be able to drag them while maintaining a constant distance between itself and the patient. The BFTS attached at the base of the arm can detect the system dynamics and in-turn stabilize the arm accounting for the disturbances experienced. The BAPI controller would then be the perfect candidate for controller to implement these functionalities for the ARNA robot.

An important aspect of this scenario is the constraints on the motion of the robot. The use of mecanum wheels during the construction of the robot, allows it to be manipu-

lating in three directions — Translation in X and Y direction and Rotation in Z direction. The omni-directional nature of the mobile base also complements the need of moving in the user intended direction. When a user is about to lose balance or fall, the 3-DOF mobility allows the robot to make instantaneous course adjustments to support the user.

The roles and responsibilities expected from the robot in this scenario are as follows [54]:

- Assessing the patient’s gait and supporting them during walking.
- Carry essential equipment along with the patient.
- Providing necessary support when the user needs rest or slows down.
- Able to carry the patient when they get tired.

### **3.3 Discussion**

We discussed the ARNA robot and its function. These required functionalities form the basis for the design of user studies. Experiments were conducted at the School of Nursing, University of Louisville, KY. The experimental process as well as the results are described in chapter 7. The subsequent chapters in this document detail the research studies conducted on each individual component of the ARNA robot.

## CHAPTER 4

### SENSORS AND HUMAN-MACHINE INTERFACES

Sensors are installed on robot platform to provide information about the surrounding environment as well as help assist during pHRI. Along with the sensor suit, discussed in chapter 3, ARNA also requires tactile sensors and Human Machine interfaces for its operation. This chapter details the research studies conducted in the area of tactile sensor simulations and tablet interfaces.

Section 4.1 elaborates on the simulation studies conducted to create a finite element analysis (FEA) model of a tactile sensor which was manufactured by the Next Generation Systems research group. These simulation models are helpful as it provides an insight into the characteristic of the sensor before it is manufactured. As the cost of manufacturing these custom sensors can be staggering, these simulations provide a way to select efficient parameters. This tactile sensor will be useful to ARNA platform as it can be implemented on the handlebar to sense user forces. It can also be wrapped around the 6-DOF arm of the robot to develop full body sensing. This section details the research study published in [2].

Section 4.2 includes the study about selecting the right autonomy level to interact with ARNA. A tablet was selected as the interfacing device and preliminary experiments were conducted with traded control based autonomy level. The use of such autonomy level and interfacing devices are aimed at being deployed in a hospital with patients as the end-users. This section presents the research work published in [4].

#### 4.1 Tactile Sensors

As future robots are deployed in human environments, their requirements for safe operation during physical human-robot interaction (pHRI) has increased in significance [55].

Several studies have concluded that the magnitude of collision forces experienced during a human-robot interaction can be significantly reduced to safe levels by using flexible robot skin with embedded sensors [56]. For the last three decades there has been numerous researches on creating robot skins, from sensor modules for data acquisition and transmission [55] to creating full body tactile suits [57] [58].

Skin sensors are often times multi-modal, e.g. they implement several tactile detection mechanisms including pressure, temperature and vibration to estimate the unknown physical environment of a robot during pHRI. In order to obtain detailed tactile information, such as texture and hardness of a surface, hemispheric tactile sensors based on the hetero-core fiber optics have also been explored [59]. Capacitive touch sensors composed of silver nanowires (AgNWs) and polymers have also been demonstrated as tactile sensors [60]. An ongoing research area into capacitive touch sensors for robots is related to ultrathin capacitive touch sensor displays [61].

A popular transducer mechanism for robot skin sensors include piezoresistive strain gauges, implemented using metals and semiconductors materials. Metal piezo-resistors comprising of Ni, Cu, Cr, Mo, Fe alloy, Pt or W alloy have a gauge factor ranging from 2.1 to 4.1 [62], whereas Polysilicon membrane based sensors have gauge factors as high as 22 [62]. A higher value of gauge factor ensures higher sensitivity to applied strain.

In this chapter, we use a new type of printed piezoresistive strain gauge sensor comprising of thin layers of PEDOT:PSS or poly(3,4-ethylenedioxythiophene)-poly(styrenesulfonate). PEDOT:PSS was printed on interdigitated electrodes as opposed to other fabrication methods such as spin coating [63], laser ablation [64] or lift-off process [65] [66]. Prior studies have concluded that PEDOT:PSS preserves its piezoresistive properties even after the printing process [67] which make it an ideal material for a flexible, inexpensive process. Also, the gauge factor of PEDOT:PSS was reported to be  $17.8 \pm 4$  which is greater than the metallic strain gauges available commercially [68]. Since the printing was carried out using a PEDOT ink with a process known as Electro-Hydro-Dynamic (EHD) printing, our gauge factor is not fully known at this time, as a result, large numbers of printed sensors were

recently fabricated and experimentally characterized [69] [70].

In this paper we create and validate a finite element (FEA) model of our packaged sensors that closely resemble the geometry, materials, and their experimental testing conditions. The model includes a flexible Kapton layer which encapsulates the sensor, the printed PEDOT piezo-resistors, the inter-digitated electrodes, and additional package encapsulant material consisting of a cast silicone polymer skin base (P10) [1]. The paper reports on simulation results obtained using FEA simulation software COMSOL<sup>®</sup> during loading of an entire sensor package rather than simulating the sensor alone [63]. Simulations on the model were run for loads applied at multiple locations or loading zones on and around the sensor closely following the experimental measurements [1]. We analyzed the deformation, as well as corresponding changes in resistance for our sensors as incremental pressures are applied, and results were compared with experimental values for validation. Results indicate that the model is able to appropriately validate experimental trends for sensor resistance within 1% on all of the four loading locations on the sensor package. The model will be used in the future as an indispensable design tool for future robotic skin sensors.

#### **4.1.1 Sensor Design, Fabrication and Experimental Loading**

##### **4.1.1.1 Sensing Principle**

The piezoresistive polymeric strain gauges described in this work rely on the use of gold micro-patterned interdigitated (ID) structures, with a PEDOT:PSS thin films deposited on top. A typical geometry of our ID structures with overall lateral dimension of approximately 1.5 mm and thicknesses of approximately 1  $\mu\text{m}$  is shown in Figure 4.1.

PEDOT:PSS is a semiconducting polymer composed of conductive molecules encased in a polymer matrix. When this film is solid, the resistivity changes relative to its physical deformation. When this film is atop the ID structure, a conductive interface is formed between the patterned structures and the PEDOT:PSS. When the sensor experiences strain, its resistivity will change as a result of piezo-electric effect of PEDOT:PSS, as well as geometric changes in the distance between the ID fingers. The concept surrounding the

use of ID structures is simply that it is easier to reveal the piezoresistive nature of the PEDOT:PSS using a smaller distance between points of measurement. So, in the case of ID sensors, the higher the density of electrical contacts and the thinner the PEDOT:PSS films, the higher the sensitivity one can achieve. These criteria were taken into consideration when designing our sensors. Experimentally, we designed and tested strain gauges composed of varying ID finger width, length, spacing, and overall density per predefined area. Our gauges have incorporated fingers of widths and spacing from 10-100 microns.

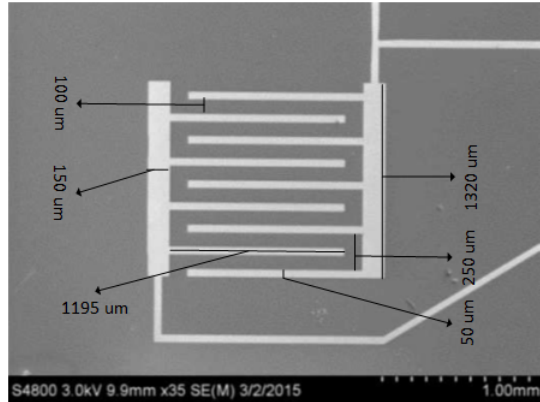


Figure 4.1: Interdigitated Structure [1]

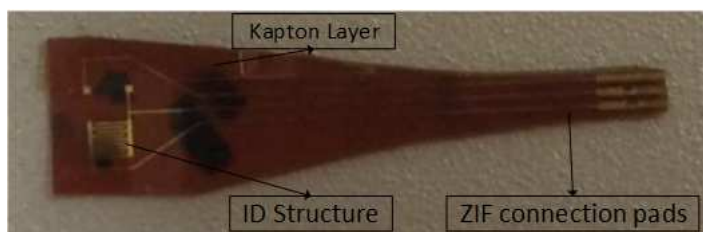
#### 4.1.2 Fabrication and Packaging

Standard clean-room photo-lithography lift-off processes are utilized to pattern the interdigitated structures. A sheet of 100  $\mu\text{m}$  thick Kapton sheet is cut to the approximate diameter of a 4 in wafer and adhered using photo resist to a Si wafer for processing to keep the sheet rigid through photo-lithography. After the sheet is backed by a wafer, a bi-layer resist is applied and prepped for exposure. The desired pattern is loaded to an aligner, along with the resist covered Kapton sheet, and exposed for a predetermined time to transfer the image to the resist. The resist is then developed, substrate dried, and Oxygen plasma cleaned for the CVD process. 250 nm of Gold is then sputter coated onto the substrate. Finally, the substrate is placed into a liftoff bath where the patterned structured is revealed and Si backing wafer deboned for later use.

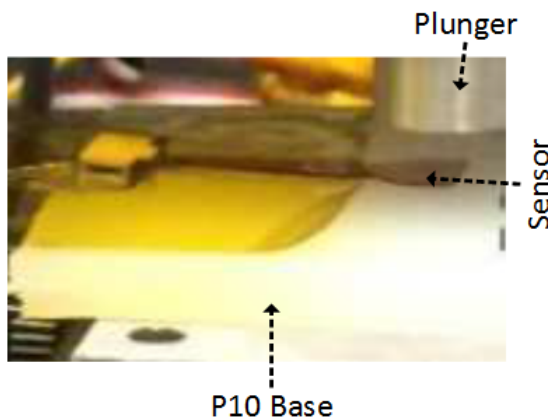
EHD printing was utilized to print PEDOT:PSS containing inks onto the Au inter-



digitated structures [70]. The ink is loaded into a reservoir and nozzle energized in such a configuration where the high voltage DC can flow from the conductive tip to the ID electrodes and subsequently jet the ink to print a pattern. The sensors are placed on motorized translation stages and coded so that the jetting occurs precisely in a predetermined path over the ID structure. Completed sensors are dehydrated in a convection oven and encased in Kapton. Additionally, ZIF connection pads are patterned with the sensors to easily connect the circuit measuring the sensor as shown in Figure 4.2a. Finally, the sensor is then placed on top of a P10 silicone polymer skin, shown in Figure 4.2b.



(a) Packaged Sensor



(b) Sensor with P10 Silicone skin bottom

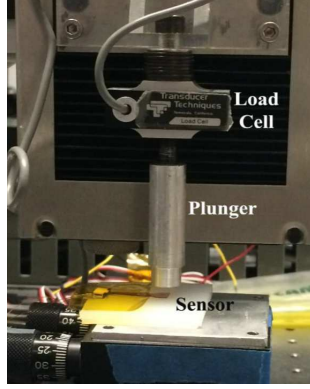
Figure 4.2: Sensor Package [1]

#### 4.1.2.1 Package Loading Experiments

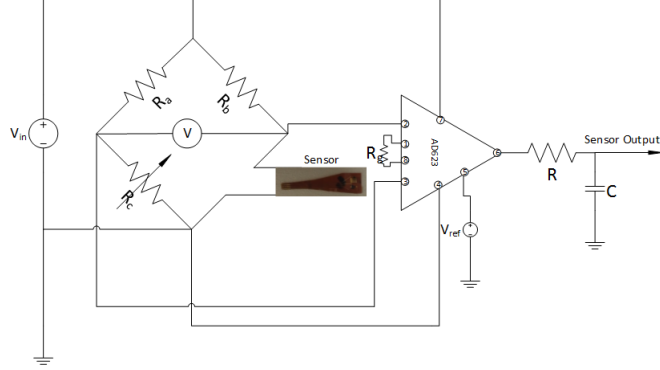
Experimental tests were carried out by applying force at different locations on and near sensors interdigitated structure with a controlled force sensing apparatus depicted in Figure 4.3a, 4.3b and 4.3c. The idea behind this test was to observe if there was any

significant change in sensors response in terms of spatial uniformity and linearity. By varying micrometric scales present at both X and Y stages, the sensors position with respect to the plunger was changed. The X-Y stage can be moved with a step resolution of 25 microns (0.0254 mm). Figure 4.3c shows how the force was applied at different positions. With respect to sensors position, force was applied on top of sensor (zone 0) as well in three different directions (zone 1, zone 2 and zone 3) as shown. In each direction, force was applied uniformly by engaging the plunger position at three different distances away from sensor. The yellow dot indicates an application of force on top of sensors interdigitated structure while red, blue and green dots indicate an application of forces at a distance of 1.27 mm, 2.54 mm and 3.81 mm, respectively, away from sensors interdigitated structure.

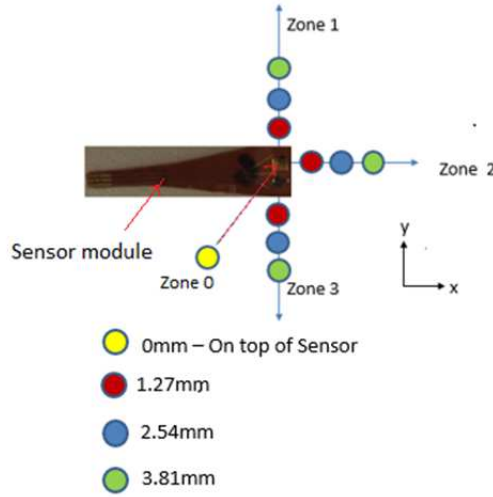
In order to measure changes in resistance due to loading, sensor integration circuitry was integrated with the sensors. A Wheatstone bridge along with an Instrumentation amplifier (Op-amp) was employed as shown in Figure 4.3b. When the bridge is balanced, the parallel resistance structures produce zero voltage difference between two parallel branches. In this circuit  $R_a$  and  $R_b$  are the fixed resistors at one leg and the strain-gauge sensor and variable resistor  $R_c$  is attached at the other.  $R_c$  is used to balance the circuit at no load. To amplify and condition the output response from the Wheatstone bridge, an Instrumentation amplifier and a RC filter were employed. This filter helps eliminating any high frequency components or noise. After passing through filter, we can measure the sensor output voltage.



(a) Experiment Setup



(b) Sensor with P10 Silicone skin bottom



(c) Force application at different loading zones



(d) Flat Plunger

Figure 4.3: Experiment Components [1]

Figure 4.3d show the plunger that was used for the purpose of application of force on the setup. Figure 4.3a illustrates the experimental setup where the plunger is connected to the load cell. The readings from the load cell indicates the amount of force applied to the sensor package. The sensor is fixed on top of the P10 polymer silicone skin using Kapton sheets and the setup is ensured to be fixed so that there is no movement while the experiments are being performed.

### 4.1.3 Sensor Package Modeling

In this section, we describe the simulation model of the sensor package. This includes the geometric specifications, material properties assigned to each constituent entity and the simulation solver settings.

#### 4.1.3.1 Package Geometry

The geometric specification from the experimental samples were used to create a resembling model in COMSOL<sup>®</sup> finite element software. Figures 4.4a, 4.4b, 4.4c and 4.4d show the constituent entities of the sensor module, while Table 4.1 summarizes the approximate dimensions of each component.

To build the interdigitated structure, a CAD model was imported, and its thickness was assigned to be  $1\text{ }\mu\text{m}$ , and shown in Figure 4.4a. The PEDOT:PSS piezoresistive layer was then defined such that it covers the interdigitated structure, with overall rectangular box dimensions of  $1800\text{ }\mu\text{m} \times 1400\text{ }\mu\text{m} \times 1\text{ }\mu\text{m}$ , shown in Figure 4.4b. The sensor was then packed between two Kapton layers, implemented as thin rectangular substrates,  $100\text{ }\mu\text{m}$  thick, one layer on top and the other beneath the sensors. The assembled structure is then put through a union operation in COMSOL<sup>®</sup> to implement the sensor entity, shown in Figure 4.4c.

Finally, a polymer silicone layer (P10) representing the base substrate is added at the bottom of the sensor entity. The dimensions of the base were chosen to be  $1.5\text{ cm} \times 1.5\text{ cm} \times 4\text{ mm}$  as shown in Figure 4.4d. After the sensor package was assembled, four loading zones were defined by creating cylindrical structures with diameters similar to the plunger, which has a radius of  $5\text{ mm}$  and a thickness of  $1\text{ }\mu\text{m}$ . The center zones were created at a distance of  $2.54\text{ mm}$  which resembles to the blue zone shown in Figure 4.3c.

#### 4.1.3.2 Material Properties

After defining the geometry of the sensor package, material properties of each entity were defined. For the base, the material properties were not known, but it can be set in

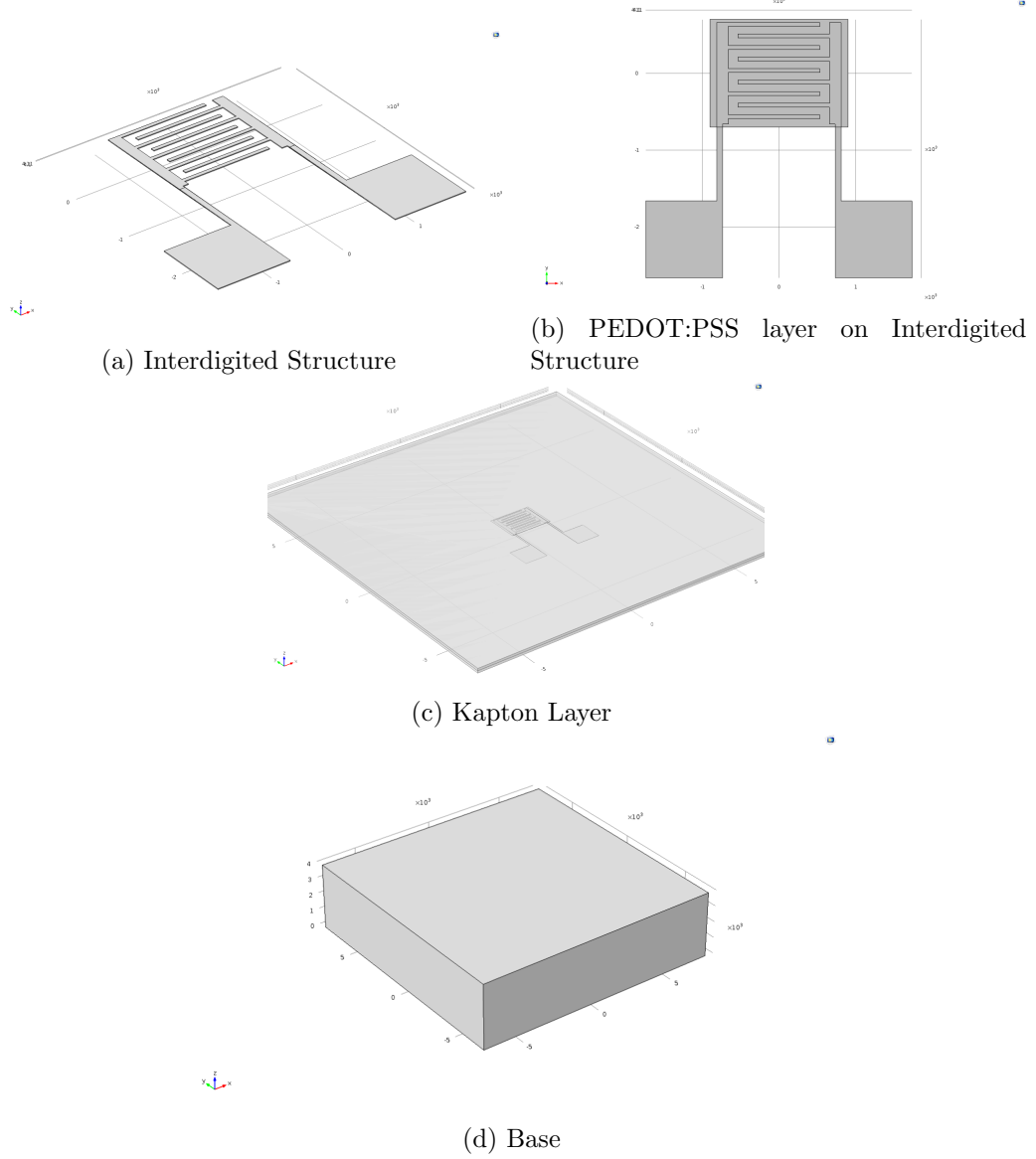


Figure 4.4: Geometric Entities [2]

a range of Young's moduli between 0.5 MPa and 50 MPa for silicone materials. For the simulation results reported in this paper a value of 5 MPa was initially used for our base layer formulation. After several simulation runs by loading the models loading zones with forces in the range of 1-6 N, the value of Young's Modulus was adjusted to 0.6 MPa to match load-displacement curves. The densities, Poisson's ratio and Young's moduli for each sensor entity are summarized in Table 4.2, which uses standard values for the three common materials.

TABLE 4.1

Dimensions of model entities

Entity	Length	Breadth	Height
Base	1.5 cm	1.5 cm	4 mm
Interdigitated Structure	1320 $\mu\text{m}$	1620 $\mu\text{m}$	1 $\mu\text{m}$
PEDOT:PSS	1800 $\mu\text{m}$	1400 $\mu\text{m}$	1 $\mu\text{m}$
Kapton	1.5 cm	1.5 cm	100 $\mu\text{m}$
Loading Zone		2.5 mm (Diameter)	1 $\mu\text{m}$

TABLE 4.2

Physical Properties of P10 Silicone Base, Gold ID structure and Kapton sheets

	P10 Silicone Base	Kapton Layer	Gold ID Structure
Density	1.1 g/cm <sup>3</sup>	1.42 g/cm <sup>3</sup>	19.3 g/cm <sup>3</sup>
Young's Modulus	$60 \times 10^4$ Pa	$20 \times 10^8$ Pa	$70 \times 10^9$ Pa
Poisson's Ratio	.47	.34	.44

Values chosen for PEDOT:PSS the piezoresistive layer are taken from research by Hussein Al-Chami [67], with a density of 1011 kg/m<sup>3</sup>. In COMSOL<sup>®</sup>, PEDOT:PSS was simulated by modifying the properties of a p-doped silicon based piezoresistive sensor. To model the piezoresistive properties of PEDOT:PSS, the piezoresistive coupling matrix of the material was defined as described below. In general, the relation between the electric field, E, and current, I, for a piezo-resistor can be defined as:

$$E = \rho I + \Delta\rho I \quad (4.1)$$

Where  $\rho$  is the resistivity of the material and  $\Delta\rho$  is the change in resistivity of the material due to geometrical torsion. The relation between the piezo-resistance tensor,  $\mu$ , and the resistivity can be shown as:

$$\Delta\rho = \mu\gamma \quad (4.2)$$

Where  $\gamma$  is the stress on the material. In a 3D simulation,  $\mu$  becomes a  $6 \times 6$  piezoresistive coupling tensor matrix which defines direct and cross coupling terms as:

$$\begin{bmatrix} \Delta\rho_{xx} \\ \Delta\rho_{yy} \\ \Delta\rho_{zz} \\ \Delta\rho_{xy} \\ \Delta\rho_{yz} \\ \Delta\rho_{zx} \end{bmatrix} = \begin{bmatrix} \mu_{6 \times 6} \end{bmatrix} \cdot \begin{bmatrix} \gamma\rho_{xx} \\ \gamma\rho_{yy} \\ \gamma\rho_{zz} \\ \gamma\rho_{xy} \\ \gamma\rho_{yz} \\ \gamma\rho_{zx} \end{bmatrix} \quad (4.3)$$

The piezoresistive properties of PEDOT:PSS is known to be  $\pi_l = -5.11 \times 10^{-10} \pm 13.7\% \text{ Pa}^{-1}$   $\pi_T = 3.45 \times 10^{-10} \pm 4.34\% \text{ Pa}^{-1}$ .  $\pi_l$  and  $\pi_T$  are the longitudinal and transversal piezoresistive coefficients [67]. This can be visualized in the Figure 4.5 [71]. These coefficients were used for modeling the sensor package.

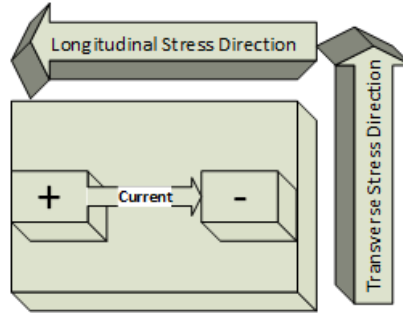


Figure 4.5: Longitudinal and Transverse Stress [2]

#### 4.1.3.3 Simulation Settings

Before simulations can be performed, a physics type, Piezoresistive Domain Current in this context, was selected in COMSOL<sup>®</sup>. The physics setting defines the areas pertaining to piezoresistive elements, the load boundaries and limits, the positive and negative terminals as well as the voltage supplied to the terminals. After the physics settings for the

experimental set-up is defined, the model is then meshed as shown in Figure 4.6b.

As shown in Figure 4.3c, four loading zones were identified for applying the force in the region. During simulations, a force from 0N to 6N was applied with an increment of 1N and the change in resistance was noted at each iteration. A constant voltage of 3V was applied to the interdigitated structure and the change in current was tabulated. Resistance changes were then calculated using Ohms law and compared with experimental results.

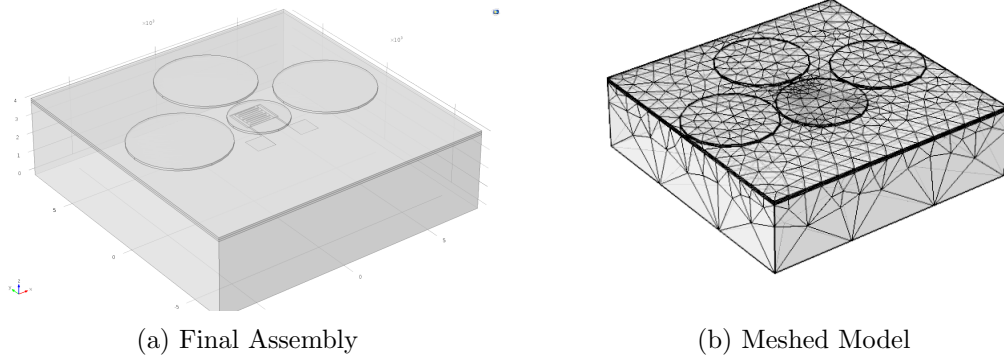


Figure 4.6: Final and Meshed model [2]

#### 4.1.4 Results and Comparison

In this section we discuss the results obtained from the simulations and compare them to experimental data. Figure 4.7 shows the displacement at the area where the force was applied, e.g. in zones 0 through 4. In each case, we ran 6 simulations by applying increasingly larger forces. Both simulation and experimental results indicate that when pressure is applied in zones 2, the resistance of the sensor increases with increase in pressure, whereas in zones 0,1 and 3, the resistance decreases as the pressure increases.





In Figure 4.8, we see that the simulated resistance values closely follow the experimental ones, and the strain-induced variation in resistance of the order of  $0.007 \text{ } \Omega/\text{N}$  in zone 0,  $0.012 \text{ } \Omega/\text{N}$  in zone 1,  $0.017 \text{ } \Omega/\text{N}$  in zone 2,  $0.011 \text{ } \Omega/\text{N}$  in zone 3.

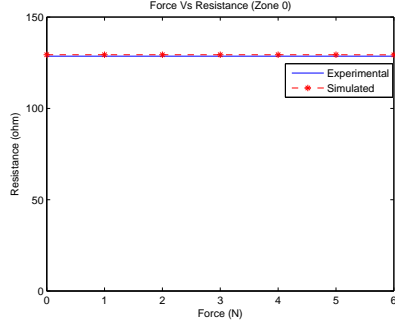
From displacement plots in Figure 4.9 we see that the displacement values in zones 1-3 are almost identical, and slightly different than the ones in zone 0. Qualitatively the simulation and experimental displacement curves are different, suggesting that the material nonlinearities of P10 may require additional tuning. This is not surprising, considering that silicone polymers are highly nonlinear, viscoelastic materials that have been approximated by quasi-elastic materials in our COMSOL<sup>®</sup> simulation.

In Table 4.3, we summarize the relative error between the simulated and experimental resistance and displacement measurements averaged over the entire set of pressure sweeps. We can conclude that our model will predict resistance changes as a function of stress with sufficient accuracy to be used in future sensor design evaluations.

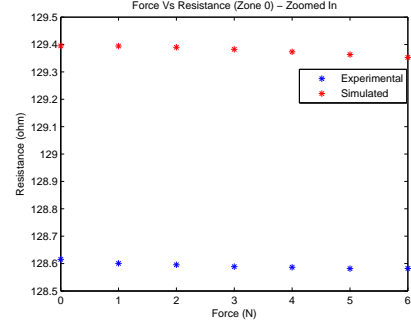
TABLE 4.3

Percent Error between Simulated and Experimental Data for change in resistance

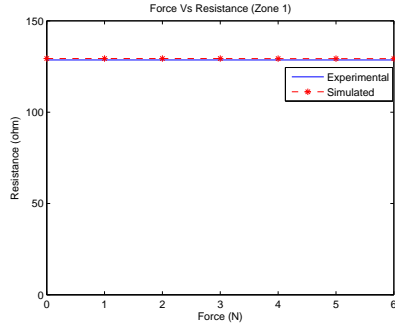
<b>Zone</b>	% Error in resistance	% Error in Displacement
Zone 0	0.611%	43.2%
Zone 1	0.5885%	25.13%
Zone 2	0.6191%	20.29%
Zone 3	0.5972%	24.95%
<b>Average</b>	0.604%	28.39%



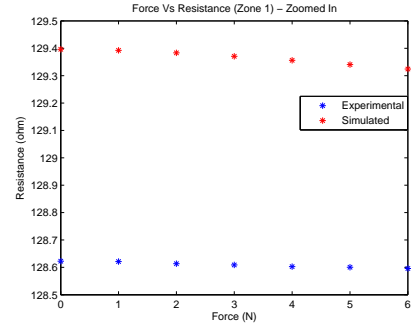
(a) Force Vs Resistance at Zone 0



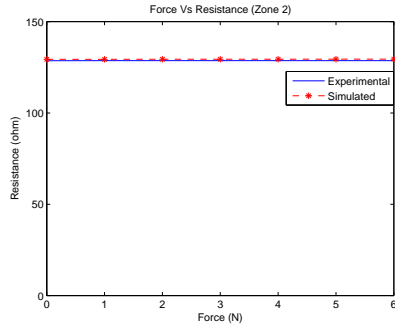
(b) Force Vs Resistance at Zone 0 - Zoomed In



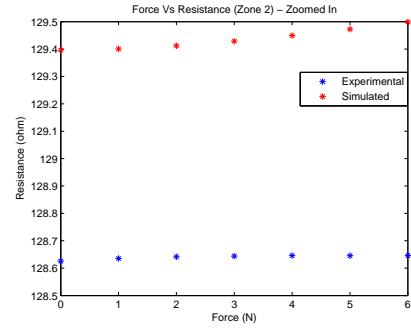
(c) Force Vs Resistance at Zone 1



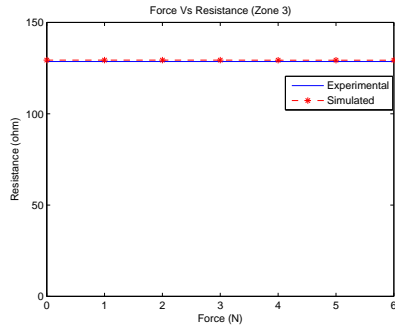
(d) Force Vs Resistance at Zone 1 - Zoomed In



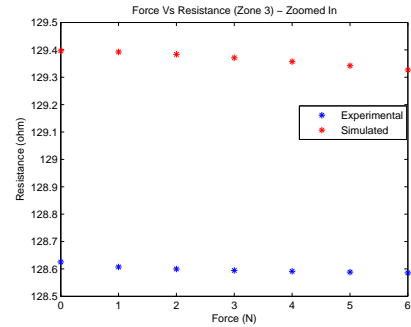
(e) Force Vs Resistance at Zone 2



(f) Force Vs Resistance at Zone 2 - Zoomed In



(g) Force Vs Resistance at Zone 3



(h) Force Vs Resistance at Zone 3 - Zoomed In

Figure 4.8: Plots comparing simulation and experimental results for Resistance ( $\Omega$ ) [2]

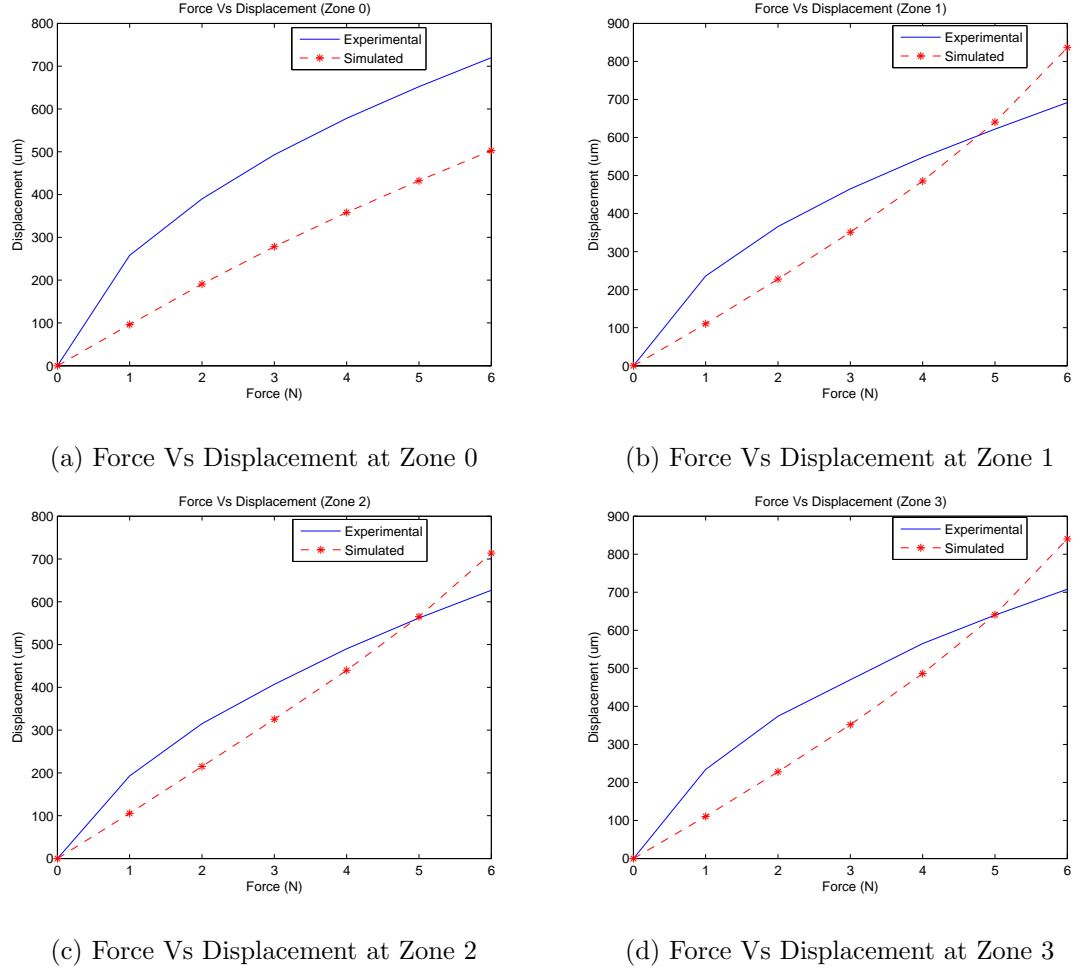


Figure 4.9: Plots comparing simulation and experimental results for Displacement [2]

#### 4.1.5 Discussion

In this chapter we created and validated a FEA model simulating a multi-layer 3D printed strain gauge sensor. The FEA model includes the base substrate which is also the robot skin, Kapton sheets encapsulating the sensor, metal strain gages fabricated lithographically and the piezoresistive 3D-printed PEDOT:PSS layer. We carried out quasi-static loading experiments in four identified zones on and around the strain gage sensor. Similar loading profiles and zones around the sensor were simulated using COMSOL<sup>®</sup> FEA software and results were compared with experimental data. Although the nonlinear, viscoelastic nature of the polymer skin substrate can be improved, the resistance data can

be matched with less than 1% error. Furthermore, there was wide agreement in terms of trends between experiments and simulation. In the future, the model can be employed to study parametric effects of package, strain gage geometry and materials properties, as well as investigate sensor responses to various loading conditions. Models will also be extended to sensor arrays for larger area robot skin. Simulation studies will be used to guide future design choices for these sensor systems.

## 4.2 Tablet Interface

Many assistive robots operate in environments shared with humans. Since the environment around the robot is constantly changing, and the robot needs to adapt to unforeseen circumstances during its operations. A typical assistive robot that operates autonomously must be able to perform perception, cognition, and manipulation tasks related to the desires of humans around the robot. There have been several studies on the development of Human Machine Interfaces (HMI) that assist humans to interact with a robot in such settings. A Home Exploring Robotic Butler or “HERB” was developed to carry out tasks and explore in a household environment [72]. The authors present the algorithm and software architecture implemented on the robot with an aim to accomplish tasks at a faster rate so that it doesn’t frustrate the user. This level of automation requires extensive planning and sensing algorithms. For accomplishing tasks in human environments, robot decision making, and artificial intelligence are essential components that guide the balance between autonomous operation and learning from demonstration. The Cog project, for instance, strived to impart human like intelligence to the robot during physical interaction tasks through teach by demonstration [73]. 2 Other, more recent examples include robots that can perform tasks in environments that are dangerous for human users. The Atlas robot [74] is a humanoid developed during the DARPA Robotics Challenge that requires stable full body control, and switches control between autonomous operation and direct teleoperation by a user. Similarly, NASAs Robonaut project implements anthropomorphic human scale hands and was designed to assist astronauts with manipulation tasks in outer space [75].

While executing commands received from the user, an assistive robot must also ensure the safety of humans present in the workspace to avoid hurting them during robot movements [76]. Several personal robots have been developed to assist users by performing safe physical Human Robot Interaction (pHRI) [77]. Ensuring this safety also allows the robot to be used in different settings, ranging from household to industrial environments. During pHRI, if proper safety precautions and procedures are not implemented on the robot, serious injuries can occur. In a paper published more than 10 years ago, Rachid Alami, et al. [3] enumerated safety guidelines that should be incorporated in a robot to make it safe for pHRI. These safety issues can be visualized in Figure 4.10, which outlines the robotics anthropic domains and the safety concerns in each domain. Rachid Alami, et al. [3] also stated several conditions such as design of lightweight robots, dependability of sensors, safety in software, implementation of emergency stop button or E-Stop buttons, ensuring reliability of control system etc. which makes a robot safe. Recently, a new standard for collaborative robots has encapsulated some of these guidelines [78].

Mobile communication device usage has become prevalent. With increase in processor power and sensors on-board a tablet or a mobile phone allows it to accomplish complex tasks. The use of smartphones to control mobile robots has been recently investigated [39, 40]. Using the on-board sensors, android applications have been developed and deployed to control robots remotely over a network [39–42].

The research study presented aims at building an intuitive interface for an assistive robot. As described by Cremer and Doelling, this assistive robot is planned to be deployed in a healthcare environment with the aim of helping the nurses with their tasks [54]. The functions identified to be performed by the robot are - help walk a patient and provide necessary support during the process, fetch items for the patient, have conversational capabilities, monitoring vital signs of the patient and alert the nurses in case of detected emergencies. The interface for this robot should also consider the safety requirements while the robot is in close proximity to the patient or the nurse.

The implementation of closed-loop controllers for robot teleoperation have been in-

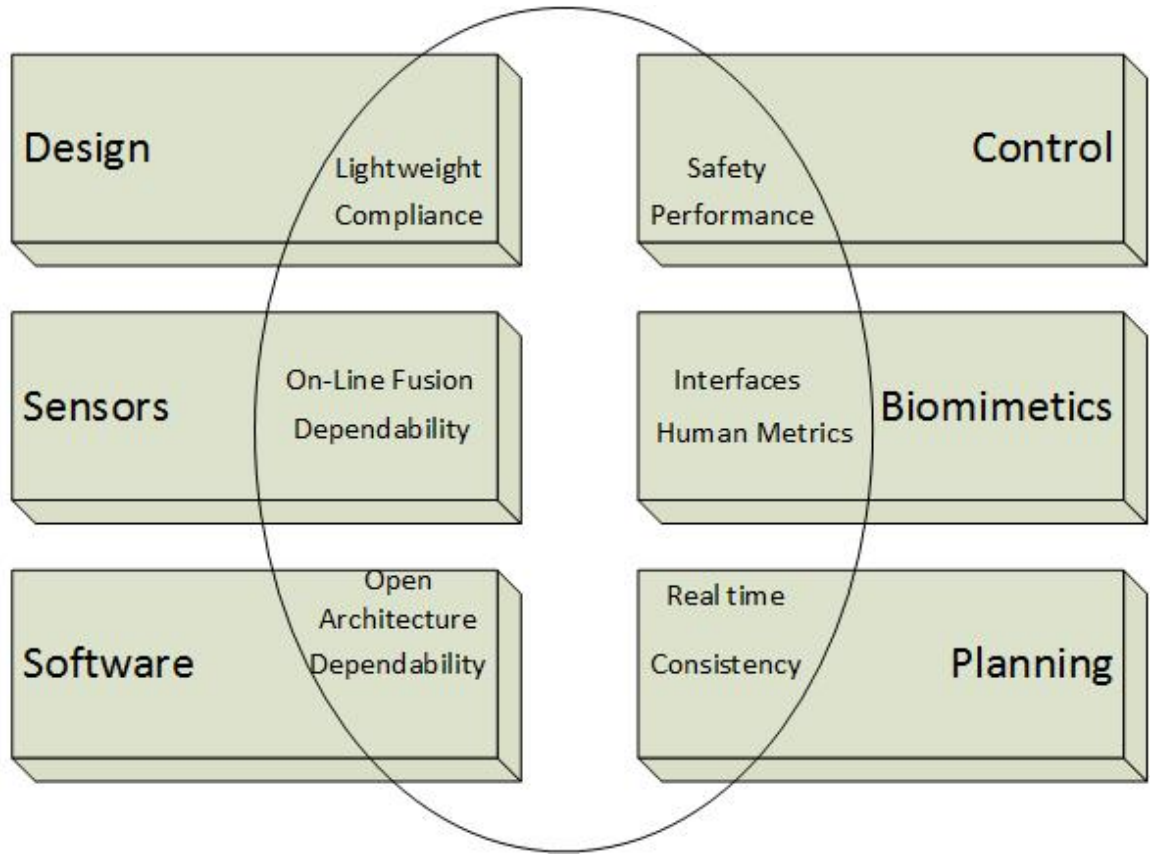


Figure 4.10: Issues related to pHRI. Adapted from [3]

vestigated in several studies [39,40,79–81] . The investigation of teleoperation control suggests that there are performance differences between novice and expert users [80]. Along with the learning curve, teleoperation of a robot presents challenges such as cognitive fatigue, time delays, etc., as discussed in [20]. On the other hand, in a traded control scheme, the user identifies the command to initiate a task, and the robot is expected to plan and perform the task autonomously without any external supervision [20]. This type of control scheme eliminates the need of driving the robot using continuous teleoperation by the user, thus, offering inherent advantages.

In this section, we present a traded control HMI framework through a mobile app that can be used to interact with Robot Sitters or Adaptive Robotic Nursing Assistant (ARNA) robots. Our current work focuses on providing a reliable interface framework

between a patient or a nurse and the sitter robot. The HMI proposed in this paper is implemented as an Android application on a tablet. This interface accepts both voice as well as on screen button press commands and relay high level user commands to the robot. The sitter robot can perform certain routine tasks such as pick-and-place objects at certain locations and pick-and-handover objects to the user. The interface is also capable of socially conversing with the patient. Since this HMI framework is intended to be used in a healthcare facility environment and the user base includes the patients and nurses, the user is allowed to issue emergency stop commands during robot operation. Such commands are given priority and allow the robot to return to a safe position. In certain situations, the patients also not be able to hold or operate the tablet using onscreen buttons. They can still interact with the robot by using the speech interface provided by the tablet application.

During testing of the tablet interface, we have used the Baxter from Rethink Robotics®. We experimentally tested four scenarios based on tasks to be undertaken by a sitter robot. The users were required to initiate tasks such as locate and pick certain items through speech interface or on-screen button interface, converse with the interface and issue emergency stop commands. Through these experiments we recorded the reliability and performance of developed interface. We also test the safety features in these scenarios, and the results are reported here.

#### **4.2.1 System Architecture**

Our system consists of an assistive robot, a Google® Nexus tablet and a base-station computer which acts as a computing platform to share the load of processing algorithms. These components are connected to the same wireless network which enables them to exchange data as depicted in Figure 4.11. A custom developed Android application runs on the tablet and subscribes to Robot Operating System (ROS) topics from the robot and the base-station to connect to the ROS ecosystem. Once the connection is established, the tablet application sends ROS Service Requests based on the input from the user. These ROS Service Requests are then catered to by the ROS Service Server. Google® speech engine



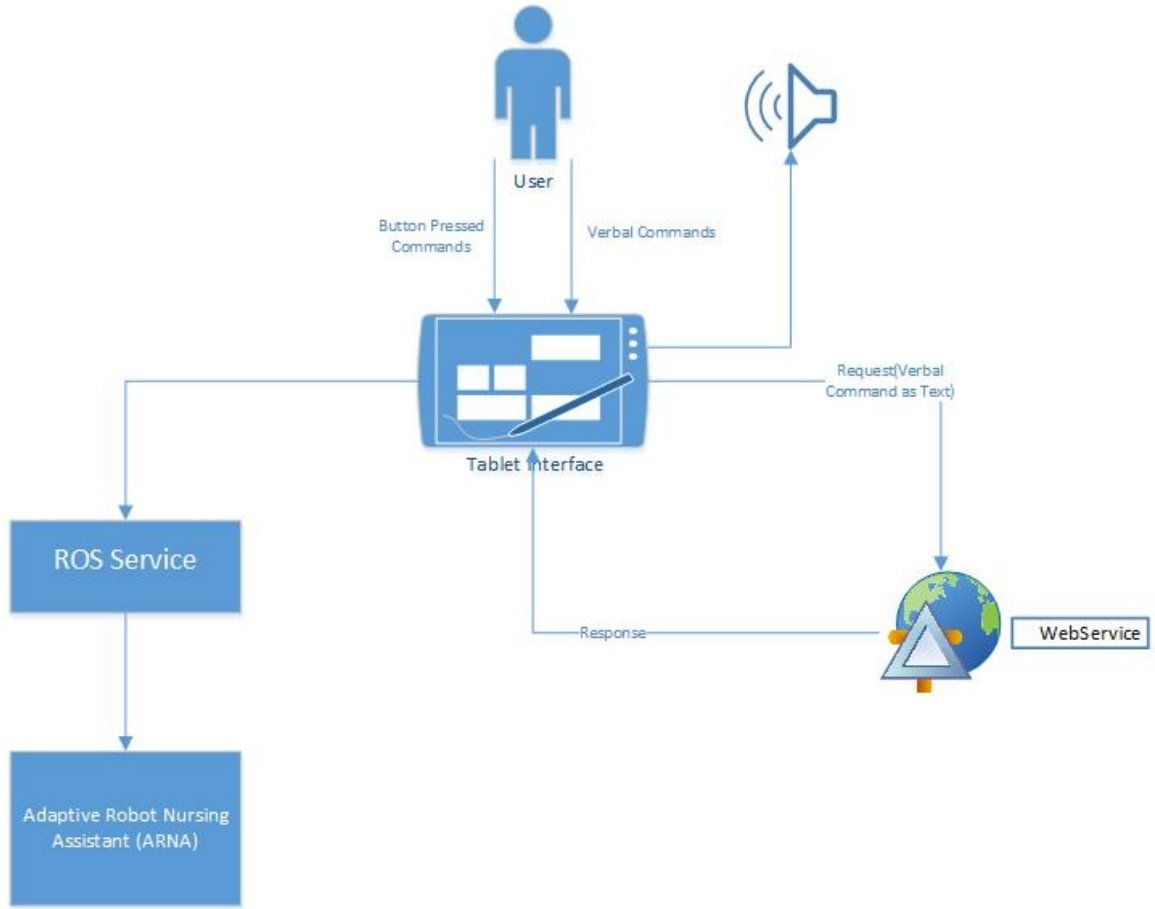


Figure 4.11: System Architecture depicting different hardware and software components [4]

was used to detect and understand the spoken words. The detected speech was then used by an Artificial Intelligence Markup Language (AIML) server to facilitate conversations with the user.

#### 4.2.1.1 Human Machine Interface (HMI) Framework

The HMI framework consists of two logically separated modules based on their functions. One is responsible for the low-level robot control while the other is responsible for high-level planning. A *Cortex module* was developed for the high-level planning which is responsible for trajectory planning, manipulation and visual servoing. Therefore, this framework can be implemented on various robot platforms as the *Cortex modules* is independent

of the low-level robot control. Fig. 4.12 illustrates the architecture of the proposed framework. This type of software architecture enables the HMI to have traded control of the robot.

The current capabilities of the Cortex include:

- Fetch items for the user
- Speech interface for conversational capabilities as well as to issue verbal command.
- Disable robot in case of emergencies

The intended users of this HMI are patients and nurses in a hospital environment, which means that it should be capable of catering to a wide user base who might find it difficult to constantly hold a tablet for long. The network connectivity is continuously checked and upon the issuance of emergency stop command, the state machine halts all operations. These safety features are put in the framework to adhere to the guidelines outlined in [3].

Due to the implementation of the traded control in the framework, it allows the user to issue a high-level command, in the form of a verbal command or a screen gesture, to the robot. Users are not required to constantly monitor and control the robot's activity as the tasks planning and completion is done autonomously. This circumvents the need to teleoperate a robot, where the quality of operation depends the level of expertise with a particular robot which may not be ideal in a healthcare environment with patients as users.

In pick-and-place scenarios, the interface provides the user with a list of choices for the items to be picked. The user can issue a verbal command or select the item by clicking the onscreen button. Once the command is issued, the Cortex uses a RGBD camera to scan and identify the location of the object. When the location of the desired item is determined, the trajectory planning for the arm is initiated. If either the trajectory planning or the visual identification of the object fails, an error message is displayed on the screen. If the trajectory planning is complete, the information is passed on to the low-level controllers to move the robot arm to the required destination and grippers are actuated to pick the item

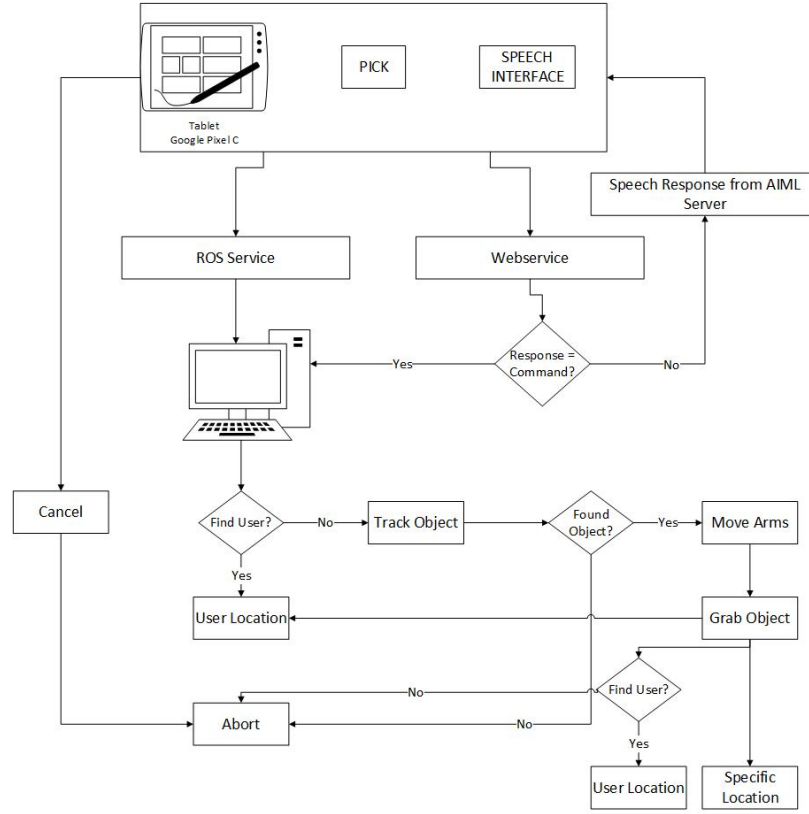


Figure 4.12: Diagram of our proposed Human Machine Interface Framework [4]

up. If this process is completed successfully, the Cortex then plans the trajectory to place the item in a desired location, but if it fails, the entire process is restarted. Along with pick-and place, pick-and-handover functionality can also be implemented by identifying the patient and bringing the item close to the user for handover.

Fig. 4.12 illustrates the workflow that is responsible for maintain the safe operation of the robot. The robot resets to its rest position in the following events:

- While the robot is completing a task a cancel task command is initiated.
- The user is not found during pick-and-handover task.
- Unable to plan trajectory after multiple tries.

Whereas, when the emergency stop command is initiated, the robot completely stops moving. These features are vital to maintain safety of the user.

#### 4.2.1.2 Android Application for Tablet and ROS interface

The robot platform uses Robot Operating System (ROS) for its operation and the tablet runs Android OS. The application for the tablet was developed using Android studio rosjava [82]. Rosjava provides the libraries for an android application to communicate with a ROS application. Using the Android Studio 2.3, the following classes were implemented for the android application:

- MainActivity: It provides the graphical user interface (GUI). The GUI provides both button based and verbal based interfaces to initiate user commands.
- NodePublisher: It is responsible to establish communication with ROS. A ROS service was created which was used by the class to execute rosservice commands.
- The CallWebservice: It connects the GUI to the Simple Object Access Protocol (SOAP) web service. The speech command is sent to the webservice to determine whether the speech was a conversational in nature or a pick-and-place command.

A Tomcat Apache server 9.0 was used to deploy the SOAP service locally. The web services are used to handle all the speech commands. It uses a chat engine named Artificial Linguistic Internet Computer Entity (ALICE) to respond to conversational queries appropriately. A console output of the chat engine is shown in Figure 4.13.

#### 4.2.1.3 Robot Hardware Description

The Baxter<sup>®</sup> robot is a new type of collaborative robot with two 7-DOF arms [83]. The maximum reachability of each arm is 1.21 m. This provides the ability to pick and place object within a slightly longer reach than a human. The Baxter robot has an Ethernet port through which it is connected to the same network as the tablet. A Kinect camera has been attached to the head of the robot to provide a constant video feedback to the tablet, as shown in Figure 4.14. In our tested scenarios, the user holding the tablet is the one items should be fetched for. The Baxter runs Robot Operating System (ROS) and can communicate with external application through ROS topics.

```

Human: Hello
STATE=Hello:THAT=unknown:TOPIC=unknown
normalized = Hello
Matched: HELLO <THAT> * <TOPIC> * salutations.aiml
writeCertainIFCaegories learnf.aiml size= 0
Robot: Hi there!
Human: What is cancer?
STATE=What is cancer?:THAT=Hi there:TOPIC=unknown
normalized = What is cancer?
Matched: WHAT IS CANCER <THAT> * <TOPIC> * knowledge.aiml
writeCertainIFCaegories learnf.aiml size= 0
Robot: Any disease resulting from malignant cell growth.
Human: Do you know anything about medicine?
STATE=Do you know anything about medicine?:THAT=Any disease resulting from
malignant cell growth:TOPIC=CANCER
normalized = Do you know anything about medicine?
Matched: DO YOU KNOW ANYTHING ABOUT * <THAT> * <TOPIC> * reduction3.safe.aiml
0. <srai>what is medicine</srai> from DO YOU KNOW ANYTHING ABOUT * <THAT> *
<TOPIC> * topic=CANCER)
Matched: WHAT IS MEDICINE <THAT> * <TOPIC> * knowledge.aiml
writeCertainIFCaegories learnf.aiml size= 0
Robot: The science and art of diagnosing, trating, curing, and preventing disease,
relieving pain, and improving and preserving health. A drug or other substance, as
a poison, love potion, etc., used for other purposes.

```

Figure 4.13: Speech Interface output during conversation [4]

The base-station is an external computer which has Baxter's SDK installed and acts as an auxiliary processing unit. ROS nodes initiated in this base-station communicate with Baxter through the common network. The ROS Service Servers are hosted on this base-station which accepts ROS Service requests from ROS Service Client nodes. After receiving these ROS Service requests, the appropriate service routine is invoked. All the modules developed for the Baxter robot were stored and executed from the external base-station. The base-station treats the ROS system of the Baxter as its `ROS_MASTER_URI`. The tablet application also treats Baxter as the master system. Once the connections are established, the base-station as well as the tablet application are able to access and manipulate Baxter.

Moveit! motion planning framework was used to control and plan trajectories for Baxter's arm [84]. During the pick-and-place task, an item is identified from the list provided by the tablet interface. The location of the required item is then determined using visual feedback. This identified location is then passed on to the Moveit! application which plans the arm trajectory as well as the joints angles required to move the arms towards the object.



Figure 4.14: External camera sensors installed on the Baxter for user identification [4]

QR codes were assigned to each object in the list. This allowed the identification of the items through these unique QR codes that were assigned to each object [85]. The onboard camera placed under the wrist of the Baxter's arm were used to scan the workspace to identify these codes, hence identifying the desired object and its location.

For the detection of user, a QR code was attached to the tablet held by the user. A Microsoft Kinect sensor was attached to the head of the Baxter to help identify the location of the tablet and the user. This assist in the pick-and-handover scenarios.

#### 4.2.2 Application Interface

The application interface is divided into three logical units: the button interface unit, the speech interface unit and the video feedback interface unit. These units are explored in detail in this section.

There are eight software buttons which are programmed to send specific ROS service calls to initiate desired routines. In the Android studio, the buttons are designed to invoke button pressed routines. Each button pressed routine initiates a ROS service call with an

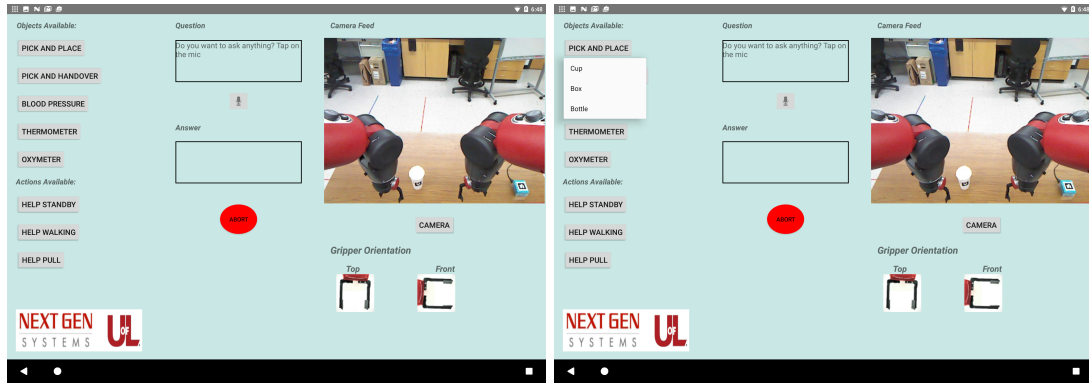


Figure 4.15: Tablet application screen [4]

associated command. For example, if pick item Cup is chosen, a message Pick Cup is sent through the ROS topic. The ROS service server, which is listening to the ROS topic, receives the message and depending on the message, an appropriate routine is initiated. If the cancel request button is pressed, all tasks being performed are halted at once. These buttons are used for predefined routine medical activity such as pick an item, check blood pressure, measure patient's temperature. The buttons provide easy access to routine commands that assist the patient as well as nurses to perform their tasks. The verbal command section accepts spoken voice as commands when the mic button is pressed, as seen in Figure 4.15.

An Artificial Intelligence Markup Language (AIML) chatbot server is also running on the base-station. The voice commands spoken to the tablet is converted to text using speech to text engine and the text is passed to the chatbot server. The chatbot server extrapolates a response based on the text received and sends it to the tablet. This response is converted to speech and is spoken through the tablet. During the course of operation, all the processed speeches are displayed on screen as text. This type of interface allows the patients to have a conversation as well as command the robot to perform desired routines.

Finally, the live video window shows direct video feed from the robot to the tablet. This gives the user the perspective of the robots surrounding, and helps with selection of grasp patterns, and with visual feedback of successful object manipulation. A Kinect is placed on the robot to acquire images from the robots surrounding. This video stream is





Figure 4.16: Markers attached to objects [4]

then published on a topic. The tablet application listens to the topic and as soon as images are received, they are shown at the image viewer section of the tablet.

#### 4.2.3 Experiments and Discussion

Expert and non-expert users were selected to perform the preliminary testing of the proposed interface. Expert users had prior knowledge of the robot system whereas the non-expert users were unaware of the functionality of the robot system. The users were briefed in detail the tasks that were to be performed and the safety features to be used in case of emergencies. Figure 4.16 illustrates the three objects, a cup, a box and a bottle, placed in front of the robot for the pick and place tasks.

The experiment was conducted through four process:

- Initiate pick-and-place with button-based commands.
- Initiate pick-and-place with verbal command.
- Initiate abort command using button based and verbal based commands.



- Have a conversation through the interface.
- Completing a questionnaire rating the system.

The metrics for evaluation of the proposed HMI framework were:

- Ease of Use, as measured by a questionnaire.
- Time to complete pick and place task.
- Conversation through speech interface rated by users.
- Success rate in pick and place task.
- Command abort process success rate.

Two expert and two non-expert users were selected to conduct the experiments. Pick-and-place tasks were initiated six times each with verbal and button based commands by each user. The item to pick was randomly chosen by the users. Thus, a total of sixty experiments were conducted during which the time of completion was recorded for each run. Figure 4.17 represents the recorded data. U1 and U2 are the trained users whereas U3 and U4 are untrained users. The label BC represents the tasks initiated with button commands and SC represents the speech command based initiation of tasks. The average time of completion for trained user was  $54.6 \pm 5.05$  seconds and  $55.54 \pm 6.91$  seconds for the untrained users. These results show that the performance of the proposed HMI framework is comparable with trained and untrained users.

Users were then directed to have a conversation through the tablet and rate their experience on a questionnaire. Each question had options to answer based on a Likert scale with five-level Likert items.

- Did the speech interface recognize what you said correctly?
- Was the response valid to a question asked by you?
- Did you have to raise your voice pitch in order to interact with the speech interface?

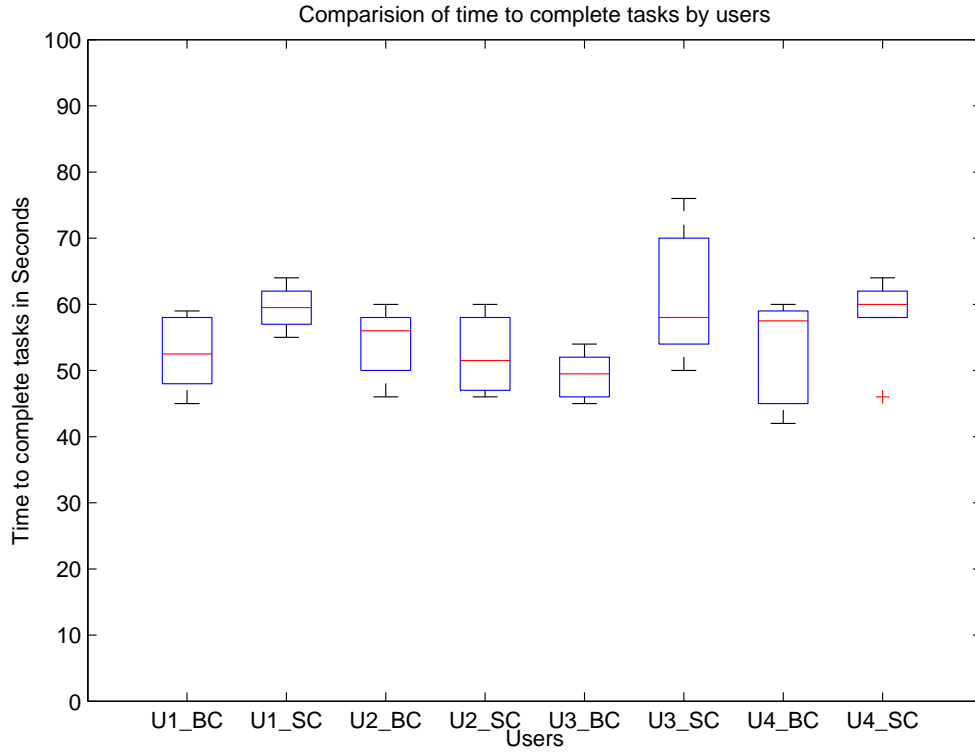


Figure 4.17: Time of completion comparison between users [4]

- Was the audio response from the interface clear?
- Was the application interface easy to use?
- Was the speech interface helpful to you?

The performance of the speech engine was observed to vary with the accent of the users. Table 4.4 represents the summery of the ratings from the users and a lower score for speech interface reflects the speech engine’s performance with accents. It was also observed that in cases where the motion planning was not successful during pick-and-place tasks, the robot moves to a known position to restart the planning process again.

TABLE 4.4

Average rating of the interface by the users

Users	Ease of Interface	Speech Interface
Trained User 1	4/5	4.5/5
Trained User 2	4.5/5	3.8/5
Untrained User 1	4.5/5	4.0/5
Untrained User 2	4.6/5	4.2/5

In this section, we presented a human-machine interface framework applied to robot sitters in a hospital environment. The framework includes a tablet interface, a 2-arm robot that can manipulate objects, and a user that can request items and engage in conversation with the robot. The framework implements a traded control scheme, in which users are responsible for certain high-level tasks, such as initiating commands, selecting objects, or terminating tasks, while the robots autonomously complete medium-level tasks such as trajectory planning and execution, object localization, and grasp planning.

Preliminary experimental results suggest that task and rate completion of tasks is similar for skilled and unskilled users. In future developments, a scaled version of the tablet app will be interfaced with a Nursing assistant robot to operate in a hospital scenario and can assist patients. Future studies will also include collecting sufficient statistics from user experiments for performance evaluation of our interface.

## CHAPTER 5

### NEURO-ADAPTIVE CONTROLLER (NAC) AND HUMAN INTENT ESTIMATION MODULE (HIE)

#### 5.1 Introduction

Collaborative robots or co-robots are tasked with working alongside human users, helping them achieve a common goal through physical human-robot interaction (pHRI). These tasks require a robot and a human user to work with the same object such as during welding [26], assisting with medical tasks [86] and providing help with industrial assembly [87]. In all these cases, the work-piece is held by both the user and the robot while moving it towards a desired trajectory. Another such application of co-robots can be seen in the field of rehabilitation robotics, where the robots guide and help patients in accomplishing their goals while providing support whenever needed [88]. Since, during these interactions the human user is in close proximity if the robot, the question of user safety has to be addressed as well. Studies have shown that in order to make pHRI safer, the robots need to be lightweight and compliant [89] [90].

To achieve safer pHRI, force-based control schemes such as impedance and admittance controllers are used [91]. These controllers account for the external forces and torques applied on the robot and determine a compliant trajectory for the robot. Model-based controllers are also used to impart safe pHRI features to a robot through modification of model parameters [92] [93] [94] [95]. One drawback of using model-based controller is that it is system dependent and for any change in robot systems, new parameter values have to be determined.

During pHRI, estimation of user intent is also important as it would allow the robot to predict and follow the intended trajectory. Wang et. al. and Futamura et. al. have

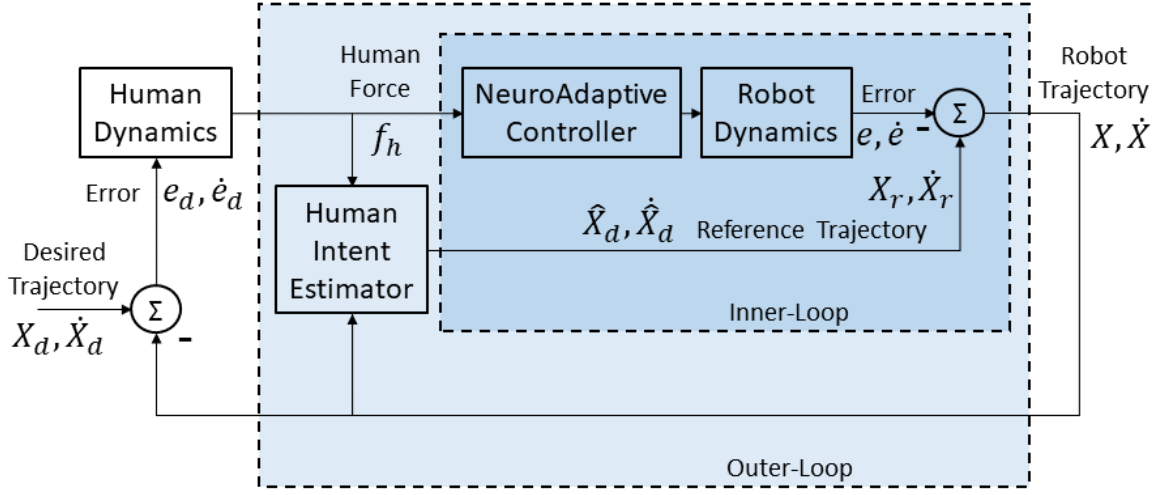


Figure 5.1: NAC with HIE and human-in-loop operation [5]

conducted studies to explore a human intent estimation (HIE) module. Studies suggest that the addition of the HIE module assists pHRI and helps with the performance of the robot. Suzuki et. al., Ragazzini et. al. and Phatak et. al. have conducted studies exploring the model of a human brain [94] [96] [97]. The studies utilized a PD controller to model the human brain dynamics which can be used to integrate a user model in the control system. This would allow the robot to continuously learn the human dynamics during pHRI. The user also reacts to the robot behavior by learning and modifying their response to maintain the desired trajectory.

The neuroadaptive controller (NAC) presented here is based on research studies by Lewis et. al. and Ranatunga et. al. [44–46, 98–101]. NAC utilizes a neural network to estimate the robot system dynamics while calculating appropriate torque for the robot. The control system presented here, uses a variation of the NAC which contains two loops, namely an outer loop and an inner loop. The inner loop is responsible for estimating the robot system dynamics whereas the outer loop is responsible to estimating the human dynamics. The controller learns these dynamics online without any prior training. In the past, a double integrator based module have been used for the purpose of HIE which reacted the user forces and accelerations. The proposed controller proactively modifies its trajectory

by sensing user forces. The HIE module senses these user forces to adapt online and help maintain a desired path.

During pHRI, the quality of interaction is also vital as it provides insight into the controller performance. Studies in the past have shown that jerk is an important metric through which the quality of pHRI can be determined [102]. In this paper, jerk has been added as a metric to determine the performance of the proposed controller.

Figure 5.1 illustrates the proposed controller.  $\hat{x}_d$  is an estimation of the desired trajectory  $x_d$  by the HIE module which uses human force  $f_h$  and the current position  $x$ . The estimated desired trajectory is then used as the reference trajectory  $x_r$  by the inner loop. The NAC in the inner loop tries to follow this reference trajectory.

An additional component called prescribed error dynamics (PED) is also added to the proposed controller. This allows the robot to behave as a second-order linear system which the user perceives.

The study presented describes the formulation of a novel model-free controller to assist during pHRI operations. The controller utilizes neural networks to estimate robot and human dynamics online during co-manipulation tasks. The study also includes the validation of the proposed controller through co-manipulation experiments with users. The performance of the proposed controller was compared with traditional controllers by using metrics such as jerk, human force and trajectory error.

## 5.2 Outer Loop

The outer loop consists of the HIE module as well as the human user. This section describes the formulation of the modules to predict the desired trajectory  $x_d$ .

### 5.2.1 Human Transfer Function

The human transfer function has been described as a PD controller [94], which can be represented as:

$$H(s) = \frac{D_h s + K_h}{Ts + 1} e^{-Ls} \quad (5.1)$$

where  $K_h, D_h \in \mathbb{R}^{6 \times 6}$  are the parameters for gain of the controller,  $T$  is the time constant and  $L$  is a delay factor.

Let  $x_d$  and  $\dot{x}_d$  be the desired pose and velocity respectively, then  $H(s)$  can be rewritten by ignoring the time constant and delay [94]:

$$D_h \dot{e}_d + K_h e_d = f_h \quad (5.2)$$

where

$$e_d = x_d - x$$

$$\dot{e}_d = \dot{x}_d - \dot{x}$$

representing errors in position and velocity. This shows that a human user applies forces during co-manipulation tasks in order to reduce the error in the trajectory. As soon as the error is eliminated  $e, \dot{e} \rightarrow 0$ , the human force  $f_h$  also ceases to exist.

The proposed controller was designed keeping in mind the following two key principles:

*User Effort Reduction:* This principle dictates that the controller takes over the majority of effort required for task completion with reduced user effort  $f_h$ . By this design, the controller would be able to put in maximum effort while user forces would be utilized for guiding purposes. This minimization of user forces allows to controller to be safer. It can be noted that this design will not be applicable in the rehabilitation tasks where the goal is not to reduce human effort but to provide support to the user.

*Crossover model:* During pHRI, if the user experiences a linear admittance model, the interaction is more favorable as described in past research [94] [103]. This means that all the nonlinearities of a complex robot should not be observable to the user.

### 5.2.2 Human Intent Estimator

The HIE component is responsible for estimating the user intended trajectory by taking into account the human dynamics represented in equation 5.2.

In the previous section, it was established that the human forces exist as long as there is an error between the desired and current trajectory. Therefore, the intended or desired trajectory can be estimated from the user forces applied and the current trajectory of the robot by modeling the human dynamics. Since the human dynamics function can be nonlinear, a neural network can be used to emulate it as:

$$\begin{aligned} \begin{bmatrix} x_d \\ \dot{x}_d \end{bmatrix} &= h(x, \dot{x}, f_h) \\ &= V^\top \phi(\xi) + \varepsilon_1 \end{aligned} \quad (5.3)$$

where  $V^\top \in \mathbb{R}^{12 \times 18}$  is the weight matrix,  $\phi(\cdot)$  is the activation function,  $\xi = [f_h^\top x^\top \dot{x}^\top]^\top \in \mathbb{R}^{18}$  constitute the input vector, and  $\varepsilon_1 \in \mathbb{R}^{12}$  represent a small residual error. The human intent can be estimated as

$$\begin{bmatrix} \hat{x}_d \\ \dot{\hat{x}}_d \end{bmatrix} = \hat{V}^\top \phi(\xi) \quad (5.4)$$

where  $\hat{V}^\top$  is the estimated weight matrix. Therefore, the error estimates are:

$$\begin{bmatrix} \hat{e}_d \\ \dot{\hat{e}}_d \end{bmatrix} = \begin{bmatrix} \hat{x}_d - x \\ \dot{\hat{x}}_d - \dot{x} \end{bmatrix} \quad (5.5)$$

Then the deviation of the estimated error from the actual error can be defined as

$$\begin{aligned} \tilde{\tilde{e}}_d = \begin{bmatrix} \tilde{\tilde{e}}_d \\ \dot{\tilde{\tilde{e}}}_d \end{bmatrix} &= \begin{bmatrix} e_d \\ \dot{e}_d \end{bmatrix} - \begin{bmatrix} \hat{e}_d \\ \dot{\hat{e}}_d \end{bmatrix} \\ &= \begin{bmatrix} x_d \\ \dot{x}_d \end{bmatrix} - \begin{bmatrix} \hat{x}_d \\ \dot{\hat{x}}_d \end{bmatrix} = \tilde{V}^\top \phi(\xi) + \varepsilon_1 \end{aligned} \quad (5.6)$$

where  $\tilde{V} = V - \hat{V}$  is the weight estimation error.



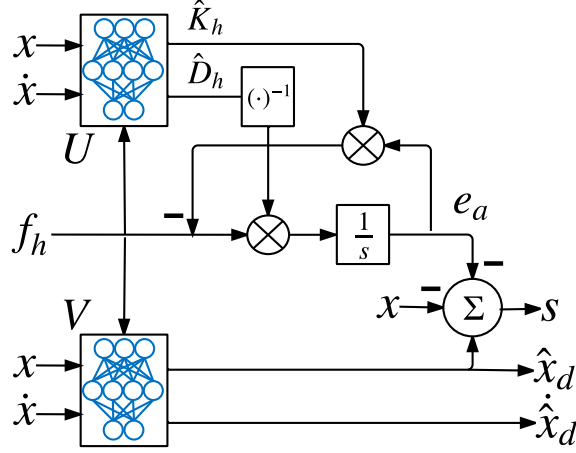


Figure 5.2: Human intent estimator [5]

A sliding mode error is defined as:

$$s = \hat{e}_d - e_a \quad (5.7)$$

where  $e_a$  is the filtered error. This filtered error is from the estimated human dynamics

$$f_h = \hat{D}_h \dot{e}_a + \hat{K}_h e_a \quad (5.8)$$

$$= J(\hat{P} \odot \bar{e}_a) \quad (5.9)$$

where  $\odot$  is the element-wise Hadamard product and  $J = [I_6 \ I_6]$ , where  $I_6 \in \mathbb{R}^{6 \times 6}$  is the identity matrix. Vector  $P \in \mathbb{R}^{12}$  consists of diagonal elements of  $K_h, D_h$  and  $\bar{e}_a \in \mathbb{R}^{12}$  are defined as

$$P = \begin{bmatrix} \vec{k} \\ \vec{d} \end{bmatrix} \quad (5.10)$$

$$\bar{e}_a = \begin{bmatrix} e_a \\ \dot{e}_a \end{bmatrix} \quad (5.11)$$

Fig. 5.2 represents the HIE module containing two neural networks (NN) which predicts the desired trajectory  $\hat{x}_d$ .

**Theorem 5.2.1.** *When  $s \rightarrow 0$ , then the desired trajectory approaches the actual trajectory,  $\hat{x}_d \rightarrow x_d$ , as long as the user provides guidance such that  $e_d \rightarrow 0$ .*

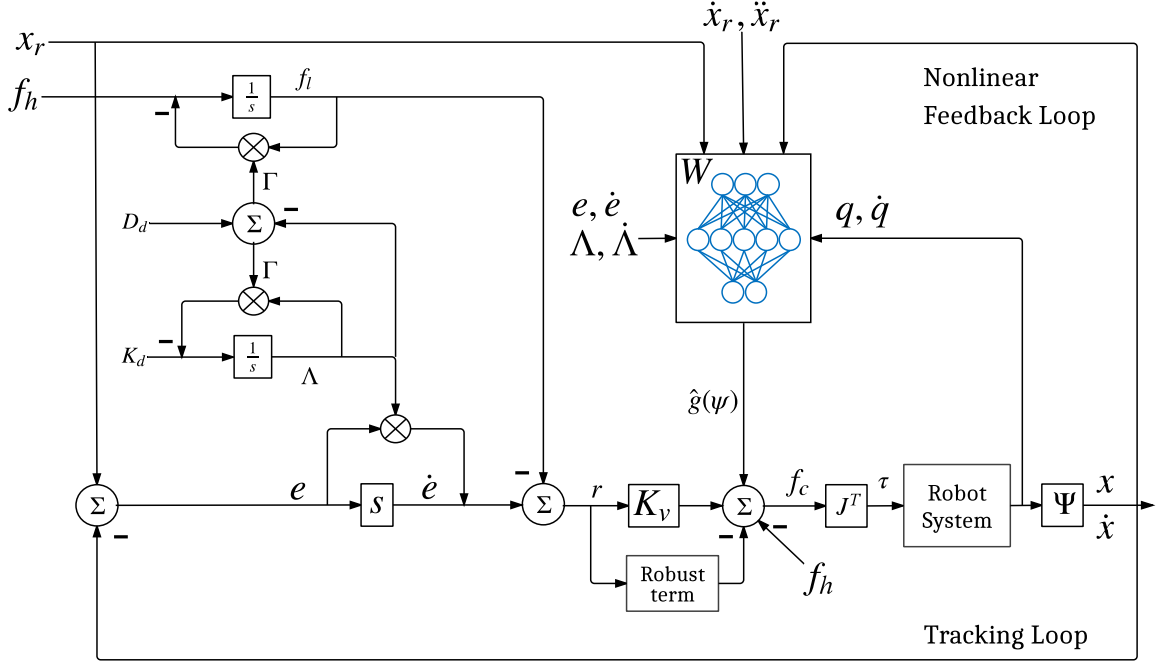


Figure 5.3: Inner Loop [5]

*Proof: From (5.7),*

$$\begin{aligned}
 s &= (\hat{x}_d - x) - e_a \\
 &= (\hat{x}_d - x_d + x_d - x) - e_a \\
 &= -\tilde{e}_d + e_d - e_a
 \end{aligned} \tag{5.12}$$

From (5.2), if  $e_d \rightarrow 0$ , then  $f_h \rightarrow 0$ . And from (5.8), if  $f_h \rightarrow 0$  then  $e_a \rightarrow 0$ . Therefore, if  $s \rightarrow 0$ , then  $\tilde{e}_d \rightarrow 0$ , which also indicates that  $\hat{x}_d \rightarrow x_d$ .

### 5.3 Inner Loop

Fig. 5.3 illustrates the inner loop of the controller. The inner loop is responsible for estimating the robot dynamics and ensuring that the robot follows the given trajectory. The PED component in the inner loop ensures that the robot behaves like a second order system during pHRI.

The robot dynamics equation is given by

$$M(q)\ddot{x} + V(q, \dot{q})\dot{x} + F(\dot{q}) + G(q) = f_c + f_h \tag{5.13}$$

where,  $q, \dot{q} \in \mathbb{R}^n$  represent the joint position and velocity and  $x \in \mathbb{R}^6$  represents the end-effector pose in Cartesian space. A forward kinematics function  $\Psi$ , as shown in Fig. 5.3, is used to map joint angles  $q$  to Cartesian coordinate of the end-effector  $x$ .  $M(q) \in \mathbb{R}^{6 \times 6}$  represents the inertia matrix,  $V(q, \dot{q}) \in \mathbb{R}^{6 \times 6}$  is the Coriolis matrix,  $G(q) \in \mathbb{R}^6$  is the gravity matrix, and  $F(\dot{q}) \in \mathbb{R}^6$  represents the friction.

The control force  $f_c \in \mathbb{R}^6$  to be applied during pHRI is calculated from the neuroadaptive controller. The NAC provides joint torque  $\tau$  to control the robot and in the case of providing a desired  $f_c$ , the Jacobian is used as  $\tau = J^\top f_c$ .

*Assumption 1:* It is assumed that during the entire pHRI process, the user is in contact with the end-effector, thus sharing the same Cartesian workspace. Also, the measured force is equal and opposite to the applied force on the end-effector.

### 5.3.1 Prescribed Robot Error Dynamics

This section explains the formulation of PED to make the robot behave like a second order system from user's perspective. For such a system, let the error dynamics be

$$\ddot{e} + D_d \dot{e} + K_d e = f_h \quad (5.14)$$

where  $D_d \in \mathbb{R}^{6 \times 6}$ ,  $K_d \in \mathbb{R}^{6 \times 6}$  are the coefficients of the admittance model.

The tracking error is defined as

$$e = x_r - x \quad (5.15)$$

and a sliding mode error is defined as

$$r = \dot{e} + \Lambda e - \dot{f}_l \quad (5.16)$$

$$\dot{r} = \ddot{e} + \Lambda \dot{e} + \dot{\Lambda} e - \ddot{f}_l \quad (5.17)$$

where  $\Lambda(t) \in \mathbb{R}^{6 \times 6}$  is a matrix and is also a function of time.  $f_l$ , the filtered force can be written as

$$f_h = \dot{f}_l + \Gamma f_l \quad (5.18)$$

where  $\Gamma(t) \in \mathbb{R}^{6 \times 6}$  is also a matrix which is a function of time. The following theorem represents the method described by Li et. al. [95].

**Theorem 5.3.1.** *By defining  $D_d$  and  $K_d$  as:*

$$D_d = \Lambda + \Gamma \quad (5.19)$$

$$K_d = \dot{\Lambda} + \Gamma\Lambda \quad (5.20)$$

*If  $r \rightarrow 0$ , then the system follows (5.14).*

*Proof: Let  $w$  be an error signal defined as:*

$$w = \ddot{e} + D_d \dot{e} + K_d e - f_h \quad (5.21)$$

*From (5.19), (5.20) and (5.18)*

$$\begin{aligned} w &= \ddot{e} + (\Lambda + \Gamma)\dot{e} + (\dot{\Lambda} + \Gamma\Lambda)e - (\dot{f}_l + \Gamma f_l) \\ &= \ddot{e} + \Lambda\dot{e} + \dot{\Lambda}e - \dot{f}_l + \Gamma(\dot{e} + \Lambda e - f_l) \\ &= \dot{r} + \Gamma r \end{aligned} \quad (5.22)$$

*Therefore, if  $r \rightarrow 0$  then  $w \rightarrow 0$ . Thus, the system follows (5.14).*

### 5.3.2 Inner-loop Error Dynamics

The inner loop dynamics can now incorporate both PED and the robot dynamics.

From (5.16)

$$\dot{e} = r - \Lambda e + f_l \quad (5.23)$$

differentiating (5.15)

$$x = x_r - e \quad (5.24)$$

$$\dot{x} = \dot{x}_r - r + \Lambda e - f_l \quad (5.25)$$

$$\ddot{x} = \ddot{x}_r - \dot{r} + \Lambda\dot{e} + \dot{\Lambda}e - \dot{f}_l \quad (5.26)$$

Using (5.13) and the  $x, \dot{x}, \ddot{x}$  the error dynamics can be written as:

$$M(q)\dot{r} = -V(q, \dot{q})r + g(\psi) - f_c - f_h \quad (5.27)$$

where,  $g(\psi)$  represents

$$\begin{aligned} g(\psi) = & M(q)(\ddot{x}_r + \Lambda\dot{e} + \dot{\Lambda}e - \dot{f}_l) \\ & + V(q, \dot{q})(\dot{x}_r + \Lambda e - f_l) \\ & + F(\dot{q}) + G(q) \end{aligned} \quad (5.28)$$

and  $\psi = [f_l \ \dot{f}_l \ \text{diag}\{\Lambda\} \ \text{diag}\{\dot{\Lambda}\} \ q^\top \ \dot{q}^\top \ e^\top \ \dot{e}^\top \ \dot{x}_r^\top \ \ddot{x}_r^\top]^\top$ .

Let the control input be

$$f_c = \hat{g}(\psi) + K_v r - f_h \quad (5.29)$$

where  $K_v = K_v^\top > 0 \in \mathbb{R}^{6 \times 6}$ . Using a neural network to estimate  $g(\psi)$

$$\hat{g}(\psi) = \hat{W}^\top \sigma(\psi) + \varepsilon \quad (5.30)$$

can be achieved as detailed in [99]. Substituting (5.29) into (6.32)

$$\begin{aligned} M(q)\dot{r} = & -V(q, \dot{q})r + g(\psi) - f_h - \hat{g}(\psi) - K_v r + f_h \\ = & -(V(q, \dot{q}) + K_v)r + \tilde{g}(\psi) \end{aligned} \quad (5.31)$$

where  $\tilde{g}(\psi) = g(\psi) - \hat{g}(\psi)$  is the approximation error. Finally, the inner-loop error dynamics are

$$M(q)\dot{r} = -(V(q, \dot{q}) + K_v)r + \tilde{W}^\top \sigma(\psi) + \varepsilon \quad (5.32)$$

## 5.4 Experimental Validation

As seen in Fig. 5.4, a Personal Robot 2 (PR2) platform was used to deploy the and test the proposed controller. The PR2 consists of an omni-directional mobile base with two 7-DOF arms. The end-effector on each arm has a Force-Torque sensor attached to sense user interaction forces. Robot Operating System (ROS) was used as the development environment for the controller which was then deployed on the PR2.

For the experiment setup, a table was placed in front of the PR2 robot. A paper containing the trajectories, as illustrated in Fig. 5.5, was pasted on the table. The PR2



Figure 5.4: Experimental Setup [5]

robot holds on to a cylindrical pipe which is also held by the user during the experiments. The user forces applied on the cylindrical pipe is sensed by the Force-Torque sensor.

There are three components of the trajectory - a square, a circle and a diamond. These trajectories were chosen to explore the performance of the controller during only x or y direction, non-linear x-y motion and mixed x-y motion respectively. The experiment also included the traversal of the trajectory in both clockwise and anti-clockwise direction. In order to follow these paths, the following sequence of path points were followed:

$$A, C, I, G, A, G, I, C, A, D, B, F, H, D, H, F, B, D, \\ B', F', H', D', H', F', B', D', A$$

where ' ' indicates a circular path.

There were three types of controller settings were identified for the experiment process and each setting had two sub tests with PED turned on or off. The experiment test-cases

are listed as below:

- Prescribed Error Dynamics
  - (a) Disabled
  - (b) Enabled
- Reference trajectory  $x_r$  set to
  - (i)  $x_d$  (pose desired by human)
  - (ii)  $\hat{x}_d$  (estimated human intent from outer-loop)
  - (iii)  $x_m$  (fixed admittance model from [45])

The total number of experiment test cases was six for each subject. In the first controller setting the reference trajectory of the inner loop was set to the trajectory shown in Fig. 5.5. The second setting consists of using the output of HIE as the reference trajectory of the inner loop and in the third setting an admittance model was used. The admittance model is based on the prior work of Ranatunga et. al. [citeranatunga2017adaptive](#) which also contains an intent prediction module. The controller has an autoregressive moving-average (ARMA) model and it was used for comparing with HIE's performance.

Four subjects (2 male and 2 female), in the 25 to 30 age group, were chosen to perform these experiments. Each experiment setting was repeated five times over the three trajectories. Therefore, each subject performed 90 experiments which totals to 360 experiments for the group. These experimental results were analyzed, and Fig. 5.7 illustrates the analysis of the data.

There were three metrics defined to measure the performance of the controllers during the experiments. These metrics were - squared jerk, mean force and mean position error. The formula to calculate mean position error is:

$$\overline{\|e\|_2} = \frac{1}{N} \sum_{k=1}^N \|x_{\text{des}}(kT) - x(kT)\|_2 \quad (5.33)$$

where  $N$  is the number of data points and  $T = 0.02$  seconds.

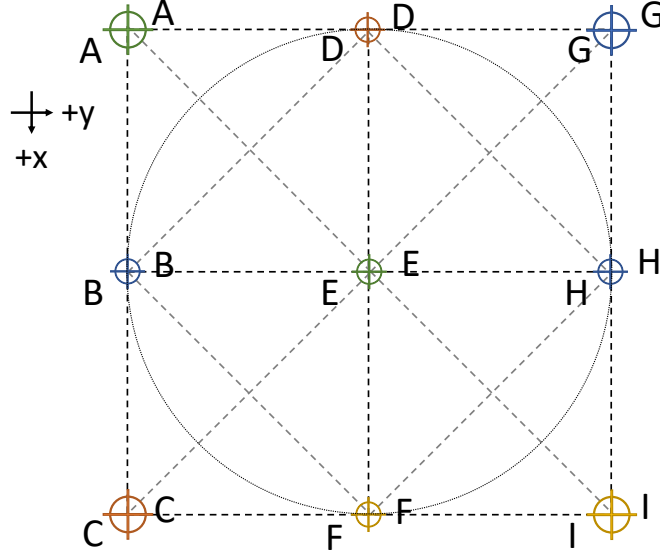


Figure 5.5: Grid layout for trajectory following. [5]

The equation to calculate squared jerk is:

$$J_{\alpha} = \int_{t_s}^{t_f} \ddot{p}(t)^2 dt \frac{(t_f - t_s)^5}{A^2} \quad (5.34)$$

where  $A$  is the path length  $p(t)$ ,  $t_s$  is the start time and  $t_f$  is the final time. The formula for mean force is:

$$\overline{\|f\|_2} = \frac{1}{N} \sum_{k=1}^N \|f(kT)\|_2 \quad (5.35)$$

ARMA model requires prior training where as the HIE module of the proposed controller learns the dynamics online. Fig. 5.7 shows that the performance of the proposed controller is comparable to its counterparts. It can also be observed that with the PED enabled, the performance of all the test cases are improved.

## 5.5 Discussion

A NAC based controller with HIE module and PED component was proposed in this research work. The controller was validated by implementing it on a PR2 robot platform and conducting co-manipulation studies with four users. The controller's performance was compared with two other controller settings along with turning the PED on and off for all three settings. The HIE's performance was comparable to the other settings but the



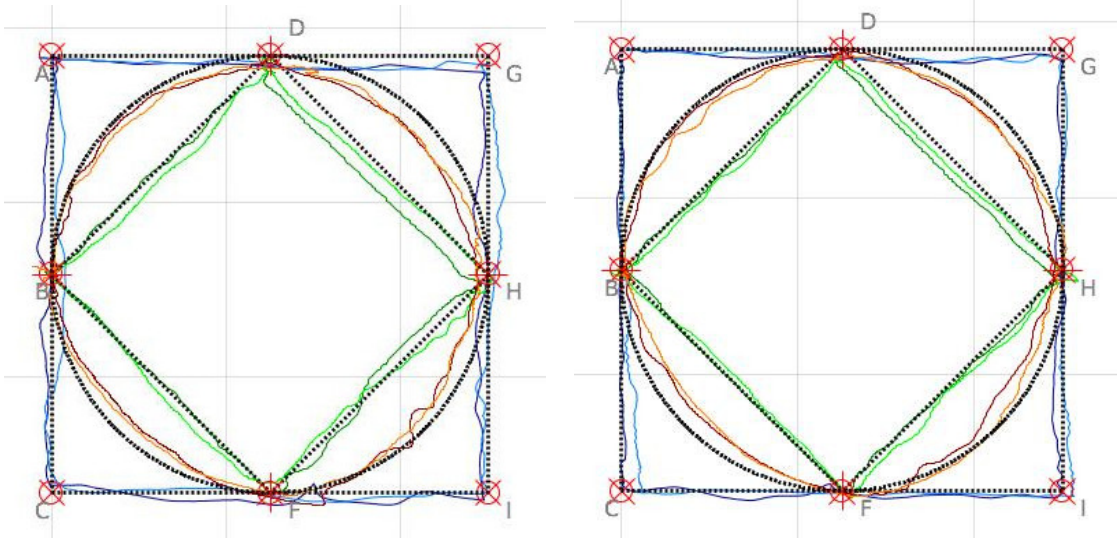


Figure 5.6: Trace of user while following the trajectory with PED disabled (left) and PED enabled (right) [5]

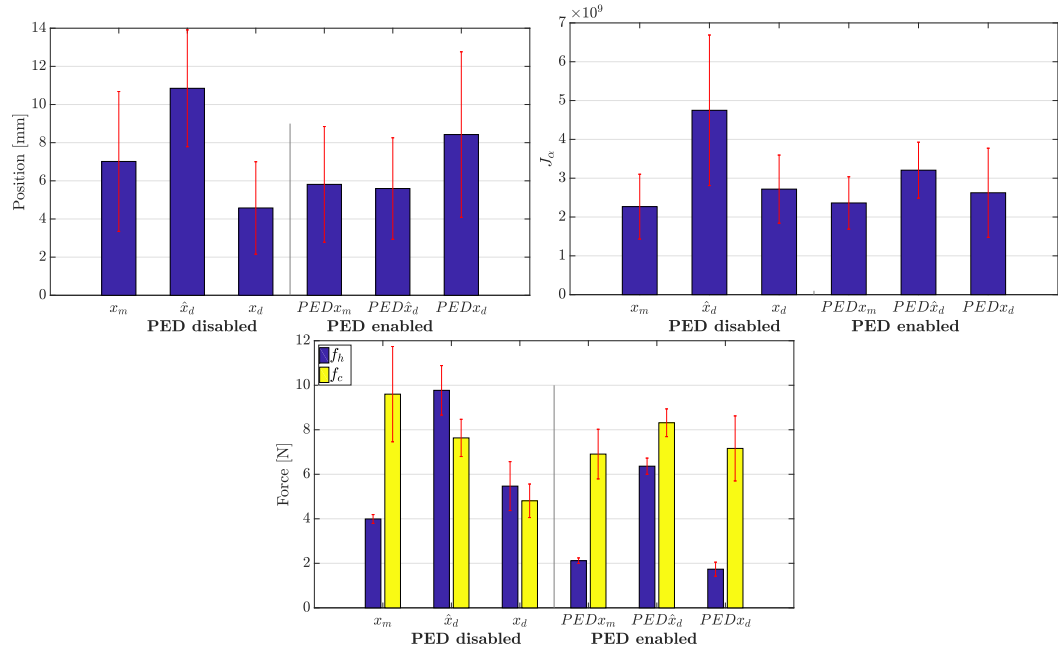


Figure 5.7: Performance measure plots [5]

advantage of HIE is that it doesn't need prior training on the robot platform making it versatile and portable.

We conducted preliminary experiments with a single user to investigate the effect of the hidden neuron layer size on the controller performance. The size of the NAC hidden layer was increased from 18 to 22, 25, and then 30. Results are summarized in Fig. 5.8, depicting

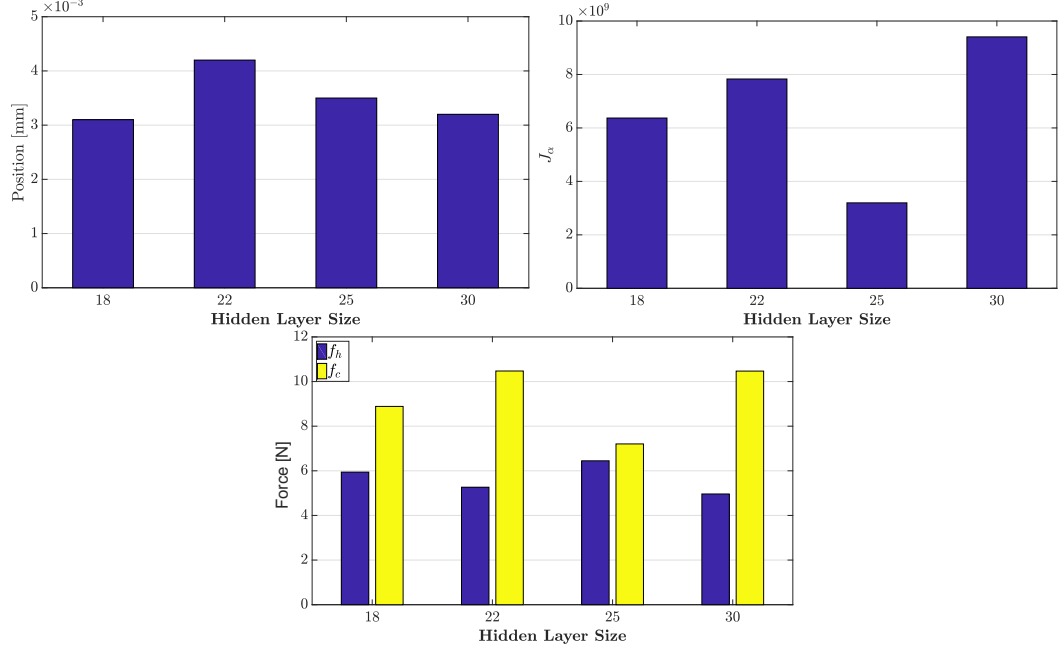


Figure 5.8: Controller performance for different number of hidden layer neurons (PED enabled with reference trajectory  $\hat{x}_d$ ). [5]

the mean squared jerk, position error, and force magnitude across all three patterns (circle, diamond, and square) repeated twice. Based on these experiments, we selected  $k = 25$  as the size of the hidden layer for all subsequent user experiments.

In future this controller is planned to be validated on the ARNA robot in the patient walker scenario. A comprehensive evaluation of this controller can be performed through user testing in different scenarios. The size and layer of the neural networks involved can be further investigated to determine the efficient size of the networks needed to better the pHRI performance.

## CHAPTER 6

### BASE-SENSOR ASSISTED PHYSICAL INTERACTION (BAPI)

This chapter details the research work conducted on the base-force torque sensor configuration to enable whole body sensing on a robot manipulator. In this configuration a force-torque sensor is attached to the base of the manipulator to sense interaction forces. This chapter contains the two separate research studies that were conducted with this setup. In the first study, the setup is used for collision detection where as in the second study, the joint torque of the manipulator is estimated from the force-torque sensor data.

Section 6.1 illustrates the research conducted to detect collision to make HRI safe and section 6.2 describes the research to estimate joint torques of a manipulator by sensing the reaction forces at its base. These research studies were published in [6] and [7] respectively.

#### **6.1 Whole-body Collision Detection using BAPI**

##### **6.1.1 Introduction**

As the field of robotics and automation is continuously evolving from structured industrial environments to more complex and uncertain workspaces shared with human workers, robots increasingly need to be equipped with sensing mechanisms in order to perceive and interact with their surroundings. Safe Human-Robot Interaction or HRI has been explored for a long time to integrate human users and robots in a shared workspace [104]. In the case of industrial robots, safety becomes even more challenging as the robots are faster and larger. Many studies have been conducted in this field using various approaches to make the robots safer to operate and to facilitate a common workspace for both robots and human users. Common approaches to the solution are to use a WFTS at the end-effector of a robot, covering the robot with tactile skin sensors [105] [106] [107], estimation

of interaction forces using measurements from tactile skin and extended Kalman filtering based algorithm [108], and, more recently, to use neuroadaptive control algorithms along with tactile sensors to accurately detect interaction forces [109] [45].

The use of force-torque sensors (FTS) at the end-effector of a robot arm only facilitates sensing of forces at the wrist. Covering the robot with tactile skin will expand the interaction sensing abilities to the whole arm, but, in such a case, complex sensor hardware and acquisition interfaces need to be provided. Moreover, utilizing these sensors for feedback has to be integrated with the control system of the robot to react to the detected user-force applications or collisions. Such integration requires prior knowledge of the robot system model and its physical parameters.

In contrast, the work presented here addresses these challenges by enabling whole body collision detection using a model-free, Neuro-Adaptive controller. Specifically, we explore the idea of using a Force-Torque sensor (FTS) at the base of a robot arm which allows detection of forces along the whole body of the robot. However, using this configuration, the BFTS detects forces from users, or due to collisions with the environment, as well as dynamical forces exerted at the base due to the movement of the robot arm. A novel method is introduced to separate the estimated user-forces from the robot dynamics by the use of an additional learning neural network (NN) controller.

Several studies have been conducted in the past using the FTS at the base of a robot manipulator. These include estimation of robot mass properties by West et. al. [110], inertial parameter estimation by Liu et. al. [111], estimation of other dynamic parameters by Grotjahn & Heimann [112], and Morel and Dubowsky [113]. In all these examples, knowledge of the system kinematics as well as the structural form of the robot dynamics are needed for estimation of robot chain masses and inertia. Nonlinear robot terms can then be compensated for increased tracking control precision of the arm. For instance, Morel & Dubowsky used a base FTS with a PUMA 550 to accurately control the position of the manipulator with an error smaller than  $30\mu m$ . Another such study of BFTS integrated admittance and impedance controller have been presented by Ott & Nakamura [114,115].

These controllers exploit robot passivity properties and make use of detailed knowledge of robot dynamic parameters.

Since force-torque sensors can be attached to both wrist (WFTS) and at the base (BFTS) of a robot manipulator, a study by Geffard et. al. [116] shows that as compared to WFTS, torque control with BFTS is better for back-drivability of a robot arm, increasing its safety.

Human-robot collision detection for safe robot operation with a human user has also been investigated by Lu et. al. [32, 33]. In particular, Human-robot collisions are detected using a trained neural network [33], using both a WFTS and a BFTS. In the studies on safe robot operation, it has been experimentally established that a human user can tolerate a collision force of  $50N$  [34]. Therefore, any controller that is implemented to detect collision with a human user, needs to be able to detect a force of less than  $50N$ .

Research has also been conducted on using neural networks (NN) to estimate the dynamic parameters of a robotic arm, in particular, landmark research from Lewis et. al. [16, 17]. The NN approach eliminates the need for an accurate robot dynamic model and allows joint error tracking within predictable bounds. The NN weights are adapted online to the inputs and errors in every controller update cycle to provide an output. In our recent work, we have expanded these ideas to physical HRI studies with robot arms, in which the interaction admittance is adapted concurrently with the robot model to match a given interaction task model. The resulting Neuro-Adaptive Controller (NAC) structure leads to a model-free control scheme for robots [44, 45].

The work presented here extends the NAC method to detect whole arm collisions and interaction forces by formalizing a new Base-sensor Assisted Physical Interaction (BAPI) controller. This method allows estimating the forces working on the robot arm using a BFTS and without any prior knowledge of the dynamic parameters of the system. An additional NN is introduced to separate the dynamical effects of robot motion from interaction forces. The advantages of our method are 1) ease and portability of implementation and 2) providing Lyapunov guarantees of convergence. We verify our work through experiments on

a custom 6-DOF robot arm, shown in Fig. 6.1.

The paper is organized as follows: Section 6.1.2 describes the BAPI controller, Section 6.1.3 details the experimental setup and the experiments conducted. Section 6.1.4 discusses the results obtained from the experiments.

### 6.1.2 Base-Sensor Assisted Physical Interaction (BAPI)

This section describes in detail the BAPI controller in three parts: 1) System dynamics of a robot arm equipped with a BFTS, 2) Neuro-Adaptive Controller or NAC for estimation of robot dynamics and 3) Base Forces Estimator or BFE for identification of interaction forces exerted by the robot arm. Because the NAC formulation has been extensively described in previous publications [109], [45], [17], [16], the reader is directed there for a more detailed derivation and the Lyapunov stability proof. The overall BAPI controller block diagram is depicted in Fig. 6.2, in which, the top part represents the NAC and the bottom part represents the BFE.

#### 6.1.2.1 BFTS System Dynamics

Since the BFTS is able to sense not only the user-forces or collisions of the robot arm, but also the twists due to the dynamical movement of the robot, it is necessary to separate the dynamic terms and the user interaction terms. Taking the robot system shown in Fig. 6.1, the robot dynamics with a human user's force acting on the robot arm with  $n$  joints can be written as:

$$\Lambda(q)\ddot{q} + C(q, \dot{q})\dot{q} + \Psi(q) + \tau_d = \tau + \tau_{user}, \quad (6.1)$$

where,  $\Lambda(q)$  represents the inertia matrix,  $C(q, \dot{q})$  represents the non-linear Coriolis term,  $\Psi(q)$  represents the gravity vector and  $\tau_d$  represents the bounded internal disturbances.  $\tau$  represents the control torque and  $\tau_{user}$  represents the torque induced on the joints due to user force and  $q(t) \in \mathbb{R}^n$  is the joint vector.

Let the force from user be represented by  $F_{user}$  at a point  $P_6$  (Fig. 6.1). Then:

$$\tau_{user} = J_{P_B P_6}^T(q) F_{user}, \quad (6.2)$$

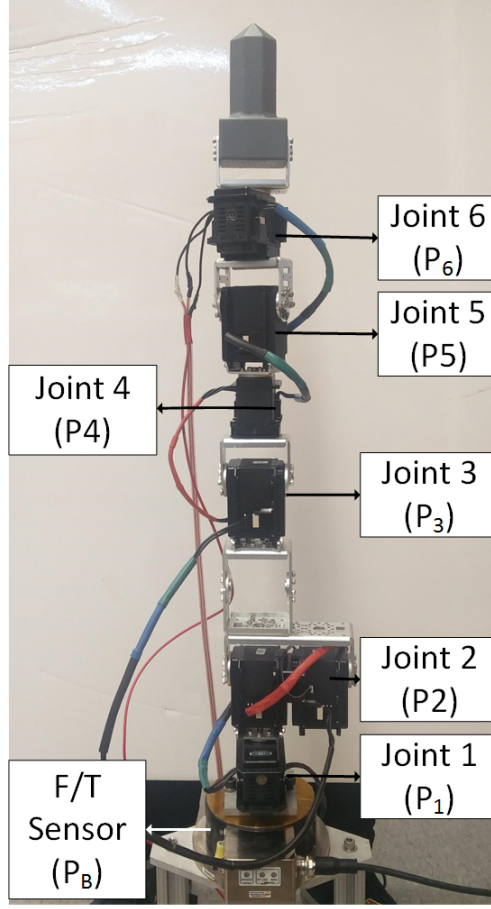


Figure 6.1: 6-DOF Robot manipulator constructed using torque-controlled Dynamixel motors MX-106, with an FTS at base [6]

in which  $J_{P_B P_6} \in \mathbb{R}^{6 \times n}$  represents the Jacobian between the point of contact  $P_6$  and the base of the robot  $P_B$ . Now the dynamics at the BFTS can be represented by [114] [33]:

$$\bar{\Lambda}(q)\ddot{q} + \bar{C}(q, \dot{q})\dot{q} + \bar{\Psi}(q) = \tau_{user\_wrench} - F_b, \quad (6.3)$$

where,  $q(t) \in \mathbb{R}^n$  is the joint vector,  $\bar{\Lambda}(q)$ ,  $\bar{C}(q, \dot{q})$  and  $\bar{\Psi}q$  are the coupling terms between the manipulator and the BFTS representing the inertial, Coriolis and gravity terms.  $F_b$  represents the wrench experienced at the base due to forces acting on the robot arm as well as the wrenches due to robot movements.

The value of  $F_b$  is sensed by the BFTS, while  $\tau_{user\_wrench}$  is the wrench exerted at the base due to the user forces  $F_{user}$  at  $P_6$  on the robot and is defined as [117]:

$$\tau_{user\_wrench} = Ad_{H_{P_B P_6}^{-1}}^T F_{user}, \quad (6.4)$$

in which  $H_{P_B P_6} \in SE(3)$  is the homogeneous transformation matrix between the point  $P_6$  and base  $P_B$ .  $H_{bp}^{-1}$  and  $Ad_{H_{P_B P_6}}^{-1}$  gives the adjoint of the homogeneous transformation function, defined as [117]:

$$Ad_{H_{P_B P_6}}^{-1} = \begin{bmatrix} R_{P_B P_6}^T & 0 \\ -R_{P_B P_6}^T \hat{d}_{P_B P_6} & R_{P_B P_6}^T \end{bmatrix}, \quad (6.5)$$

where,  $\hat{d}_{P_B P_6}$  is the skew-symmetric matrix of  $d_{P_B P_6}$ , the  $3 \times 1$  position vector in the homogeneous transformation matrix, and  $R_{P_B P_6}$  is the rotation component.

Equations Eq. 6.1 and Eq. 6.3 can be used to detect and estimate the user forces. From Eq. 6.3, if  $\bar{\Lambda}(q)\ddot{q} + \bar{C}(q, \dot{q})\dot{q} + \bar{\Psi}(q) = -X(\ddot{q}, \dot{q})$  is estimated, then the user force  $F_{user}$  can be calculated by:

$$-X(\ddot{q}, \dot{q}) = \tau_{user\_wrench} - F_b \quad (6.6)$$

$$\implies -X(\ddot{q}, \dot{q}) = Ad_{H_{P_B P_6}}^{-1} F_{user} - F_b \quad (6.7)$$

$$\implies F_{user} = (Ad_{H_{P_B P_6}}^{-1})^{-1} (-X(\ddot{q}, \dot{q}) + F_b). \quad (6.8)$$

A NN can be trained to learn  $-X(\ddot{q}, \dot{q})$  against an expected output of  $F_b$ , and as per Eq. 6.7, when no user forces  $F_{user}$  are acting on the robot, these two terms are equal. This training process is described in detail in section 6.1.2.3.

To run the robot, the controller has to estimate, and feed forwardly compensate the robot dynamics in Eq. 6.1. This estimation is accomplished using a Neuroadaptive Controller (NAC) elaborated on in section 6.1.2.2.

### 6.1.2.2 Neuro-Adaptive Controller (NAC)

The NAC is responsible for estimating the robot dynamics and providing torque input to its joints to cancel nonlinearities. These torque values are adjusted by the NAC on-line, based on the error between the desired and actual joint trajectories, with the goal of driving the tracking error to zero. This section describes briefly the formulation of NAC adapted to the robot BFTS.



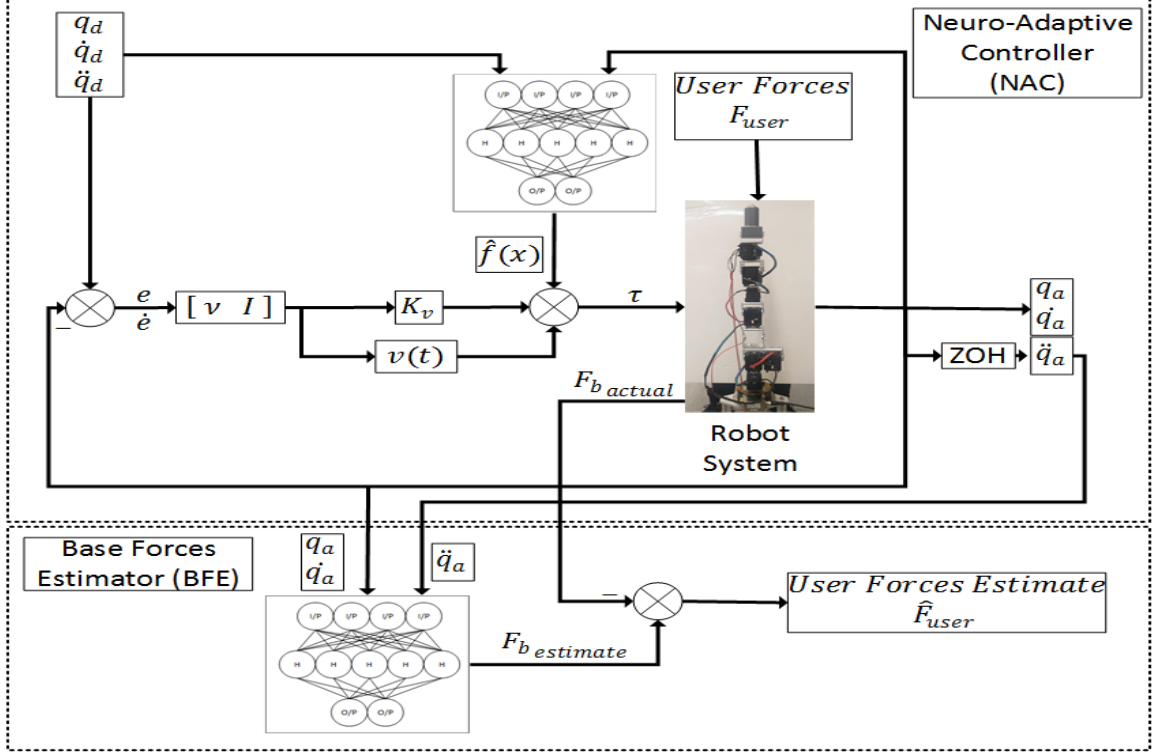


Figure 6.2: BAPI controller block diagram. The upper portion represents the NAC non-linear compensator, while the BFE NN in the lower block estimates the robot interaction force at the base [6]

Starting with Eq. 6.1, we can write:

$$\Lambda(q)\ddot{q} + C(q, \dot{q})\dot{q} + \Psi(q) + \tau_d = \tau \quad (6.9)$$

We define the error and sliding mode error in following a desired trajectory  $q_d$  as described in [16]:

$$e = q_d - q, \quad (6.10)$$

and

$$r = \dot{e} + \nu e, \quad (6.11)$$

where  $\nu$  is a symmetric positive definite matrix which is used as a design parameter for the system. The error dynamics of the system is then represented by:

$$\Lambda(q)\dot{r} = -C(q, \dot{q})r + f(x) + \tau_d - \tau, \quad (6.12)$$

and  $f(x)$  is defined as:

$$f(x) = \Lambda(q)(\ddot{q}_d + \nu\dot{e}) + C(q, \dot{q})(\dot{q}_d + \nu e) + \Psi(q). \quad (6.13)$$

Notice that as the error  $e$  goes to zero,  $r$  also goes to zero in Eq. 6.11 and vice-versa. Defining the control torque as:

$$\tau = \hat{f}(x) + K_v r - v(t), \quad (6.14)$$

where  $\hat{f}(x)$  is an approximation of  $f(x)$  and  $K_v$  represents a constant gain matrix. A robustifying term  $v(t)$  is also added to the torque to compensate for any unmodeled forces. Combining Eq. 6.12 and 6.14, we obtain:

$$\Lambda(q)\dot{r} = -C(q, \dot{q})r + \tau_d + \tilde{f}(x) - K_v r + v(t), \quad (6.15)$$

In which  $\tilde{f}(x)$  represents the error in estimating  $f(x)$ . An I/O equation for the 1 hidden layer NAC neural network can be written as an approximator for  $f(x)$ :

$$f(x) = W^T \sigma(V^T x) + \epsilon, \quad (6.16)$$

in which  $W$  and  $V$  are the required NN weights,  $\sigma$  is the sigmoid activation function of the hidden layer of the neural network and  $x$  is the input vector to the NN. Let  $\hat{W}$  and  $\hat{V}$  be the approximation of the ideal weights then, the control torque in Eq. 6.14 becomes:

$$\tau = \hat{W}^t \sigma(\hat{V}^T x) + K_v r - v, \quad (6.17)$$

where  $x$  is a vector containing  $[e^T \ \dot{e}^T \ q_d^T \ \dot{q}_d^T \ \ddot{q}_d^T]$ . In [16, 17, 45, 109], it was shown that appropriate weight update equations can be found to guarantee convergence of the tracking errors to zero via a Lyapunov proof. We direct the reader to our previous publications for details omitted here.

### 6.1.2.3 Base Forces Estimator (BFE)

We introduce BFE as a new controller block, which estimates the dynamic interaction forces acting on the robot base, and separates the user pushing force from the dynamical loads due to the movements of the robot arm. The BFE is implemented as a single hidden layer NN trained with appropriate inputs and outputs described in Fig. 6.2 when no outside user/collision forces are acting on the robot body. From Eq. 6.7, if  $F_{user} = 0$ , then

$X(\ddot{q}, \dot{q}) = F_b$ . Therefore, the estimation of  $-\hat{X}(\ddot{q}, \dot{q})$  can be accomplished as a nonlinear equation with three input variables  $Y = [\ddot{q} \ \dot{q} \ q]$  and  $F_b$  as the output. Therefore, a NN can be used to estimate such an equation system using:

$$-\hat{X} = A^T \rho(B^T Y) + \epsilon, \quad (6.18)$$

where,  $A$  and  $B$  are the output and input weights of the BFE NN respectively and  $\rho$  is a hyperbolic tan-sigmoid activation function of the NN.

The BFE requires one-time training to estimate the dynamic parameters and then the trained weights can be used online during normal operation to detect collision. The output of the BFE NN would be the estimated forces measured at the base in case of no user-forces/collisions, whereas the actual sensor data would contain both the user-forces/collisions as well as the dynamic interaction forces. If the output of the BFE NN is represented by  $F_{b\_estimate}$ , then by using Eq. 6.7, we can separate the human-robot or the robot environment interaction forces from the rest of the dynamics:

$$Ad_{H_{P_6}^{-1}}^T F_{user} = F_{b\_estimate} - F_b \quad (6.19)$$

$$\implies \tau_{user\_wrench} = F_{b\_estimate} - F_b \quad (6.20)$$

$$\implies \tau_{user\_wrench} = \hat{F}_{user} \quad (6.21)$$

The training of the NN weights in Eq. 6.18 can be done off-line, through classical NN training methods. A Levenberg-Marquardt based NN training algorithm was implemented for this paper using the Matlab toolbox [118]. The training algorithm uses a Hessian matrix and a gradient term, which are defined below, for updation of the weights:

$$H_{nn} = J_{nn}^T J_{nn} \quad (6.22)$$

and

$$\Delta = J_{nn}^T e_{nn}. \quad (6.23)$$

The term  $H_{nn}$  represents the Hessian matrix,  $J_{nn}$  is the Jacobian matrix of the errors w.r.t NN weights,  $\Delta$  is the gradient term and  $e_{nn}$  is the error of the NN. Using these terms, the

weights are updated by:

$$W_{new} = W_{old} - [H_{nn} + \mu I]^{-1} \Delta, \quad (6.24)$$

where  $\mu$  is a scalar design parameter and its value is modified based on the performance of the NN.

### 6.1.3 Experimental Testbed and NN Training

In order to validate the BAPI controller introduced in section 6.1.2, experiments were conducted on a custom 6-DOF robot which was constructed using Dynamixel MX-106 motors from Robotis, South Korea. An EtherCAT enabled Delta FTS from ATI, USA was placed at the base of the robot arm. This setup can be seen in Fig. 6.1. The robot arm as well as the FTS are connected to an EtherCAT network to ensure real-time capabilities of the system and data synchronization. A Teensy 3.6 micro-controller and an EtherCAT PRO module from Bausano, Italy, were used to construct an EtherCAT slave which was used to connect to the Dynamixel motors. The entire BAPI control loop was executed in real-time at 400Hz.

A desired trajectory based on sine wave inputs  $[q_d \dot{q}_d \ddot{q}_d]$ , with a position range of  $\pm\pi/4$  radians and phase difference of  $\pm\pi/2$  radians between subsequent joints, are provided to NAC which calculates the joint torques. The robot arm receives torque values to each joint as input and sends position and velocity of each joint as output. The velocity value is filtered using an IIR filter and put through a Zero Order Hold to get the acceleration value. NAC has 30 inputs ( $[e \dot{e} q_d \dot{q}_d \ddot{q}_d] \times 6$ ), 10 neurons in hidden layer and 6 outputs ( $\tau = [\tau_1 \tau_2 \tau_3 \tau_4 \tau_5 \tau_6]$ ). No off-line training is required for this network as it estimates the dynamic parameters of the robot online.

Before the NAC can be run, the BFE NN is trained by running the arm with predefined, sufficiently exciting trajectories using NAC and not applying any external force. The NN was trained in Matlab NN Toolbox using Levenberg-Marquardt algorithm over 200,000 data points and validated over another separate set of 200,000 data points. The NN has 18 inputs ( $[\text{position, velocity, acceleration}] \times 6$ ), 30 neurons in hidden layer and 6 outputs  $[F_x,$

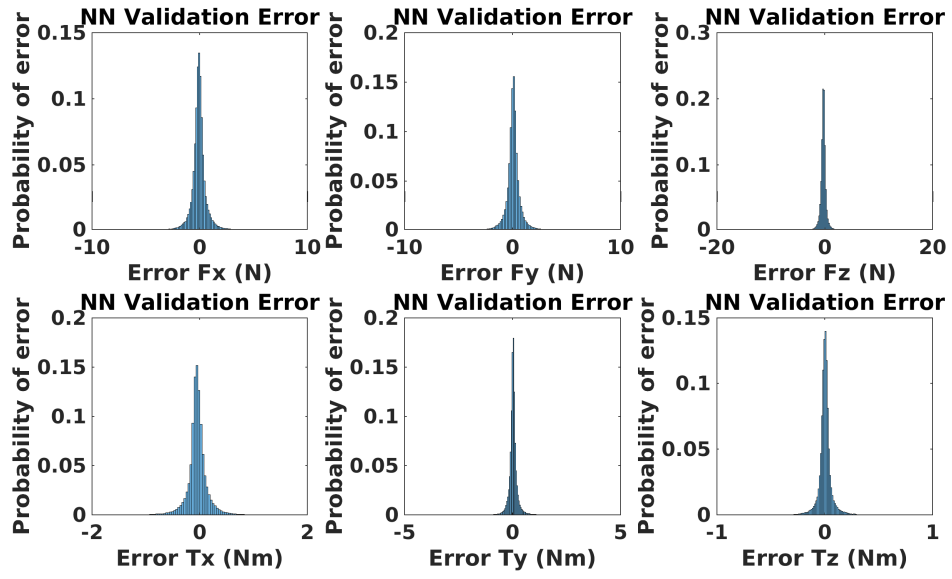


Figure 6.3: NN Validation Error Distribution [6]

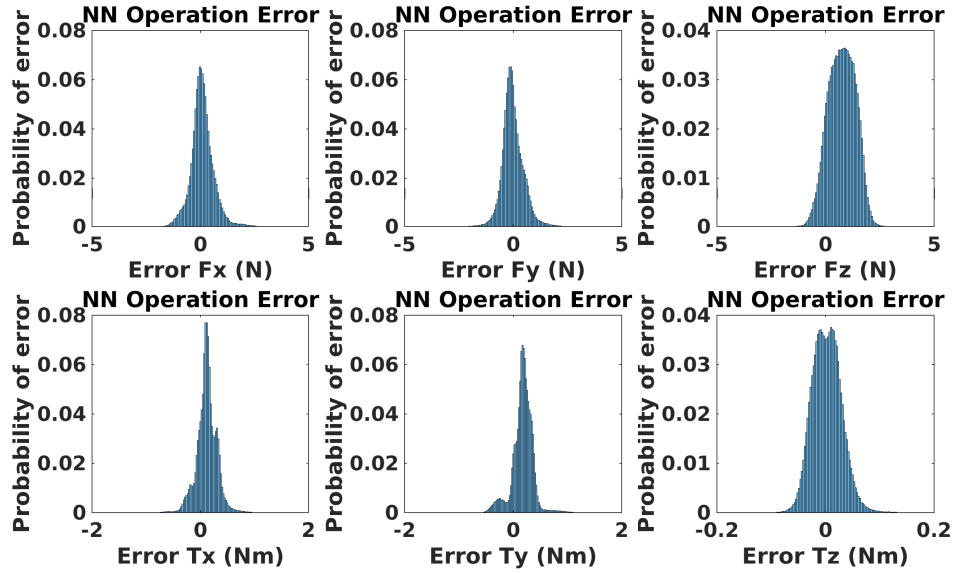


Figure 6.4: NN Operation Error Distribution [6]

Fy, Fz, Tx, Ty, Tz]. The trained weights and biases are then made saved for the operation of the arm.

The mean square error (MSE) between the BFE NN output and the FTS were calculated as follows:

$$MSE = [(Fx_{nn} - Fx_{exp})^2 + (Fy_{nn} - Fy_{exp})^2 + (Fz_{nn} - Fz_{exp})^2 + (Tx_{nn} - Tx_{exp})^2 + (Ty_{nn} - Ty_{exp})^2 + (Tz_{nn} - Tz_{exp})^2]^{1/2} \quad (6.25)$$

where, the subscript *nn* represents NN output and the *exp* represents the expected output from the NN which is also the output of the BFTS. *N* represents the number of samples being evaluated.

A histogram analysis of the MSE, over the period of experiments, gives the probability of error, which is a metric of NN performance. Fig. 6.3 shows the probability of error distribution during the NN validation operation. The y-axis of the subplots represents the probability of error and the x-axis depicts the error in forces and torques. The errors are observed to be bounded and centered around 0.

The BFE NN was then used online with the NAC to estimate forces and torques. Fig. 6.4 shows the probability of error distribution when sensor data was compared against NN estimates at zero interaction forces. The graph shows similar result as compared to the validation results. The graphs also show that the error in forces are bounded about  $\pm 2N$  and the torque errors are bounded about  $\pm 1Nm$ .

Collision experiments were then conducted by running the robot system for 120 seconds and applying user-forces every 20 seconds. The experiments were conducted on all 6 joints to show the robustness and versatility of the proposed solution. These results are discussed in the following section.

#### 6.1.4 Results and Discussion

Fig. 6.5 shows the result of experiment on joint 1 of the robot system and Table 6.2 shows the experiments conducted on all six joints and the corresponding MSE value at that

time. The difference between the BFE NN output and actual BFTS data was considered as the error of the BFE NN. This error, as per Eq. 6.20, also represents the  $\tau_{user\_wrench}$ . The mean square error (MSE) analysis reveals that at every collision during the experiment process, the value of MSE increases sharply. This spike in MSE data represents the detection of  $\tau_{user}$ . If a softbound of  $MSE = 2$  is drawn, then any MSE value above the softbound can be recorded as a collision event.

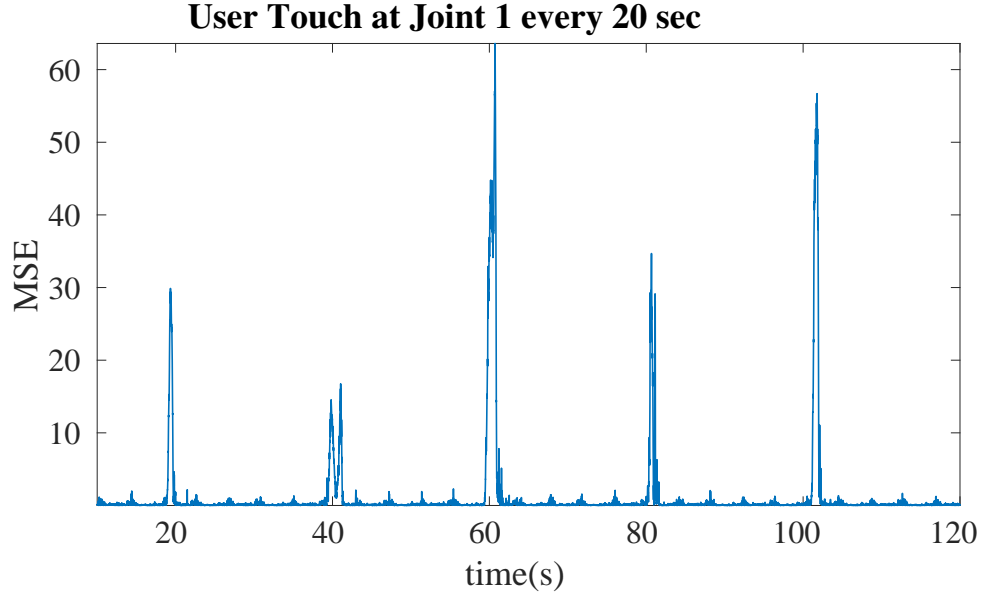


Figure 6.5: User interaction forces at Joint 1 ( $P_1$ ) at 20s, 40s, 60s, 80s and 100s [6]

TABLE 6.1

## Experimental Results

	MSE Data for Collisions at time interval in seconds				
	20s	40s	60s	80s	100s
Joint1	29.88	15.26	63.6	34.68	56.69
Joint2	62.08	26.04	26.45	37.46	26.56
Joint3	27.08	98.17	65.74	12.33	36.22
Joint4	29.38	22.15	31.08	28.18	31.44
Joint5	67.63	142.7	43.31	102.8	121.3
Joint6	61.07	260	105.5	65.15	136.6

The low noise jitters visible in the graph are caused due to the wrench exerted on the BFTS when one of the links in the robot changes direction while following the required sine-wave path. These forces are accounted for by the BFE NN and therefore these events are not recorded as collisions.

It can also be observed that the MSE values vary for different experiments. This is due to the fact that low interaction forces will yield low MSE and high interaction forces will yield higher MSE values. Another factor affecting the value of MSE is the location of application of forces on the robot arm as well as its configuration at the time. These factors affect the value of  $Ad_{H_{P_B P_6}}^{-1T}$  from Eq. 6.20.

These results also show the quick detection of the forces at every 20 seconds when the actual collision occurred. Such robust detection can be used in a robot system to react to the collision either by stopping or moving away from the point of collision. The results also underline the fact that the detection of collision doesn't depend on the location of the collision. One question may arise on the justification of use of the BAPI controller as any disturbance beyond the operating range of the BFTS can be detected as a collision and a mere soft-bound on the sensor data is enough to detect collision. To address this question, we direct the reader to the results shown in Fig. 6.6, 6.7 and 6.8. Fig. 6.8 and



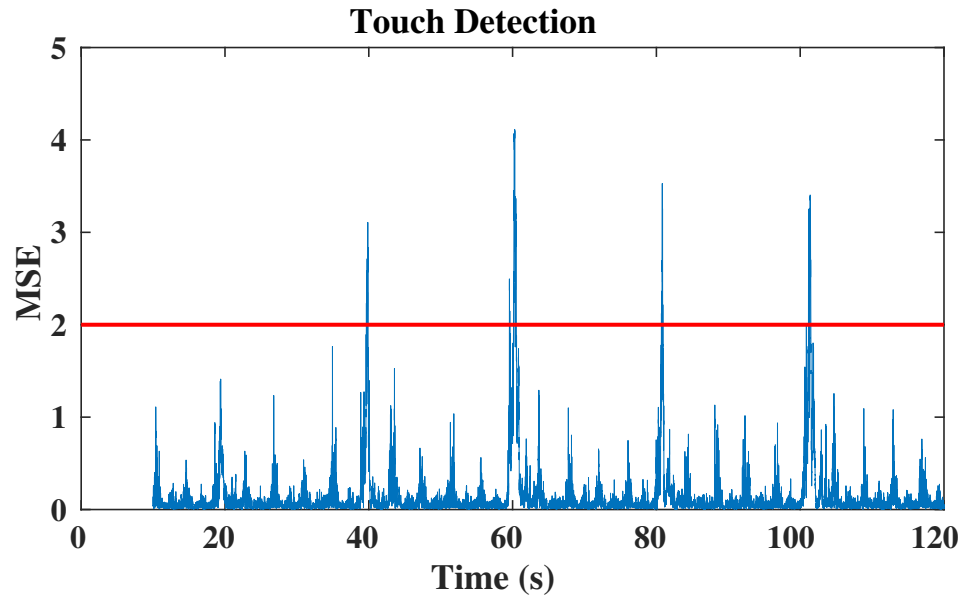


Figure 6.6: Low Interaction Forces at Joint 6 ( $P_6$ ) [6]

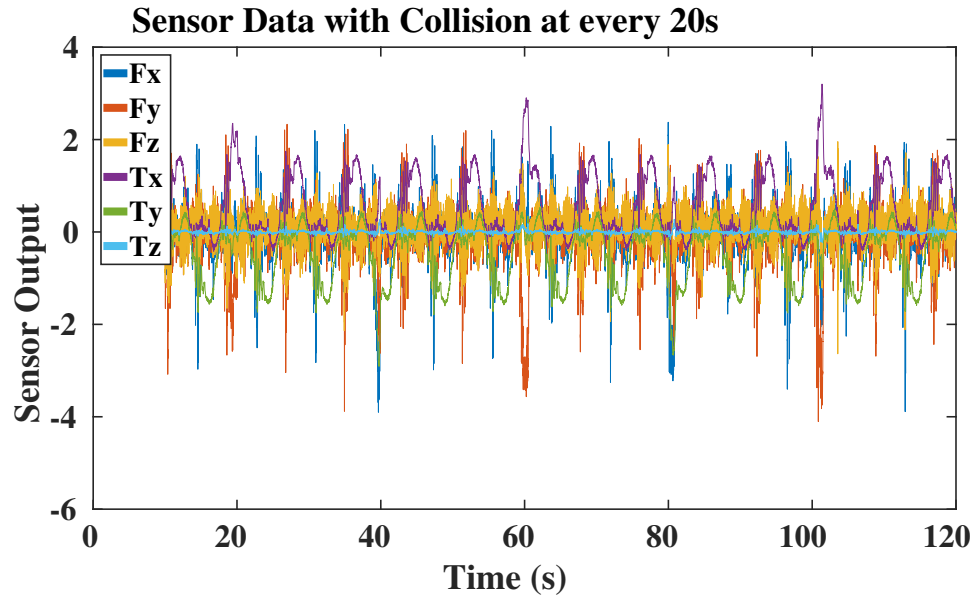


Figure 6.7: Sensor data (forces and torques) from BFTS during collision [6]

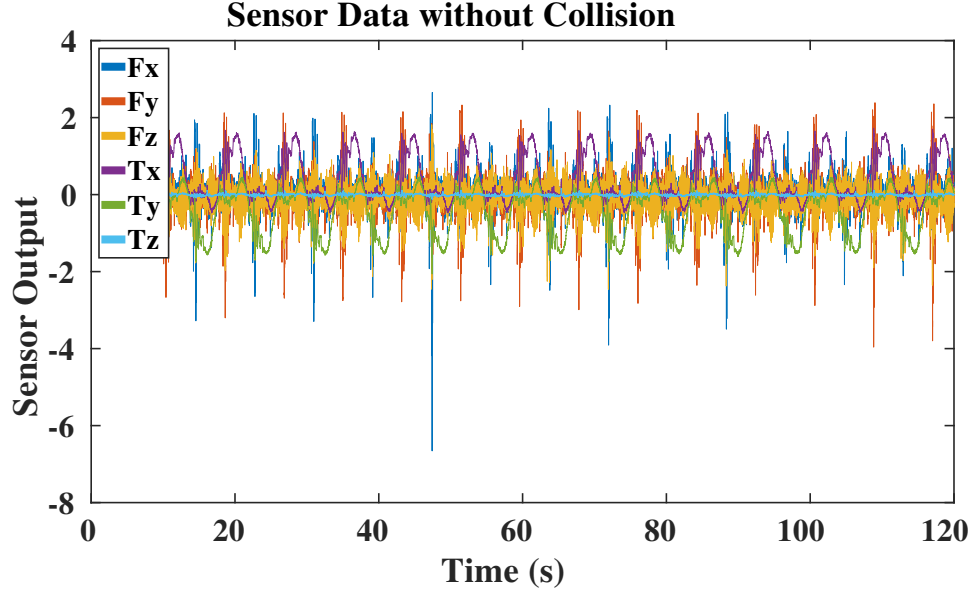


Figure 6.8: Sensor data (forces and torques) from BFTS during no collision [6]

Fig. 6.7 represent the sensor data during normal operation and during collisions respectively. We observe that the sensor values as well as range of sensor output to be similar and no conclusive decision can be made on collision by having a softbound on the range of data. Whereas when compared to the base NN controller output and taking the mean square error of the results, we see 4 spikes in the data at 40, 60, 80 and 100 seconds representing collision depicted in Fig. 6.6. The collision at 20 seconds was not detected as it as the force of collision was so low that it fell under the error bounds of  $\pm 2N$  as discussed in section 6.1.1. These graphs also highlight the effectiveness and sensitivity of our proposed controller.

## 6.2 Joint Torque Estimation using BAPI

### 6.2.1 Introduction

Collaborative robots (co-robots or cobots) are intended to share their workspace with humans by physical interaction, while helping them perform ever-increasingly complex tasks. For this purpose, co-robots need to be equipped with advanced sensing schemes, which allow them to perceive their immediate surroundings, comprehend user's physical intent, and guarantee user safety [104].

Traditionally, co-robots have implemented intuitive physical human-robot interaction (pHRI) schemes and collision detection through the use of force-torque sensors at the end-effector [119] or other joints [120], or by integration of tactile robot skin technology [121]. Many commercial robots, however, are not equipped with joint force-torque sensors due to a number of practical considerations. For instance, instrumenting robots with joint force-torque sensors introduces structural flexibilities and nonlinearities and causes performance degradation [122]. Force-sensitive robot skins also are costly, adding to the wiring complexity of the hardware, and require a separate electronic data acquisition system and interfacing with the robot controller. These tactile robot skins also require the knowledge of the position of each tactile element so that the controller can accurately interpret the applied forces [45, 105–109].

In the case of wrist force-torque sensors (WFTS), interactive control can be obtained merely in response to forces applied to the end-effector, where the force-torque sensor is installed, rendering the robot blind to the forces applied along the robot’s remaining body. To mitigate this drawback, some researchers have utilized a six-axis force-torque sensor installed under the manipulator’s base instead of at the wrist and estimated the net force-torque acting along the manipulator’s links [33, 111, 113–115, 123–125]. This methodology obviates the need for joint sensors or incorporation of tactile robot skins on the robot outer shell and enables the implementation of compliant control schemes for interactions all along the manipulator’s structure. Use of a base force-torque sensor (BFTS) also allows to retrofit any robot arm which lacks a WFTS and facilitates in making the robot compliant. Use of BFTS also means detecting all the forces and torques caused due to the movement of the manipulator and the forces interacting on it. Any controller facilitating HRI with BFTS, needs to distinguish between the effect of interaction forces and the dynamic forces due to manipulator movement on the BFTS.

Liu et al. used the base force-torque measurements along with the joint velocities to identify inertial parameters of the manipulator [111]. They showed that their proposed identification algorithm was robust against unmodeled joint frictions and resulted in precise

estimation of inertial parameters even in the absence of acceleration measurements. In [33], Lu and colleagues used a model-based neural network as well as the measurements from two force-torque sensors mounted at the manipulators wrist and base to detect collisions between the robot and environment or human. Using BFTS, estimation of dynamic parameters of a robot was presented by Grotjahn & Heimann [112] while West et al. presented the estimation of robot mass properties [110]. Christoforou used the reaction forces at the manipulator's base to online estimate the robot's model parameters and enhance the trajectory tracking performance of a model-based control scheme as well as a nonlinear adaptive controller [124]. Buondonno and Luca combined the base force-torque measurements and model-based information on the contact link, provided by residual vectors, to develop an algorithm that efficiently estimated unknown parameters of the robot and enabled collision detection [125]. In [114] [115], Ott and Nakamura used the readings from a base force-torque sensor to implement an admittance controller. Various attributes of the proposed compliant control scheme were analyzed including stability, passivity, and steady-state properties.

In [113], Morel and Dubowsky utilized the feedback from the base sensor to estimate the individual joint torques and implemented a torque control scheme which nullified joint friction and gravity effects. They showed that, upon the elimination of those effects, a simple proportional-derivative position controller is adequate for a high-quality motion control even at small amplitude. In [123], Morel and colleagues proposed a straightforward approach, based on the base force-torque sensor, to cancel out joint frictions and attain a highly-precise motion control at low speed without the need for complex analytical friction models. Morel et al. also proposed that using low-velocity trajectory can make the dynamic effect on the BFTS negligible, thus rendering a simpler relationship between the joint torques and the BFTS reading.

Neural networks (NN) have also been used to estimate the dynamic parameters of robotics manipulator by Lewis et al. [16, 17]. This idea has also been further investigated in our recent work, where the control scheme is expanded to be used in pHRI scenarios. A Neuro-Adaptive Controller (NAC) was formulated which resulted in model-free control of

the robot [44,45]. Our recently proposed Base-sensor Assisted Physical Interaction (BAPI) controller, utilizes NAC and an additional neural network (NN) to estimate the dynamic forces experienced by BFTS [6]. The BAPI controller was used to detect collisions on the robot body to safely operate in an environment shared by human users. In this scheme, the NAC was used in conjunction with an off-line trained NN to estimate the magnitude of collision forces. While this controller was useful to detect the sudden collisions of the manipulator with the environment, it could not accurately estimate the interaction force and its location.

In this paper, we are extending our approach to an on-line, accurate estimator of interaction forces. We use a force-torque sensor attached to the base of a custom 6-DOF robot manipulator to validate our approach. The resulting controller arises naturally from the neuro-adaptive loop after estimating the robot dynamics at a real-time loop of 200 Hz. The advantages of our method are 1) A model-free approach, 2) co-manipulation capability without WFTS and 3) providing Lyapunov guarantees of convergence. Experiments show that a mean-square joint torque estimation error below 5% can be obtained with this method.

The paper is organized as follows: Section II describes the proposed controller; Section III details the experimental setup and the experiments conducted; Section IV discusses the results obtained from the experiments; Finally, section V concludes the paper with description of our future work.

### 6.2.2 Controller Formulation

In this section the theoretical basis of our controller is described. The controller consists of two control loops a Neuro-Adaptive Controller (NAC) and a Base-force estimator (BFE). The NAC is responsible for setting the torque of the robot arm, whereas the BFE is responsible for estimating external interaction forces acting on the robot. In this section we go over the system dynamics and the relationship between the wrench experienced at the base of a robot and the torques at the joints. Here, we describe the NAC in a succinct way,

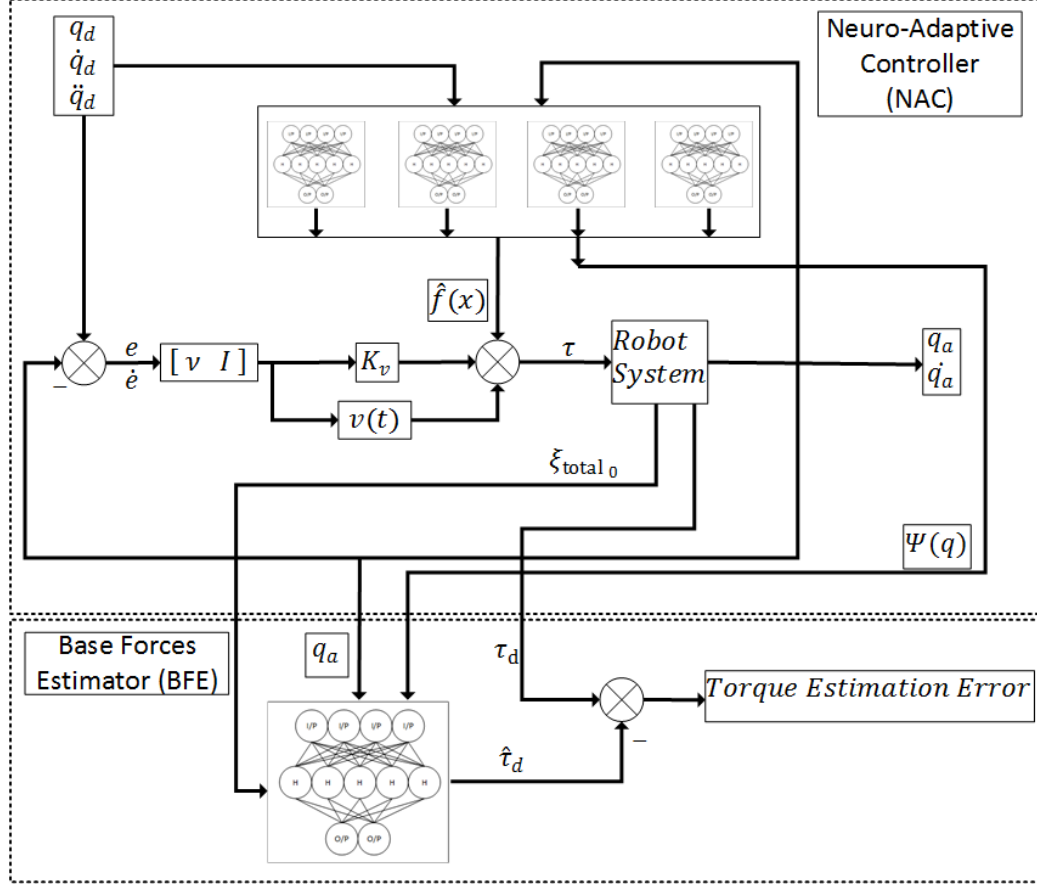


Figure 6.9: Proposed controller block diagram. The NAC (top) learns the robot dynamics, while the BFE (bottom) estimates the joint torques based on wrench experienced at the base of the robot. [7]

since it has already been covered in detail elsewhere [109], [45] [17] [16]. The classical NAC controller diagram is shown in the upper portion of Fig. 6.9. The controller consists of a neural network (NN) injected directly into the feed-forward loop to cancel out the robot nonlinearities. The NN can be implemented using a 1 or 2 layer “shallow” structure, and the weight update equations of the NN are selected such that the controller is stable. In this paper, we augment the original NAC with an on-line BFE block, as shown in the bottom portion of BAPI control architecture, illustrated in Fig. 6.9.

### 6.2.2.1 BFTS System Dynamics

This section describes the dynamics that relate the joint motor torques to the BFTS wrench. In case of a robotic manipulator shown in Fig. 6.10, the wrench experienced at a

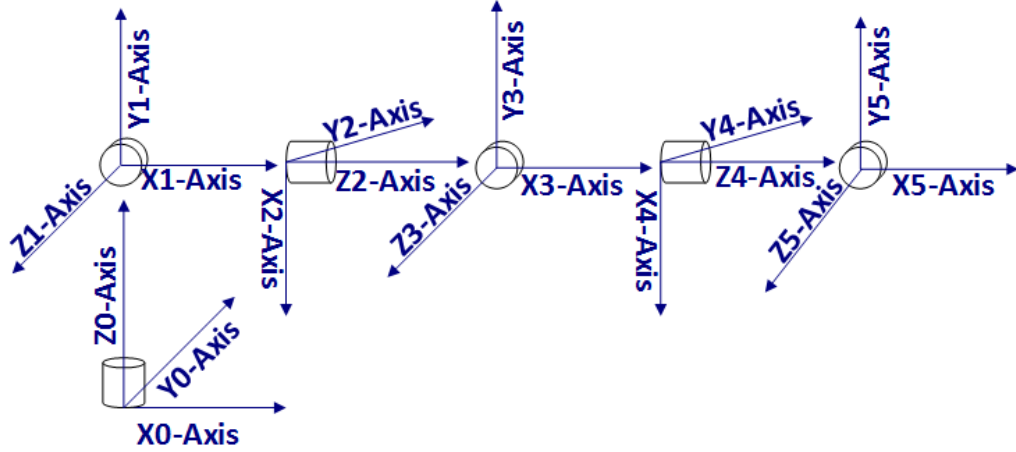


Figure 6.10: 6-DOF Robot manipulator coordinate system [7]

joint  $i$  can be described as [123]:

$$\xi_{total_i} = \xi_{dynamic_i} + \xi_{gravity_i} \quad (6.26)$$

where,  $\xi_{total_i}$  is the total wrench experienced by joint  $i$ ,  $\xi_{dynamic_i}$  is the dynamic wrench at joint  $i$  due to manipulator motion and  $\xi_{gravity_i}$  is the wrench due to gravity.

Furthermore, the wrench due to gravity at the base of the robot can be described as [123] [110]:

$$\xi_{gravity_0} = \begin{bmatrix} W_{total}\mathbf{g} \\ \mathbf{M}(\mathbf{q})\mathbf{n} \end{bmatrix} \quad (6.27)$$

where,  $W_{total}$  is the mass of the robot,  $\mathbf{g}$  is the acceleration due to gravity,  $\mathbf{M}(\mathbf{q})$  is the geometry matrix, representing the displacement vector of each joint from the BFTS origin.

Using Eq. 6.27 and 6.26, the dynamic wrench at the base can be written as:

$$\xi_{dynamic_0} = \begin{bmatrix} \mathbf{f}_{total_0} - m_{total}\mathbf{g} \\ \tau_{total_0} - \mathbf{M}(\mathbf{q})\mathbf{n} \end{bmatrix} \quad (6.28)$$

where,  $\mathbf{f}_{total_0}$  and  $\tau_{total_0}$  are part of total wrench,  $\xi_{total_0}$ , the total forces and torques experienced at the BFTS. Using these systems of equations, Morel et al. illustrated the relationship between joint torques and the wrench at the base of a manipulator as follows:

$$\tau_d = \alpha(\mathbf{q}) [\xi_{total_0} - \xi_{gravity_0}] \quad (6.29)$$

where,  $\tau_d$  is the joint torques of the manipulator and  $\alpha$  is defined as [123]:

$$\alpha_i(\mathbf{q}) = [\mathbf{z}_{i-1}^T \mathbf{S}_{O_{i-1}O_0} \quad \mathbf{z}_{i-1}^T] \quad (6.30)$$

The  $\mathbf{S}_{O_{i-1}O_0}$  is the skew-symmetric matrix of the vectors representing the displacement of joints w.r.t. the BFTS origin. The  $\alpha$  matrix depends only on the joint positions and Denavit-Hartenberg (DH) parameters, i.e. the robot kinematics.

Morel et al. state in their previous study that if the robot is operated in a very small region, the effect of  $\xi_{gravity_0}$  can be considered static and thus the joint torques can be calculated by just using the BFTS data and  $\alpha$  [123].

In order to calculate the joint torques while maneuvering in the general 3D workspace,  $\xi_{gravity_0}$  must be estimated. Once the  $\xi_{gravity_0}$  is estimated, a neural network can be trained to learn the  $\alpha(\mathbf{q})$  matrix as well. The subsequent sections elaborate on the method to calculate the terms in Eq. 6.29.

### 6.2.2.2 Neuro-Adaptive Controller (NAC)

The Neuro-Adaptive Controller or NAC, as shown in Fig. 6.9, is used to drive the robot to follow a desired path  $q_d$ . The NAC contains a neural network which estimates the robot's dynamic model and outputs joint torques to drive the error between the desired and the actual trajectory to zero. The Euler-Lagrange equation of any robot arm can be defined as follows:

$$\Lambda(q)\ddot{q} + C(q, \dot{q})\dot{q} + \Psi(q) + \eta(\dot{q}) + \tau_d = \tau \quad (6.31)$$

where,  $\Lambda(q)$  is the inertia matrix,  $C(q, \dot{q})$  is the Coriolis term,  $\Psi(q)$  is the torque due to gravity and  $\eta(\dot{q})$  is the friction term.  $\tau_d$  is the disturbance and  $\tau$  is the joint torque.

The error dynamics of the system is then defined as:

$$\Lambda(q)\dot{r} = -C(q, \dot{q})r + f(x) + \tau_d - \tau \quad (6.32)$$



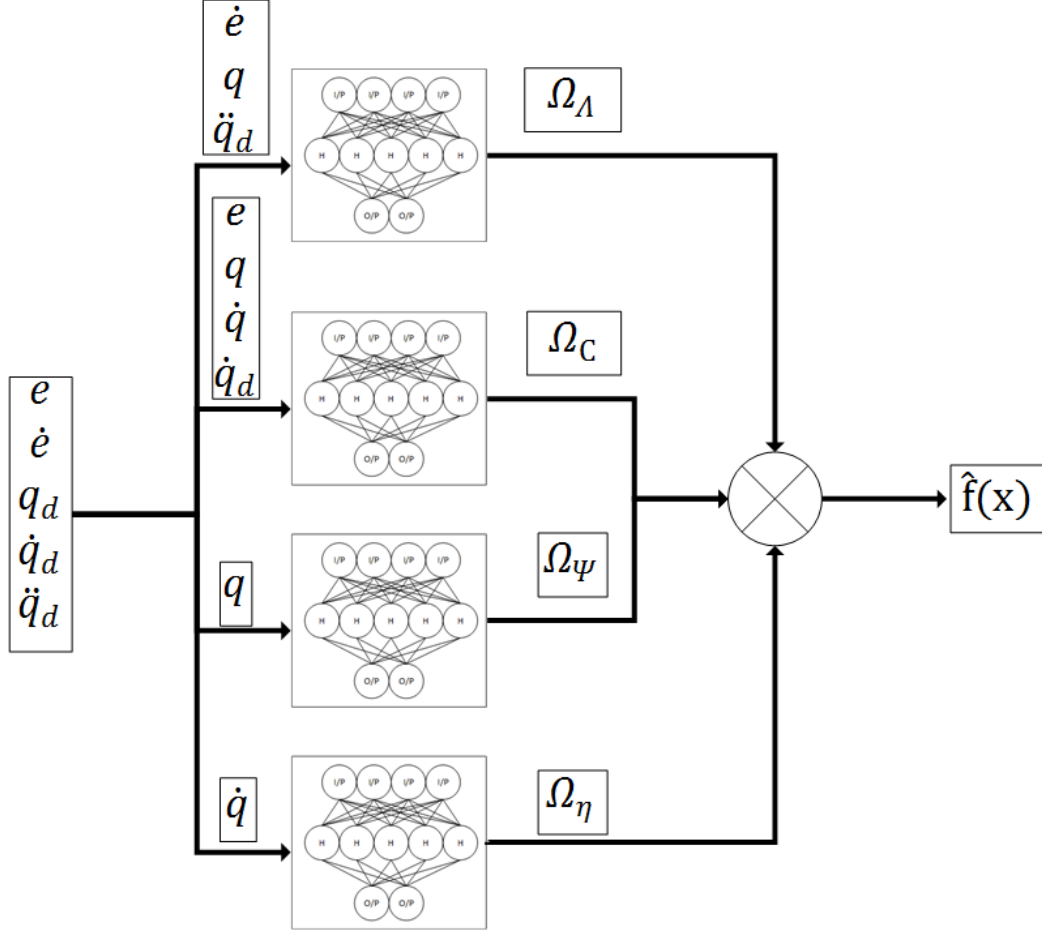


Figure 6.11: Segmenting NAC NN to extract dynamics terms. [7]

where,  $r$  is the filtered sliding mode error, and  $f(x)$  represents a combination of reference trajectory dynamics given by:

$$f(x) = \Lambda(q)(\ddot{q}_d + \nu\dot{e}) + C(q, \dot{q})(\dot{q}_d + \nu e) + \Psi(q) + \eta(\dot{q}) \quad (6.33)$$

where,  $\nu$  is a symmetric positive definite matrix and is used to define  $r$  as  $(\dot{e} + \nu e)$ .  $e$  is the error between desired trajectory,  $q_d$  and the actual trajectory  $q$ .

The control torque for NAC can be calculated as:

$$\tau = \hat{f}(x) + K_V r - v(t) \quad (6.34)$$

where,  $K_V$  is a constant gain matrix.  $\hat{f}(x)$  is an approximation of Eq. 6.33. The  $\hat{f}(x)$  is estimated by NN as:

$$\hat{f}(x) = W^t \times \rho(V^T * x) \quad (6.35)$$

where,  $x$  is the input vector to the NN and is constructed as  $x = [e^T \ \dot{e}^T \ q_d^T \ \dot{q}_d^T \ \ddot{q}_d^T]$ , while  $V$  and  $W$  are weights associated with the NN input and output layers, and  $\rho$  is the activation function. Note that in the NAC, we use a ‘shallow’ NN with a single hidden layer, and that specific weight update laws detailed in [45] guarantee the system asymptotic stability in the Lyapunov sense.

From Eq. 6.29, the effect of gravitational torque at the base is required for joint torque estimation. To extract this information from our NAC,  $f(x)$  was divided into four NNs whose summation would reconstruct  $f(x)$  as follows:

$$f(x) = \Omega_\Lambda + \Omega_C + \Omega_\Psi + \Omega_\eta \quad (6.36)$$

where,  $\Omega_\Lambda$ ,  $\Omega_C$ ,  $\Omega_\Psi$  and  $\Omega_\eta$  are output of NN estimating  $\Lambda(q)(\ddot{q}_d + \nu\dot{e})$ ,  $C(q, \dot{q})(\dot{q}_d + \nu e)$ ,  $\Psi(q)$  and  $\eta(\dot{q})$  respectively. The output of the NNs can be described with equations:

$$\Omega_\Lambda = \hat{W}_\Lambda^T \times \rho(\hat{V}_\Lambda^T * \sigma_\Lambda) \quad (6.37)$$

$$\Omega_C = \hat{W}_C^T \times \rho(\hat{V}_C^T * \sigma_C) \quad (6.38)$$

$$\Omega_\Psi = \hat{W}_\Psi^T \times \rho(\hat{V}_\Psi^T * \sigma_\Psi) \quad (6.39)$$

$$\Omega_\eta = \hat{W}_\eta^T \times \rho(\hat{V}_\eta^T * \sigma_\eta) \quad (6.40)$$

where,  $\hat{W}$  and  $\hat{V}$  are estimated input and output weights for their respective NNs.  $\rho$  is the sigmoid activation function in the hidden layer. Since each dynamic term depends on a specific set of input variables, each of these NNs are provided with separate input basis vectors.  $\sigma$  are the basis vectors for these NNs and can be defined as:

$$\sigma_{\Lambda} = (\ddot{q}_d + \nu \dot{e}) \otimes ([q \sin(q) \cos(q)]) \quad (6.41)$$

$$\sigma_C = (\dot{q}_d + \nu e) \otimes ([q \dot{q} \sin(q) \cos(q)]) \quad (6.42)$$

$$\sigma_{\Psi} = ([q \sin(q) \cos(q) \sin(q) \cos(q)]) \quad (6.43)$$

$$\sigma_{\eta} = ([\dot{q} \operatorname{sgn}(\dot{q})]) \quad (6.44)$$

where,  $\otimes$  is the Kronecker product. Since it is well-known that gravity term depends on sines and cosines of the joint angles, it has been observed that providing it in the basis vector improves the performance of the NAC [16] [17].

Finally, after extracting the  $\Psi(q)$  term from the NAC, its effect at the base will be calculated to derive the gravity term  $\xi_{gravity_0}$ . This task is accomplished by the Base Force Estimator module of the controller.

### 6.2.2.3 Base Forces Estimator (BFE)

The base force estimator (BFE) portion of our controller is responsible for translating the joint torques due to gravity to wrenches at the base of the manipulator. By using the robot Jacobian expressed in the base frame,  $J_0^T$ , we use the relationship in Eq. 6.45 to estimate  $\hat{\xi}_{gravity_0}$  from  $\hat{\Psi}(q)$ , which was estimated by NAC:

$$\hat{\Psi}(q) = J_0^T * \hat{\xi}_{gravity_0}. \quad (6.45)$$

From Eq. 6.29, only  $\alpha$  remains to be calculated. This can be achieved through two methods - by solving the Eq. 6.30, which depend on the DH-parameters of the robot or by constructing another NN to learn the kinematic dependencies. In this paper, we use a NN to estimate  $\alpha$  off-line, by constructing an input basis vector as described in Eq. 6.46. The NN is defined in Eq. 6.47, where  $W_{BFE}^T$  and  $V_{BFE}^T$  are the learned input and output weights and  $\rho_{BFE}$  is the sigmoid activation function of the hidden layer.

$$X_{BFE} = [(\xi_{total_0} - \hat{\xi}_{gravity_0}) \quad q] \quad (6.46)$$

$$\hat{\alpha}(q) = W_{BFE}^T \times \rho_{BFE}(V_{BFE}^T X_{BFE}) \quad (6.47)$$

Finally, after  $\alpha$  is estimated to be  $\hat{\alpha}$ , we use the BFTS data and  $\hat{\xi}_{gravity_0}$  from NAC to estimate manipulator joint torques,  $\hat{\tau}_d$ , as follows:

$$\hat{\tau}_d = \hat{\alpha}(\mathbf{q}) \left[ \xi_{total_0} - \hat{\xi}_{gravity_0} \right]. \quad (6.48)$$

### 6.2.3 Experimental Implementation

In order to validate the on-line estimation algorithm described in the previous section, we configured an experimental testbed consisting of a custom 6-DOF arm from Re<sup>2</sup>, Inc., USA. The robot arm contains carbon-fiber, light-weight links and high torque back-drivable motors from HEBI Robotics, USA. As illustrated in Fig. 6.12, the arm is placed on a base force torque sensor, a Delta model from ATI, USA.

A single board computer, Blackbird from Versallogic, USA, running the Robot Operating System (ROS) was used to network the sensors and actuators and implement our controller in a real-time loop. A real-time patch was applied on the Ubuntu OS running our controller, and the algorithm was executed at 200Hz in real-time. The HEBI X-series motors communicate over Ethernet and are capable of taking torque command as input and provide current position, velocity and motor torque as feedback. The BFTS is EtherCAT enabled which ensures real-time capability of communication.

The desired trajectories for NAC were generated through sine and cosine functions of varying frequencies between  $\pi/8$  and  $\pi/32$  and amplitudes between  $\pm\pi/4$ . The joints were given the trajectories with phase differences of  $\pm\pi/2$  between each joint such that the combined trajectory generated would cover a large workspace. Since the desired trajectories are trigonometric functions, their velocity and acceleration can also be generated by using their derivative functions. As mentioned by Morel et al. [123], to reduce the effect of non-

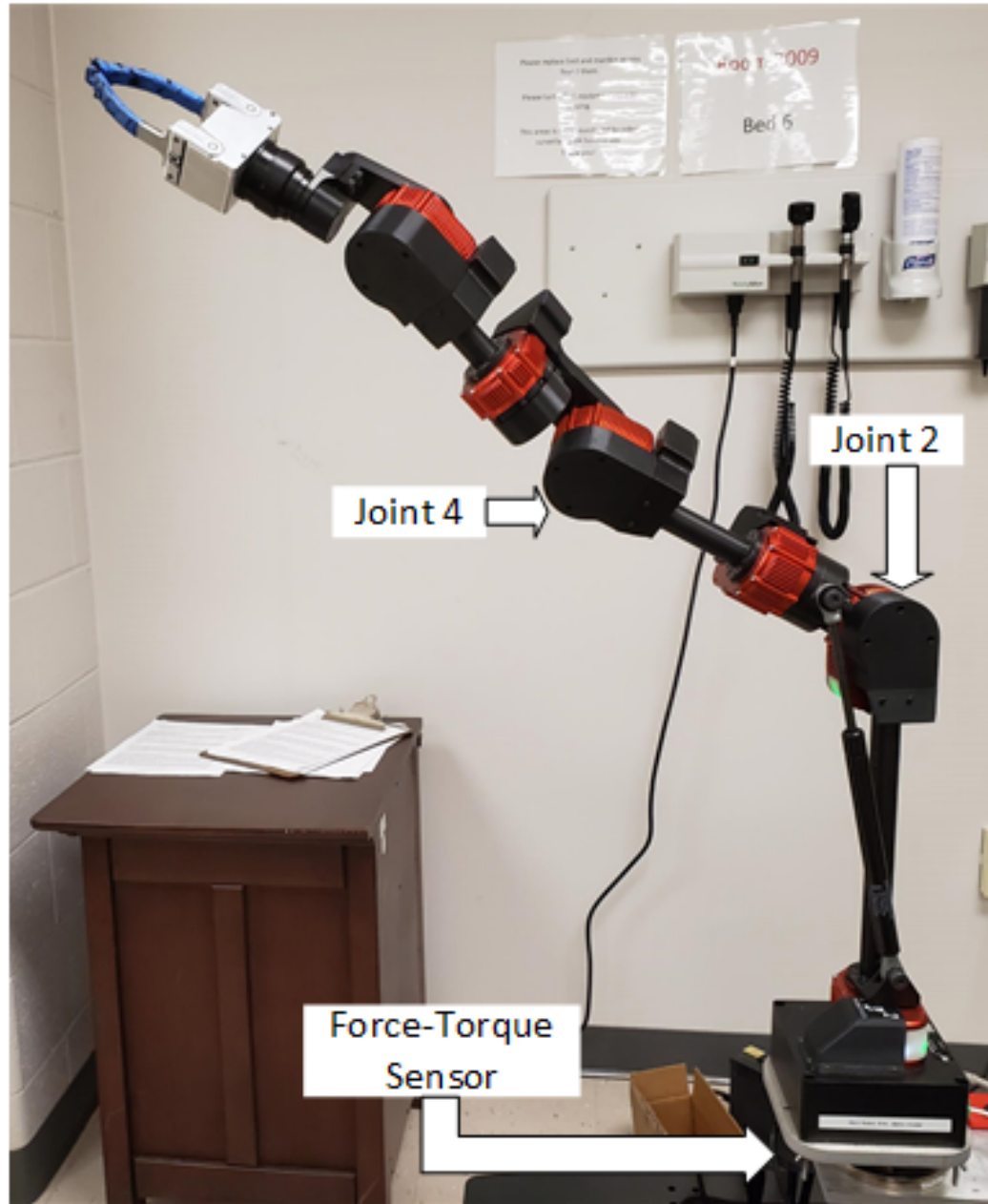


Figure 6.12: 6-DOF robot manipulator placed on a base force-torque sensor. [7]

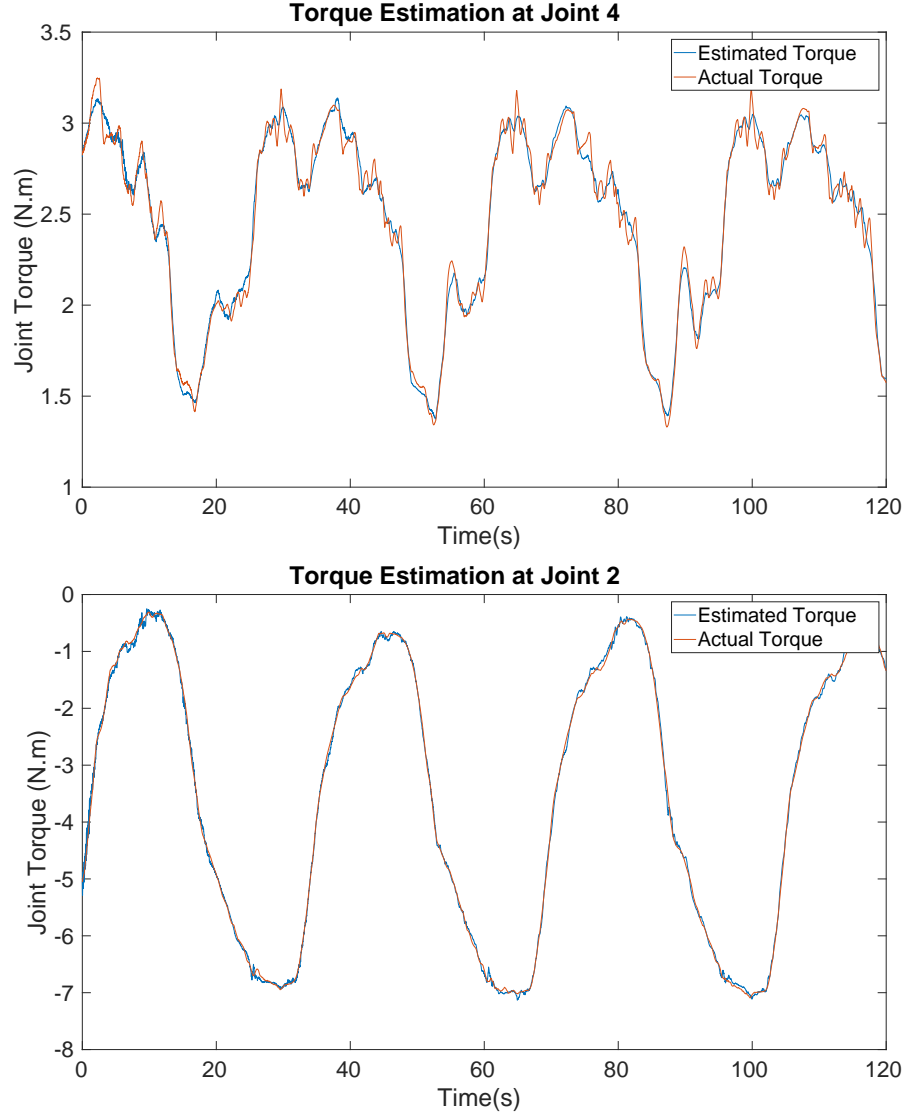


Figure 6.13: Actual vs Estimated torque value at Joint 4 (top), and Joint 2 (bottom). [7]

linear system dynamics such as Coriolis and inertia terms at the BFTS, the velocity of the joints was lowered to  $\pi/16$  rad/sec for each joint.

Four constituting NNs were implemented in ROS, as per Eq. 6.36. Each NNs consists of one input layer, one output layer and one hidden layer. The size of the input layer depends on the length of the basis vector associated with the network, as mentioned in Eq. 6.41, 6.42, 6.43 and 6.44. The hidden layer in all the NNs consisted of 20 neurons and the output layer's size depends on the number of joints in the robot, which is 6 in this case. Separate basis functions were created as inputs to the NN as mentioned in Section

6.2.2.2 and visualized in Fig. 6.11.

As the trajectories are executed, using the Jacobian at the base frame, the wrench at BFTS due to gravitational torque was calculated and subtracted from the BFTS data, as mentioned in Eq. 6.29. After the first run, the joint torque data and the resultant BFTS data were used to train the BFE NN. This process helps learn the  $\alpha$  term. A MATLAB neural network toolbox was used with Levenberg-Marquardt algorithm to accomplish this task, off-line. After this initial learning process was completed, the resulting  $\alpha$  term can be subsequently used in the on-line controller. Note that the  $\alpha$  term can also be derived formally from the robot DH parameters.

The on-line controller was run at 200 Hz, and used to estimate the joint torques, while comparing them with actual sensed values. There were four experiment runs conducted and the corresponding data was recorded and analyzed. In the next section, we discuss our results and the performance of the algorithm.

#### 6.2.4 Results and Discussion

TABLE 6.2

Experimental Results

	MSE Data Joint Torque Estimation					
	Joint1	Joint2	Joint3	Joint4	Joint5	Joint6
Run1	0.0030	0.0048	0.0044	0.0033	0.0035	0.0017
Run2	0.0028	0.0170	0.0055	0.0044	0.0036	0.0027
Run3	0.0034	0.0160	0.0056	0.0056	0.0038	0.0029
Run4	0.0030	0.0189	0.0060	0.0063	0.0037	0.0034

Fig. 6.13 depicts the estimated and measured torques at joints 2 and 4 of the robot, respectively. These joints were selected because they experience the largest loading effects during arm movements. Even though the commanded reference trajectories are sine waves in both cases, the joints have to exert considerably non-sinusoidal torques in order to track these trajectories. Furthermore, tracking errors converge close to zero as shown in Fig.

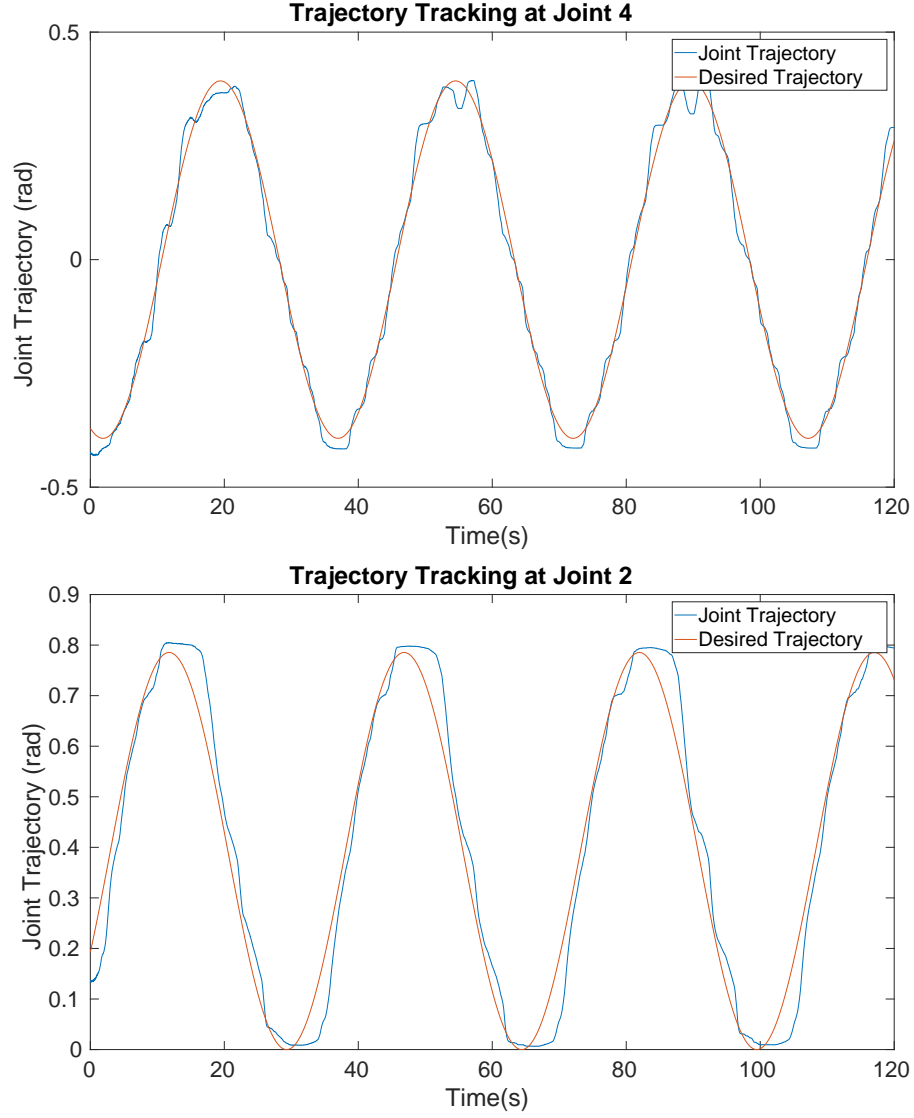


Figure 6.14: Trajectory Tracking at Joint 4 (top) and Joint 2 (bottom). [7]

6.14.

Furthermore, the joint torque estimation error was computed along entire trajectories using the mean-square-metric defined as:

$$MSE_{joint_i} = \left( \frac{1}{N_{data.rows}} \right) \sum_{data.rows} ((\tau_{d_i} - \hat{\tau}_{d_i})^2) \quad (6.49)$$

The results containing MSE computations are tabulated in Table 6.2. Results show that the estimation errors are consistently below 0.1 Nm for torque ranges between 3 and 7 Nm. After comparing the estimated joint torque and the actual values, we can conclude



that the estimation closely followed the actual data as seen in Fig. 6.13. Similarly, in Table 6.3, we see that the trajectory tracking errors are below 0.003 rad, which we can also visualize in Fig. 6.14.

TABLE 6.3

Experimental Results

	MSE Data Trajectory Estimation					
	Joint1	Joint2	Joint3	Joint4	Joint5	Joint6
Run1	0.0001	0.0036	0.0013	0.0007	0.0004	0.0002
Run2	0.00009	0.0035	0.0013	0.0007	0.0004	0.0002
Run3	0.0001	0.0036	0.0013	0.0008	0.0004	0.0003
Run4	0.0001	0.0036	0.0013	0.0007	0.0004	0.0002

### 6.2.5 Conclusions

In the presented study, we proposed a novel, Neuro-Adaptive pHRI controller for robot manipulators, that utilizes a force-torque sensor placed at its base. The main advantage of this scheme, when compared to traditional force measurements methods, is that it can estimate external interaction forces everywhere along the robot chain. The controller arises naturally from the NAC framework via an additional joint effort estimator implemented using four neural nets. The controller was implemented in ROS and ran on a real-time OS, to command a custom 6-DOF light-weight arm. The advantage of our proposed controller is that it does not need a dynamic model of the robot. However, the controller does need an initial estimate of the robot kinematics and the robot Jacobian. Experimental results show that joint torque estimation errors smaller than 5% of the torque magnitudes can be expected.

In the future, we will combine the proposed neuro-adaptive controller with other adaptive admittance controllers, and with robotic skin sensors to accomplish high performance pHRI with non-collaborative robots and with mobile manipulators.

## CHAPTER 7

### ARNA USER STUDIES

User studies were conducted on ARNA robot to better validate the system and its efficacy in a healthcare environment. The studies were conducted with nursing students to gather quantitative as well as qualitative data for analysis. The studies were designed such that the functionality of the robot is tested for both sitter and walker scenario as described in chapter 3. This chapter describes the user-testing process and analyzes the gathered data.

#### 7.1 Introduction

As new technological advancements are being made in nursing and healthcare domain, their acceptance and usability need to be analyzed. Many studies have been conducted in the past to test the viability of new technologies in the field of healthcare. Davis has proposed in these studies on the method of forming questions that can be grouped together into two broad categories, namely - perceived ease of use and perceived usefulness [126]. Having such kind of a questionnaire is important as it provides comprehensive information about how a user sees the new technology.

Based on the questionnaire and the logical grouping for the questions, it was later proposed that a technology acceptance model can be formed [127] [8]. A technology acceptance model is required to accurately predict the intent to use a technology in future based on the perceived usefulness (PU) and perceived ease of use (PEOU) of any technology by the users. The intent to use metric is important as new technology may seem to solve a problem but if the user finds it difficult to use, the technology may not be accepted in the proposed field.

Figure 7.1 shows the technology acceptance model (TAM) as proposed by Davis, which is based on the psychological research by Fishbein and Ajzen [128]. The technology acceptance model proposes hypotheses based on figure 7.1 and then using statistical analysis on the response to the questionnaires, these hypotheses are verified. The hypotheses based on the figure are as follows [8]:

- Attitude towards using a technology will have positive effect on actual system usage.
- PEOU will have positive effect on PU.
- PEOU will have positive effect on Attitude.
- PU will have positive effect on Attitude.
- The technological system will affect the PU and PEOU.
- PU, PEOU and system will not affect actual usage directly.
- System will not affect Attitude towards technology.

A questionnaire can be grouped into these five categories - System, PU, PEOU, Attitude and intent to Use. In the case of experiments conducted with ARNA, the first hypothesis is tested. This was done to establish that the technology fits the TAM. After the data was gathered from the questionnaire, the questions and responses were grouped into two categories, PU and PEOU, and then statistical analysis was conducted to test the hypothesis.

The questions in Table B.1 were grouped by taking items [1, 2, 5, 8, 11, 12, 15, 16, 17, 19] as indicators for PU and items [3, 4, 6, 7, 9, 10, 13, 14, 18, 20] as indicators for PEOU. Similarly, in Table B.2, questions 1 to 14 were considered as part of PU and questions 15 to 28 were grouped to form PEOU indicators. These groupings were made by following the metrics laid out by Davis in his research paper [126].

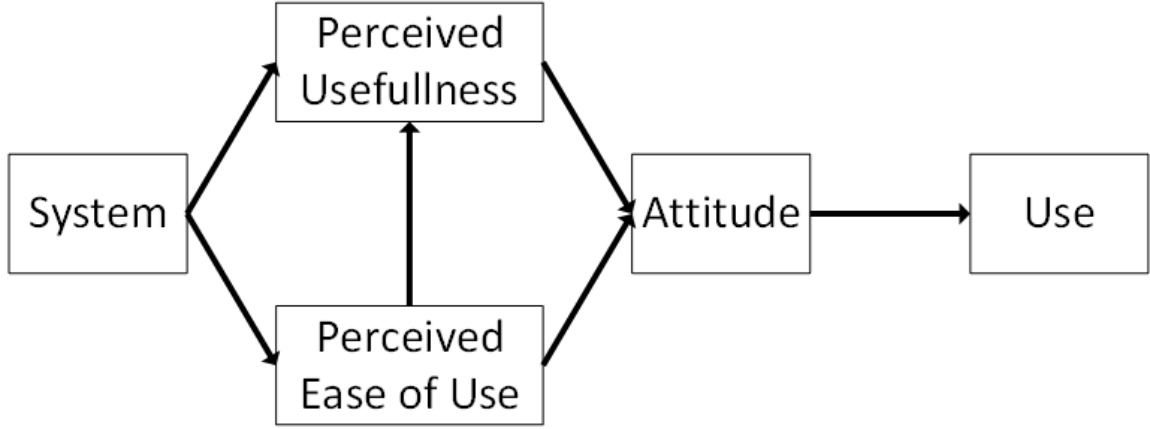


Figure 7.1: Technology Acceptance Model (TAM) proposed by Davis [8]

## 7.2 Experiment Setup

The facility at School of Nursing (SON), University of Louisville, KY was used for testing the ARNA robot. This helped in simulating a hospital environment for the experiments, as seen in fig 7.2. The experiments were conducted in two phases. Each phase had twelve groups of nursing students, each group containing two people. During the experiments, one of the students played the role of a patient and the other played the role of a nurse. Upon the completion of one set of experiment, the roles were switched, thus providing the role of nurse and patient to both the participants. Each experiment phase lasted for six days, where experiments were conducted with two groups per day. During the course of experiments, data were recorded using ROS bag files and after the end of each run, a set of questionnaires were presented to the subjects. Subsequent subsections describe the experiment protocols in detail.

### 7.2.1 Sitter Scenario

The sitter scenario experiment was designed, keeping in mind the task outlines of chapter 3. A tablet was used as the HMI device through which the subject, playing the role of a patient, had to control the robot to accomplish tasks. during these experiments the second subject was not required to perform any task. The subject was situated inside the

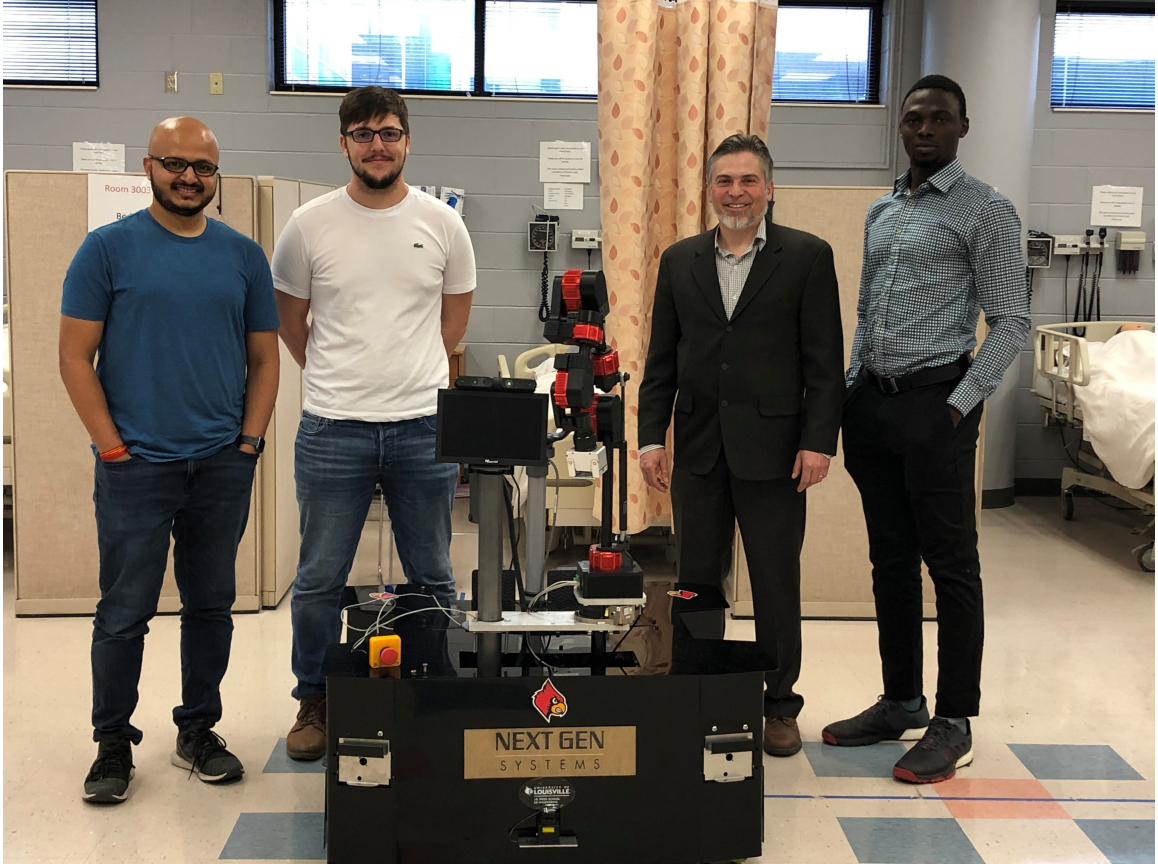


Figure 7.2: ARNA at School of Nursing, University of Louisville, KY

simulated hospital room on the bed, which can be seen in the background of fig 7.2. From that static location, the subject was required to control the ARNA robot.

The task that the subject had to complete is illustrated in fig 7.3. The task is divided into 4 parts. First the subject had to drive the robot to a table situated at about 4.5 meters from the simulated room. During this process, the feedback from the RGB-D camera was used to assist with the driving of the robot. Then the subject had to control the 6-DOF arm on the robot to grab a box placed on the table. This box contained vital sign measurement equipment such as thermometer and pulse oximeter. The tablet provided the necessary interfaces for the control of the arm and the base of the robot. Once the box was captured, the subject was required to rotate the robot and drive it back to the room. Once the robot was near the subject, the box had to be retrieved and the equipment inside the box were used to take measurements of the subject. Once, this step was completed, the

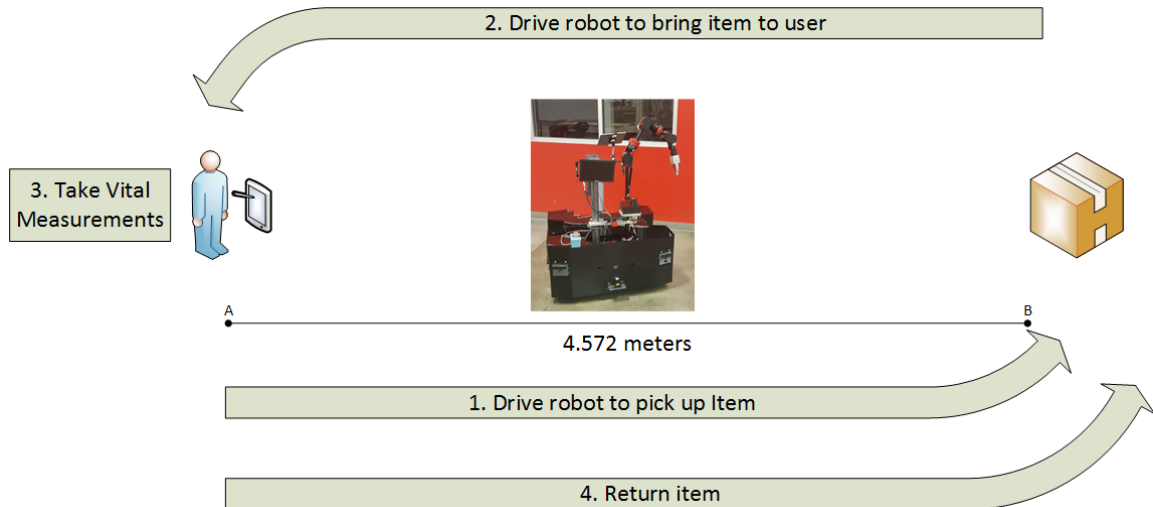


Figure 7.3: Sitter Task Experiment

box was re-attached to the arm of the robot and was driven back to the table to place it at its original position.

Each subject had to perform the experiment thrice. After each run the subjects were asked to fill out a questionnaire describing their experience. The questionnaire in table B.4 was used during the experiments and is detailed in section 7.2.3. Along with the questionnaire, data related to robot operation was also collected, such as completion time and pick time. The completion time represents the time taken by the subject to finish each experiment. Within the experiment, picking up the object while remotely tele-operating it through a tablet takes significant effort and expertise. Therefore, the time taken by each subject to capture the box was also collected as data.

### 7.2.2 Walker Scenario

In the walker scenario, a rectangular area was marked on the floor, as seen in fig 7.4, which the subject had to follow while walking with the robot. This provided a standard metric to compare the performance of the subjects as well as the robot. During the course of experiments, one of the subjects played the role of a patient and the other played the role of a nurse. The patient's task was to walk with the robot while following the marked lines on the floor while the nurse's task was to hold the emergency stop button and walk

alongside the patient as a supervisory role. This provided additional safety measure as the subject playing the nurse can stop the robot if necessary. While walking, the robot is also holding on to an IV pole with the arm and its position was maintained constantly by the robot's side during the entire experiment run.

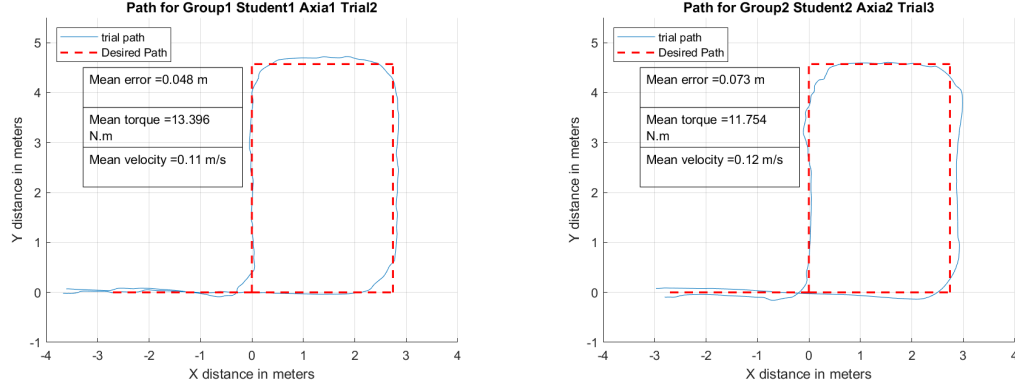


Figure 7.4: Walker Task Experiment

During the experiment, the user interaction forces and torques were sensed by the force-torque sensor attached to the handlebar of the robot. In this particular case, the torques were used as inputs for driving the robot. This provided a scooter-like interface where in order to move forward, the handlebar was to be twisted in the forward direction. Three controllers, named axia1, axia2 and axia3, with varying stiffness were used where the subject has to perform three trials per controller to complete the entire experiment run. Therefore, each experiment set consisted of 9 experiment runs per subject. The entire sequence of the controller used was randomized to exclude any inherent biases. After each experiment, the subject playing the role of patient was asked to fill out the questionnaire in table B.1. Whereas the subject playing the nurse had to fill out the questionnaire in table B.2 after the entire experiment run was complete. Then the roles are switched, and the entire process is repeated.

During the course of experiments, along with the questionnaire, data related to the speed of the walker, course completion time and user interaction torques were collected. These metrics provide an insight on the way the robot was operated. After the completion

of Phase 1 of testing, during Phase 2, only axia3 setting was used for testing purposes.

### 7.2.3 Questionnaire

The questionnaires were designed in order to quantify the level of comfort in using the robot and its efficacy according to the subjects. Table B.1 enumerates the questions that were asked to the subjects after the end of each experiment run. These questions were rated from 1 to 5 with 1 being the lowest score to the likeliness of the question. From example, in question 2 the responsiveness of the robot is asked to which a score of 1 would mean least responsive and a score of 5 would mean very responsive. Each subject had to fill this questionnaire nine times corresponding to nine experiment runs.

The questionnaire in Table B.2 was asked, to the subject playing the nurse, once after the end of the nine experiments. These questions seek to understand the usefulness and the ease of use of the ARNA robot. These questions were rated from 1 to 7, with 1 being that the user disagrees with the assertion. For example, in question 3 the assertion is that the robot improves the performance from nurses' perspective and a rating of 1 would mean that the subject disagrees with the statement. In the questionnaire, questions 1 through 13 inquire about the usefulness of the robot and questions 14 to 28 pertains to the ease of use of the robot.

Table B.3 represents the questionnaire that was asked of all the participants in the first phase of the user testing only. But this questionnaire was removed from phase two of the testing due to the redundancy in questions. Instead the questionnaire in Table B.4 were asked after the completion of the sitter tasks.

The questions represent two contexts - Ease of Use and Usefulness. The response to the questions about usefulness of the robot provides and insight to whether the robot provides value in the current scenario. Where as in the case of ease of use, the learning or training required to operate the robot is understood. Even if the robot provides value, it should be easy to use in order to increase efficiency. Therefore, both these areas of questions are necessary to explore.



After completing each trial of the sitter experiment, the user was presented with questions in Table B.3. These questions were rated between 1 and 5. For questions 1 to 29, the rating relate which word the subject would most associate the robot with. For example, in question 26, a rating of 1 would mean the robot is efficient whereas a score of 5 would mean the robot is inefficient. From question 30 to 41, a rating of 1 means the assertion is unlikely and 5 means that the assertion is very likely.

### 7.3 Results and Discussion

In case of sitter scenario, the average completion time for all the subjects was 209.52 seconds with a standard deviation of 86.84 seconds. Considering only the object pick time, the average was 49,66 seconds with a standard deviation of 24.64 seconds. Figure 7.5 and figure 7.6 illustrate the time taken by the subjects for each trial.

It was observed that the completion time as well as the object pick time was better for subjects who are accustomed to gaming controllers. For example, trial 45 and 46 have very low timings from the rest of the group.

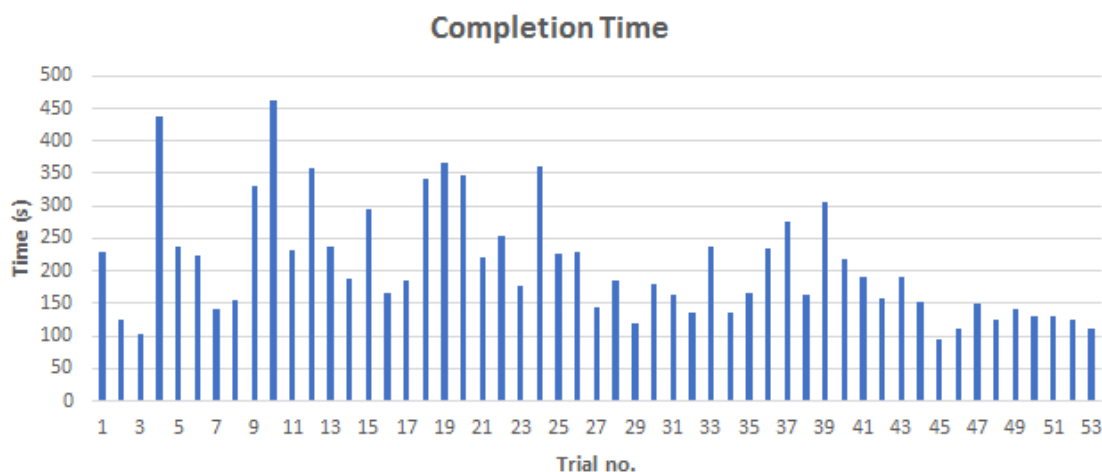


Figure 7.5: Sitter Average Completion Time

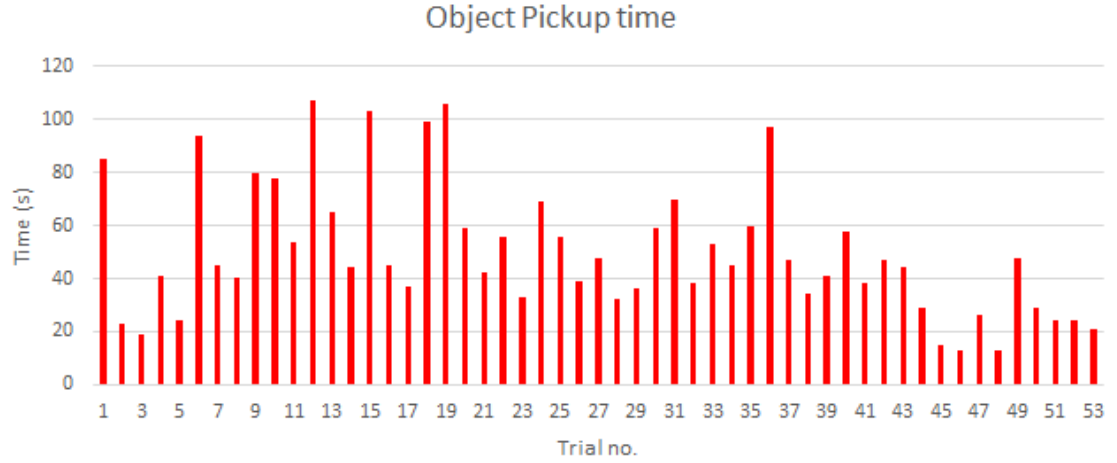


Figure 7.6: Sitter Average Pick Time

Figures 7.7, 7.8 and 7.9 illustrate the results of walker scenario experiments. The results of the Phase 1 testing are tabulated as below:

TABLE 7.1

Phase 1 Walker Experiment Result

	Controller Stiffness Settings		
	Axia1	Axia2	Axia3
Average Completion Time in s	101.73	92.09	91.36
Average Trajectory Error in m	0.57	0.47	0.43
Average Velocity in m/s	0.21	0.22	0.225

With lower trajectory error and faster finishing time, the controller setting Axia3 was selected for Phase 2 of the experiment where tests were conducted with one setting only. Figures 7.7, 7.8 and 7.9 represent the average completion time, average velocity and average error during the experiments respectively.

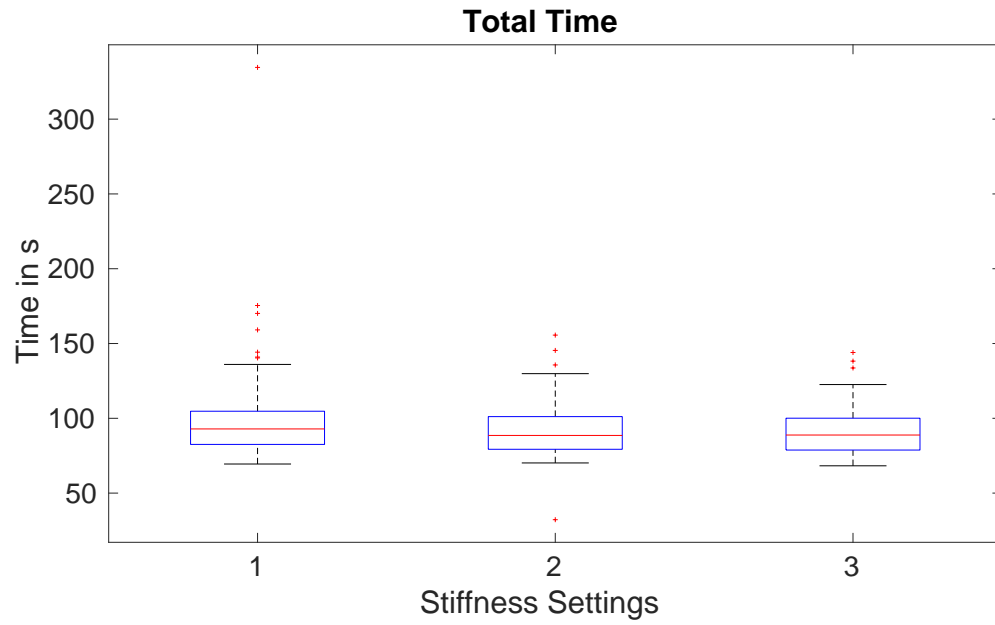


Figure 7.7: Walker Average Completion Time

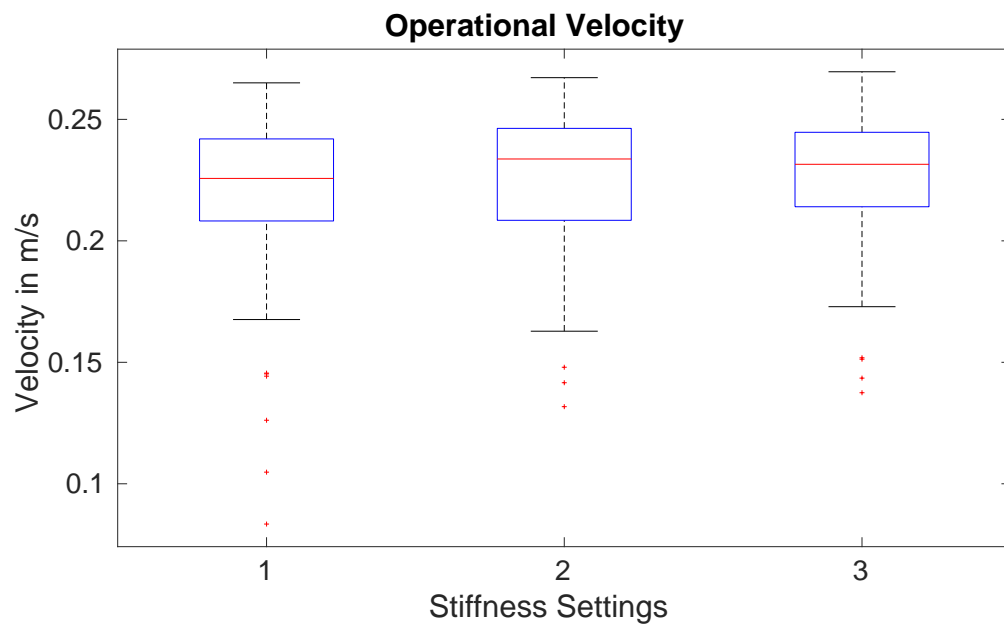


Figure 7.8: Walker Average Speed

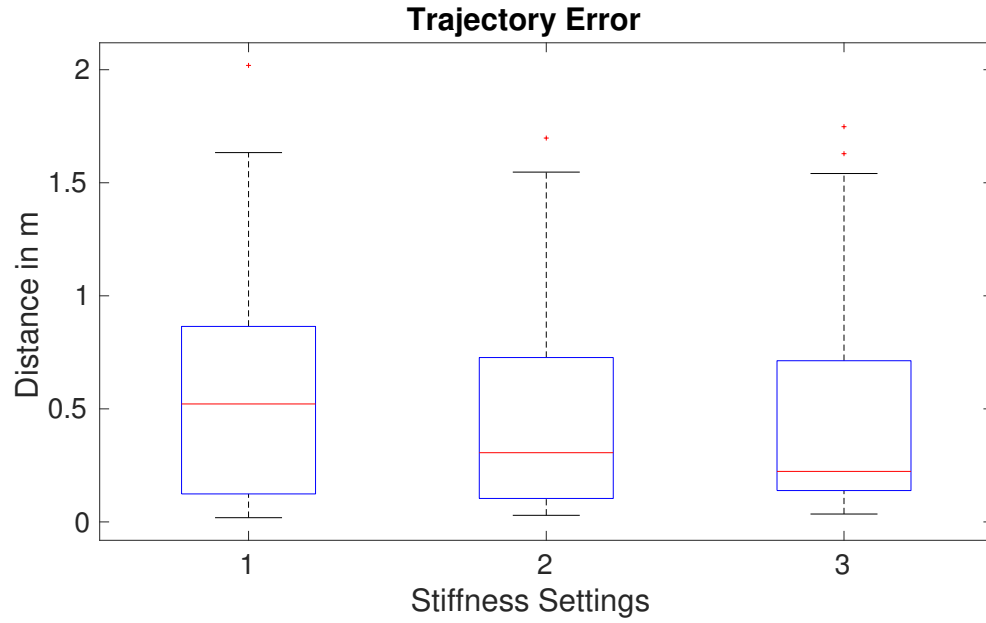


Figure 7.9: Walker Error in Trajectory Following

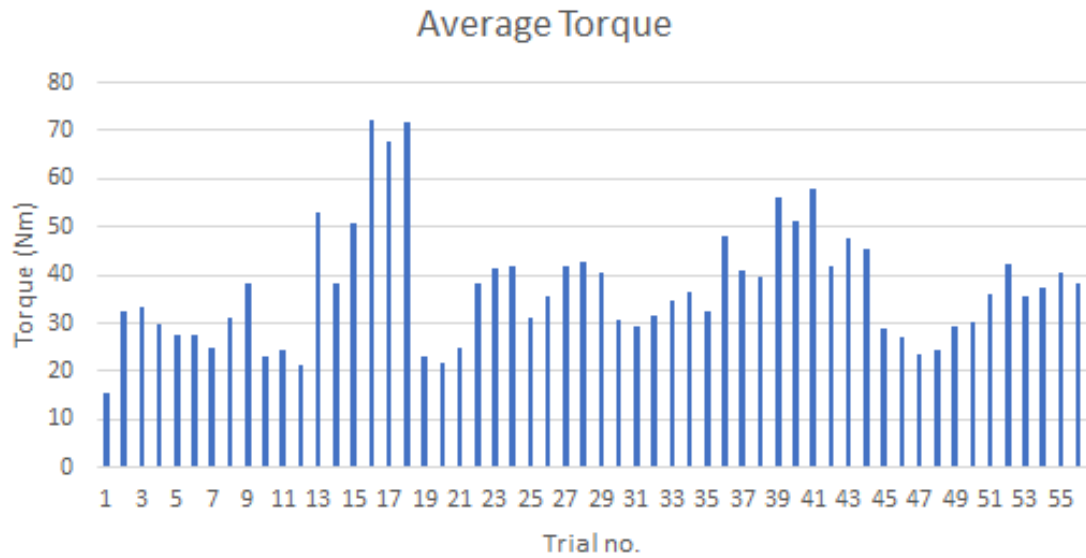


Figure 7.10: Walker Average Torque

In the second phase of testing, the average completion time with Axia3 setting was found to be 87.91 seconds with a standard deviation of 19.99 seconds. The mean torque applied during the experiment was 37.23Nm with a standard deviation of 12.11Nm. And

TABLE 7.2

## Phase 1 Walker Questionnaire Result

Metrics	Mean	Std. Dev	Min	Max
PU Axia1	3.00	1.36	1	5
PEOU Axia1	3.28	1.37	1	5
PU Axia2	3.06	1.30	1	5
PEOU Axia2	3.33	1.38	1	5
PU Axia3	3.16	1.25	1	5
PEOU Axia3	3.42	1.35	1	5

TABLE 7.3

## Phase 1 Walker Regression Result

Independent Variable	Dependent Variable	SumSq	DF	MeanSq	F	pValue	$R^2$
PEOU Axia1	PU Axia1	5.47	1	5.47	27.13	2.37 e-06	.308
PEOU Axia2	PU Axia2	2.72	1	2.72	12.55	0.0008	.171
PEOU Axia3	PU Axia3	3.86	1	3.86	16.13	0.0001	.209

the average velocity of the robot during the experiments was 0.11 m/s with a standard deviation of 0.02 m/s. Since the safety limits were added to the robot's response to user interaction forces, it can be observed that even when high torques were applied the velocity of the robot didn't change as much. This can also be seen in the low standard deviation of the velocity of the robot.

In Phase 1, the EOU and PU data from user survey were analyzed to test the hypothesis that PEOU affects PU. Table 7.2 tabulates the average, standard deviation, min and max score that PU and PEOU received from the users. The average scores for each metric was calculated by taking the average score of all the responses for each question in the metric group. The analysis was done separately for each controller settings. It can be observed that each metric has an average score of 3 or higher out of 5. For further analysis of how PEOU affects PU, regression analysis was conducted.

ANOVA was used to analyze the effect of PEOU on PU. The result of the regression analysis is documented in Table 7.3. The pValue for all the relations are  $\leq 0.0001$  which

points to the fact that there is a dependency between these two factors irrespective of the controller settings. The  $R^2$  value ranged from .308 to .171 which means that PEOU effect PU by 30.8% to 17.1% depending on the controller settings. This substantiates our initial hypothesis that PEOU affect PU in case of walker scenario.

## CHAPTER 8

### CONCLUSION AND FUTURE WORK

#### 8.1 Conclusion

The research work presented here is intended to better enable the ARNA robot's functions in a hospital environment. The ARNA robot was designed and built to assist nurses with their routine tasks that can be automated. Two broad areas of tasks were also identified for the robot, namely — Sitter task and Walker task. The tasks would entail the ARNA robot to be in close proximity of human users. During these tasks the HRI can be physical or non-physical based on the nature of the intended tasks.

Sensors play a vital role of sensing the environment around the robot. It also senses the user forces during pHRI. Artificial skin, with sensors embedded in it, are used to sensorize a robot part that is supposed to sense human forces or collisions. A simulation module was built for such a sensor package which includes the sensor as well as the artificial skin in which it is embedded in. The simulation model predicts the sensor behavior by estimating the level of deformation in the skin and the dissipation of forces due to the physical properties of the skin. Therefore, for a certain amount of force, the sensor might experience a diminished force due to the dissipation in the skin. This is particularly important as the controllers need to have the accurate force information for its operation. The model was also validated through comparisons with actual experiment data. It was shown that the model can be used to predict the sensor outputs under various conditions.

An HMI for ARNA related tasks was investigated. The performance of a tablet-based HMI was tested for sitter scenario. The autonomy level of traded control was decided upon for operation of ARNA in this scenario. It was illustrated that using a traded control scheme, a uniform performance can be achieved by both trained as well as untrained users. The idea

being that the HMI should be easily operable by a patient, nurse or system administrator. Experiments were conducted using trained and expert users and the completion rate of tasks for both groups was shown to be similar. The proposed HMI was tested using a 2-arm robot called Baxter from Rethink Robotics. The fetch tasks were used for the purpose of testing the system. The robot was commanded to fetch random items using the HMI and robot was able to complete medium-level tasks such as trajectory planning and execution, object localization, and grasp planning.

The tablet-based HMI was developed to be capable of taking voice as well as onscreen button-based commands. The voice input also allows the HMI to facilitate conversation with the robot, which is also one of the functionalities in the sitter scenario. A context-based conversation engine was implemented to enable a meaningful dialog. Experiments were conducted where the subjects had to talk to the robot using this interface. A questionnaire was prepared to evaluate the user experience while using this system. The system was shown to be user-friendly and was rated equally by both expert as well as novice users.

A neuroadaptive controller with HIE was validated on a PR2 robot. The controller contained two loops to estimate user intent and the system dynamics. It was observed that the controller was able to approximate the human operator's intended trajectory online without any prior knowledge of the system. It was demonstrated that the controller's performance was similar to that of traditional methods but without the need of training them. The effect of neural network size on the performance of the robot was also investigated by conducting preliminary research studies.

A novel model free controller was designed for full body collision detection using a BFTS attached to the base of a 6-DOF manipulator. The BAPI controller is model free and is able to detect collisions throughout the robot body during normal operation. The forces experienced at the base were decoupled by identifying the forces that are expected to be sensed at the base due to the movements and the forces sensed due to collisions. The controller was experimentally validated with an error bound for forces of  $\pm 2\text{N}$  and torque of  $\pm 1\text{Nm}$  which is less than 50N of robot collision which a human can tolerate.



A holistic DOLA framework for the robot operation was proposed. The DOLA framework is designed to adapt to the different levels of autonomy during HRI. The ten levels of autonomy define the roles and responsibilities of the human user, which decreases as the system becomes autonomous. In a hospital environment, the flexibility of choosing an autonomy level allows the robot to be used by different patients and in accordance to their capabilities. The Observer block of the DOLA framework was defined to continuously monitor the patient's intent during operation. Using a HIE module, the Observer block enables the robot to adapt to the user's mannerisms. The proposed framework utilizes neural network based controllers which makes it model-free and allows it to be implemented on any service robot.

## 8.2 Future Work

The DOLA framework is planned to be implemented and investigated against currently existing frameworks. The experiments would cover all the scenarios for ARNA robot in the health-care facility. Human subjects would be able to use this framework to provide feedback on the usability of such a system in assisting the nurses.

The future work involves validating NAC with HIE on the ARNA robot. The controller performance will be investigated in a hospital environment while performing walker scenario based tasks with human operators. The controller can be further improved to adapt to user intentions using different neural network sizes in the NAC and HIE.

The BAPI controller is to be implemented on the 6-DOF arm on the robot which would be tasked with holding the essential equipment during walker scenario and pick and place tasks in sitter scenario. The BAPI controller will be able to detect the forces acting on the arm and stabilize its performance during the tasks.

## REFERENCES

- [1] Ritvij Sahasrabuddhe, “Development, Testing and Characterization of Electronic Skins for Robots,” M.S. thesis, The University of Texas at Arlington, 2015.
- [2] Sumit Kumar Das, Joshua R Baptist, Ritvij Sahasrabuddhe, Woo H Lee, and Dan O Popa, “Package analysis of 3d-printed piezoresistive strain gauge sensors,” in *Sensors for Next-Generation Robotics III*. International Society for Optics and Photonics, 2016, vol. 9859, p. 985905.
- [3] Rachid Alami, Alin Albu-Schäffer, Antonio Bicchi, Rainer Bischoff, Raja Chatila, Alessandro De Luca, Agostino De Santis, Georges Giralt, Jérémie Guiochet, Gerd Hirzinger, et al., “Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges,” IEEE, 2006, pp. 1–16.
- [4] Sumit Kumar Das, Ankita Sahu, and Dan O Popa, “Mobile app for human-interaction with sitter robots,” in *Smart Biomedical and Physiological Sensor Technology XIV*. International Society for Optics and Photonics, 2017, vol. 10216, p. 102160D.
- [5] Sven Cremer, Sumit K Das, Indika B Wijayasinghe, Dan O Popa, and Frank L Lewis, “Model-free online neuroadaptive controller with intent estimation for physical human-robot interaction,” TRO Journal - Conditionally Accepted.
- [6] Sumit K Das, Indika B Wijayasinghe, Mohammad Nasser Saadatzi, and Dan O Popa, “Whole body human-robot collision detection using base-sensor neuroadaptive interaction,” in *Automation Science and Engineering (CASE), 2018 IEEE International Conference on*. IEEE, 2018.
- [7] Sumit K Das, Mohammad Nasser Saadatzi, Shamsudeen Abubakar, and Dan O Popa, “Joint torque estimation using base force-torque sensor to facilitate physical human-robot interaction (phri),” in *2019 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019.
- [8] Fred D Davis, “User acceptance of information technology: system characteristics, user perceptions and behavioral impacts,” *International journal of man-machine studies*, vol. 38, no. 3, pp. 475–487, 1993.
- [9] Robotic Industries Association et al., “Ansi/ria r15. 06: 2012 safety requirements for industrial robots and robot systems,” *Ann Arbor: Robotic Industries Association*, 2012.
- [10] I Iso, “Iso 10218-1: 2011: Robots and robotic devices—safety requirements for industrial robots—part 1: Robots,” *Geneva, Switzerland: International Organization for Standardization*, 2011.
- [11] Antonio Bicchi, Michael A Peshkin, and J Edward Colgate, “Safety for physical human–robot interaction,” in *Springer handbook of robotics*, pp. 1335–1348. Springer, 2008.
- [12] Ryan A Beasley, “Medical robots: current systems and research directions,” *Journal of Robotics*, vol. 2012, 2012.

- [13] Rainer Konietzschke, Tobias Ortmaier, Holger Weiss, Gerd Hirzinger, and Robert Engelke, "Manipulability and accuracy measures for a medical robot in minimally invasive surgery," in *On advances in robot kinematics*, pp. 191–198. Springer, 2004.
- [14] Russell H Taylor, Arianna Menciassi, Gabor Fichtinger, Paolo Fiorini, and Paolo Dario, "Medical robotics and computer-integrated surgery," in *Springer handbook of robotics*, pp. 1657–1684. Springer, 2016.
- [15] Allison M Okamura, Maja J Mataric, and Henrik I Christensen, "Medical and health-care robotics," *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 26–37, 2010.
- [16] Frank L Lewis, "Neural network control of robot manipulators," *IEEE Expert*, vol. 11, no. 3, pp. 64–75, 1996.
- [17] Frank L Lewis, Kai Liu, and Aydin Yesildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 703–715, 1995.
- [18] Ankur Kapoor, Anton Deguet, and Peter Kazanzides, "Software components and frameworks for medical robot control," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 3813–3818.
- [19] Yuan-Fang Wang, Darrin R Uecker, and Yulun Wang, "A new framework for vision-enabled and robotically assisted minimally invasive surgery," *Computerized Medical Imaging and Graphics*, vol. 22, no. 6, pp. 429–437, 1998.
- [20] Robin Murphy, Robin R Murphy, and Ronald C Arkin, *Introduction to AI robotics*, MIT press, 2000.
- [21] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng, "Ros: an open-source robot operating system," *ICRA workshop on open source software*, vol. 3, no. 3.2, pp. 5, 2009.
- [22] Dragoljub Surdilovic, Gerhard Schreck, and Uwe Schmidt, "Development of collaborative robots (cobots) for flexible human-integrated assembly automation," in *Robotics (ISR), 2010 41st international symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*. VDE, 2010, pp. 1–8.
- [23] Manuela Veloso, Joydeep Biswas, Brian Coltin, Stephanie Rosenthal, Tom Kollar, Cetin Mericli, Mehdi Samadi, Susana Brandao, and Rodrigo Ventura, "Cobots: Collaborative robots servicing multi-floor buildings," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5446–5447.
- [24] Rosemarijn Looije, Mark Neerincx, and Geert-Jan M Kruijff, "Affective collaborative robots for safety & crisis management in the field," *Intelligent human computer systems for crisis response and management (ISCRAM 2007), Delft, Netherlands*, vol. 1, 2007.
- [25] Harshal Maske, Milecia Matthews, Allan Axelrod, Hossein Mohamadipanah, Girish Chowdhary, Christopher Crick, and Prabhakar Pagilla, "Collaborative goal and policy learning from human operators of construction co-robots," in *Neural Information Processing Systems (NIPS)*, 2014.
- [26] Brenda E Booth BSN MSed, "Robotics in nursing," *Journal of Practical Nursing*, vol. 61, no. 4, pp. 12–13, 2011.

- [27] Pramila Rani, Nilanjan Sarkar, and Craig A Smith, “Affect-sensitive human-robot cooperation: Theory and experiments,” in *IEEE international conference on robotics and automation*. IEEE; 1999, 2003, vol. 2, pp. 2382–2387.
- [28] Lei Wu, Jiang Lu, Ting Zhang, and Jiaqi Gong, “Robot-assisted intelligent emergency system for individual elderly independent living,” in *Global Humanitarian Technology Conference (GHTC), 2016*. IEEE, 2016, pp. 628–633.
- [29] Bjoern Matthias, Soenke Kock, Henrik Jerregard, Mats Källman, and Ivan Lundberg, “Safety of collaborative industrial robots: Certification possibilities for a collaborative assembly robot concept,” in *Assembly and Manufacturing (ISAM), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 1–6.
- [30] Changliu Liu and Masayoshi Tomizuka, “Algorithmic safety measures for intelligent industrial co-robots,” in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3095–3102.
- [31] GS Virk, S Moon, and R Gelin, “Iso standards for service robots,” in *Advances In Mobile Robotics*, pp. 133–138. World Scientific, 2008.
- [32] Shujun Lu and Jae H Chung, “Collision detection enabled weighted path planning: a wrist and base force/torque sensors approach,” in *Advanced Robotics, 2005. ICAR’05. Proceedings., 12th International Conference on*. IEEE, 2005, pp. 165–170.
- [33] Shujun Lu, Jae Heon Chung, and Steven A Velinsky, “Human-robot collision detection and identification based on wrist and base force/torque sensors,” in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 3796–3801.
- [34] Yoji Yamada, Yasuhiro Hirasawa, Shengyang Huang, Yoji Umetani, and Kazutsugu Suita, “Human-robot contact in the safeguarding space,” *IEEE/ASME transactions on mechatronics*, vol. 2, no. 4, pp. 230–236, 1997.
- [35] Steve Charles, Hari Das, Timothy Ohm, Curtis Boswell, Guillermo Rodriguez, Robert Steele, and Dan Istrate, “Dexterity-enhanced telerobotic microsurgery,” in *1997 8th International Conference on Advanced Robotics. Proceedings. ICAR’97*. IEEE, 1997, pp. 5–10.
- [36] Hari Das, TIM Ohm, Curtis Boswell, ROB Steele, and Guillermo Rodriguez, “Robot-assisted microsurgery development at jpl,” *IN MEDICINE*, p. 85, 2001.
- [37] Federico Vicentini, Nicola Pedrocchi, Matteo Malosio, and Lorenzo Molinari Tosatti, “Safenet: A methodology for integrating general-purpose unsafe devices in safe-robot rehabilitation systems,” *Computer methods and programs in biomedicine*, vol. 116, no. 2, pp. 156–168, 2014.
- [38] Paolo Barattini, Claire Morand, and Neil M Robertson, “A proposed gesture set for the control of industrial collaborative robots,” in *RO-MAN, 2012 IEEE*. IEEE, 2012, pp. 132–137.
- [39] Juan C Yepes, Juan J Yepes, Jose R Martinez, and Vera Z Pérez, “Implementation of an android based teleoperation application for controlling a kuka-kr6 robot by using sensor fusion,” in *Health Care Exchanges (PAHCE), 2013 Pan American*. IEEE, 2013, pp. 1–5.

- [40] Rafael V Aroca, Antônio Péricles, BS de Oliveira, L Marcos, and G Gonçalves, “Towards smarter robots with smartphones,” in *5th workshop in applied robotics and automation, Robocontrol*. sn, 2012, pp. 1–6.
- [41] Tatiana Alexenko, Megan Biondo, Deya Banisakher, and Marjorie Skubic, “Android-based speech processing for eldercare robotics,” in *Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion*. ACM, 2013, pp. 87–88.
- [42] Kris Doelling, Jeongsik Shin, and Dan O Popa, “Service robotics for the home: a state of the art review,” in *Proceedings of the 7th International Conference on Pervasive Technologies Related to Assistive Environments*. ACM, 2014, p. 35.
- [43] Jenay M Beer, Arthur D Fisk, and Wendy A Rogers, “Toward a framework for levels of robot autonomy in human-robot interaction,” *Journal of Human-Robot Interaction*, vol. 3, no. 2, pp. 74–99, 2014.
- [44] Isura Ranatunga, Sven Cremer, Dan O. Popa, and Frank L. Lewis, “Intent aware adaptive admittance control for physical Human-Robot Interaction,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5635–5640.
- [45] Isura Ranatunga, Sven Cremer, Frank L. Lewis, and Dan O. Popa, “Neuroadaptive control for safe robots in human environments: A case study,” in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2015, pp. 322–327.
- [46] Isura Ranatunga, Frank L. Lewis, Dan O. Popa, and Shaikh M. Tousif, “Adaptive Admittance Control for Human-Robot Interaction Using Model Reference Design and Adaptive Inverse Filtering,” *IEEE Transactions on Control Systems Technology*, pp. 1–8, 2016.
- [47] Hui-Min Huang, Elena Messina, Robert Wade, Ralph English, Brian Novak, and James Albus, “Autonomy measures for robots,” in *ASME 2004 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers, 2004, pp. 1241–1247.
- [48] Sebastian Thrun, “Toward a framework for human-robot interaction,” *Human-Computer Interaction*, vol. 19, no. 1, pp. 9–24, 2004.
- [49] IHS Markit, “The complexities of physician supply and demand: projections from 2017 to 2032,” 2019.
- [50] Craig DeWolfe, Sarah Birch, Anne Callen Washofsky, Catherine Gardner, Robert McCarter, and Neha H Shah, “Patient outcomes in a pediatric hospital medicine service staffed with physicians and advanced practice providers,” *Hospital pediatrics*, vol. 9, no. 2, pp. 121–128, 2019.
- [51] Ellen T Kurtzman and Burt S Barnow, “A comparison of nurse practitioners, physician assistants, and primary care physicians patterns of practice and quality of care in health centers,” *Medical care*, vol. 55, no. 6, pp. 615–622, 2017.
- [52] Marleen H Lovink, Anke Persoon, Raymond TCM Koopmans, Anneke JAH Van Vught, Lisette Schoonhoven, and Miranda GH Laurant, “Effects of substituting nurse practitioners, physician assistants or nurses for physicians concerning health-care for the ageing population: a systematic literature review,” *Journal of advanced nursing*, vol. 73, no. 9, pp. 2084–2102, 2017.

- [53] Yihan Yang, Qi Long, Sandra L Jackson, Mary K Rhee, Anne Tomolo, Darin Olson, and Lawrence S Phillips, “Nurse practitioners, physician assistants, and physicians are comparable in managing the first five years of diabetes,” *The American journal of medicine*, vol. 131, no. 3, pp. 276–283, 2018.
- [54] Sven Cremer, Kris Doelling, Cody L Lundberg, Mike McNair, Jeongsik Shin, and Dan Popa, “Application requirements for robotic nursing assistants in hospital environments,” *International Society for Optics and Photonics*, 2016, pp. 98590E–98590E.
- [55] Takayuki Hoshi and Hiroyuki Shinoda, “Robot skin based on touch-area-sensitive tactile element,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2006, no. May, pp. 3463–3468, 2006.
- [56] Vincent Duchaine, Nicolas Lauzier, Mathieu Baril, Marc-Antoine Lacasse, and Clement Clément Gosselin, “A flexible robot skin for safe physical human robot interaction,” *2009 IEEE International Conference on Robotics and Automation*, pp. 3676–3681, 2009.
- [57] M. Inaba, Y. Hoshino, K. Nagasaka, T. Ninomiya, S. Kagami, and H. Inoue, “A full-body tactile sensor suit using electrically conductive fabric and strings,” *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS '96*, vol. 2, pp. 450–457, 1996.
- [58] Giorgio Cannata, Marco Maggiali, Giorgio Metta, and Giulio Sandini, “An embedded artificial skin for humanoid robots,” *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 434–438, 2008.
- [59] Hiroshi Yamazaki, Michiko Nishiyama, and Kazuhiro Watanabe, “A hemispheric hetero-core fiber optic tactile sensor for texture and hardness detection,” vol. 9754, pp. 97540X, 2016.
- [60] Youngmin Kim and Jong-Woong Kim, “Silver nanowire networks embedded in urethane acrylate for flexible capacitive touch sensor,” *Applied Surface Science*, vol. 363, pp. 1–6, 2016.
- [61] Sunkook Kim, Woong Choi, Woojin Rim, Youngtea Chun, Hongsik Shim, Hyukjun Kwon, Jongsoo Kim, Inseo Kee, Sungchul Kim, Sangyoon Lee, and Jongsun Park, “A highly sensitive capacitive touch sensor integrated on a thin-film-encapsulated active-matrix OLED for ultrathin displays,” *IEEE Transactions on Electron Devices*, vol. 58, no. 10, pp. 3609–3615, 2011.
- [62] Li Cao, Tae Song Kim, Susan C. Mantell, and Dennis L. Polla, “Simulation and fabrication of piezoresistive membrane type MEMS strain sensors,” *Sensors and Actuators, A: Physical*, vol. 80, no. 3, pp. 273–279, 2000.
- [63] Julián Castellanos-Ramos, Rafael Navas-González, Iván Fernández, and Fernando Vidal-Verdú, “Insights into the mechanical behaviour of a layered flexible tactile sensor,” *Sensors (Switzerland)*, vol. 15, no. 10, pp. 25433–25462, 2015.
- [64] M. Schrödner, R. I. Stohn, K. Schultheis, S. Sensfuss, and H. K. Roth, “Polymer field effect transistors made by laser patterning,” *Organic Electronics: physics, materials, applications*, vol. 6, no. 4, pp. 161–167, 2005.
- [65] Marcus Halik, Hagen Klauk, Ute Zschieschang, Tarik Kriem, G?nter Schmid, Wolfgang Radlik, and Klaus Wussow, “Fully patterned all-organic thin film transistors,” *Applied Physics Letters*, vol. 81, no. 2, pp. 289–291, 2002.

- [66] Udo Lang, Philipp Rust, and Jurg Dual, “Towards fully polymeric MEMS: Fabrication and testing of PEDOT/PSS strain gauges,” *Microelectronic Engineering*, vol. 85, no. 5-6, pp. 1050–1053, 2008.
- [67] Hussein Al-chami, “Inkjet Printing of Transducers,” M.S. thesis, The University of British Columbia, 2010.
- [68] G. Latessa, F. Brunetti, A. Reale, G. Saggio, and A. Di Carlo, “Piezoresistive behaviour of flexible PEDOT:PSS based sensors,” *Sensors and Actuators, B: Chemical*, vol. 139, no. 2, pp. 304–309, 2009.
- [69] Kyle R Shook, Ahsan Habib, Woo Ho Lee, and Dan O Popa, “Experimental testbed for robot skin characterization and interaction control,” pp. 911606–911606, 2014.
- [70] Caleb Nothnagle, Joshua R. Baptist, Joe Sanford, Woo Ho Lee, Dan O. Popa, and Muthu B. J. Wijesundara, “EHD printing of PEDOT: PSS inks for fabricating pressure and strain sensor arrays on flexible substrates,” in *Next-Generation Robotics II; and Machine Intelligence and Bio-inspired Computation: Theory and Applications IX*, 2015, vol. 9494, p. 949403.
- [71] Thomas M. Schweizer, “Electrical characterization and investigation of the piezoresistive effect of PEDOT:PSS thin film,” M.S. thesis, Georgia Institute of Technology, 2005.
- [72] Siddhartha S Srinivasa, Dave Ferguson, Casey J Helfrich, Dmitry Berenson, Alvaro Collet, Rosen Diankov, Garratt Gallagher, Geoffrey Hollinger, James Kuffner, and Michael Vande Weghe, “Herb: a home exploring robotic butler,” *Autonomous Robots*, vol. 28, no. 1, pp. 5–20, 2010.
- [73] Rodney A Brooks, Cynthia Breazeal, Matthew Marjanović, Brian Scassellati, and Matthew M Williamson, “The cog project: Building a humanoid robot,” pp. 52–87. Springer, 1999.
- [74] Siyuan Feng, Eric Whitman, X Xinjilefu, and Christopher G Atkeson, “Optimization based full body control for the atlas robot,” *IEEE*, 2014, pp. 120–127.
- [75] CS Lovchik and Myron A Diftler, “The robonaut hand: A dexterous robot hand for space,” *IEEE*, 1999, vol. 2, pp. 907–912.
- [76] Joe Sanford, Isura Ranatunga, and Dan Popa, “Physical human-robot interaction with a mobile manipulator through pressure sensitive robot skin,” *ACM*, 2013, p. 60.
- [77] Keenan A Wyrobek, Eric H Berger, HF Machiel Van der Loos, and J Kenneth Salisbury, “Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot,” *IEEE*, 2008, pp. 2165–2170.
- [78] Jeremy A Marvel and Rick Norcross, “Implementing speed and separation monitoring in collaborative robot workcells,” *Robotics and Computer-Integrated Manufacturing*, vol. 44, pp. 144–155, 2017.
- [79] Rommel Alonzo, Sven Cremer, Fahad Mirza, Sandesh Gowda, Larry Mastromoro, and Dan O Popa, “Multi-modal sensor and hmi integration with applications in personal robotics,” *International Society for Optics and Photonics*, 2015, pp. 949409–949409.
- [80] Sven Cremer, Fahad Mirza, Yathartha Tuladhar, Rommel Alonzo, Anthony Hingeley, and Dan O Popa, “Investigation of human-robot interface performance in household environments,” 2016, vol. 9859, pp. 985904–1.

- [81] Sung Wook Moon, Young Jin Kim, Ho Jun Myeong, Chang Soo Kim, Nam Ju Cha, and Dong Hwan Kim, "Implementation of smartphone environment remote control and monitoring system for android operating system-based robot platform," *IEEE*, 2011, pp. 211–214.
- [82] Damon Kohler and Ken Conley, "rojava—an implementation of ros in pure java with android support," [2013-05-15]. <https://github.com/rojava>, 2011, Accessed: 2017-02-01.
- [83] Rethink Robotics, "Baxter datasheet," *Online at* <http://www.rethinkrobotics.com/baxter/tech-specs/>, 2016, Accessed: 2017-02-01.
- [84] Ioan A Sucan and Sachin Chitta, "Moveit!," *Online at* <http://moveit.ros.org>, 2013, Accessed: 2017-02-01.
- [85] S Niekum, "Ar tag tracking library - ar\_track\_alvar - ros wiki," *Online at* [http://www.ros.org/wiki/ar\\_track\\_alvar](http://www.ros.org/wiki/ar_track_alvar), 2013, Accessed: 2017-02-01.
- [86] Ali Safavi and Mehrdad H. Zadeh, "Model-Based Haptic Guidance in Surgical Skill Improvement," in *IEEE International Conference on Systems, Man, and Cybernetics*, oct 2015, pp. 1104–1109.
- [87] Andrea Cherubini, Robin Passama, André Crosnier, Antoine Lasnier, and Philippe Fraisse, "Collaborative manufacturing with physical human-robot interaction," *Robotics and Computer-Integrated Manufacturing*, vol. 40, pp. 1–13, 2016.
- [88] Haoyong Yu, Sunan Huang, Gong Chen, Yongping Pan, and Zhao Guo, "Human-Robot Interaction Control of Rehabilitation Robots with Series Elastic Actuators," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1089–1100, 2015.
- [89] Agostino De Santis, Bruno Siciliano, Alessandro De Luca, and Antonio Bicchi, "An atlas of physical human-robot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, mar 2008.
- [90] Brenna D. Argall and Aude G. Billard, "A survey of tactile human-robot interactions," *Robotics and Autonomous Systems*, vol. 58, no. 10, pp. 1159–1176, 2010.
- [91] Aslam Pervez and Jeha Ryu, "Safe physical human robot interaction-past, present and future," *Journal of Mechanical Science and Technology*, vol. 22, no. 3, pp. 469–483, may 2008.
- [92] Frank L Lewis, D M Dawson, and Chaouki T Abdallah, "Robot Manipulator Control," *Theory and Practice*, p. 638, 2004.
- [93] Elena Gribovskaya, Abderrahmane Kheddar, and Aude Billard, "Motion learning and adaptive impedance for robot control during physical interaction with humans," in *IEEE International Conference on Robotics and Automation*, may 2011, pp. 4326–4332.
- [94] Satoshi Suzuki and Katsuhisa Furuta, "Adaptive impedance control to enhance human skill on a haptic interface system," *Journal of Control Science and Engineering*, vol. 2012, pp. 1–10, 2012.
- [95] Yanan Li, Shuzhi Sam Ge, and Chenguang Yang, "Learning impedance control for physical robot-environment interaction," *International Journal of Control*, vol. 85, no. 2, pp. 182–193, feb 2012.



- [96] JR Ragazzini, "Engineering aspects of the human being as a servo-mechanism," in *Proceedings of the Meeting of the American Psychological Association*, 1948.
- [97] Anil Phatak, Howard Weinert, Ilana Segall, and Carroll N Day, "Identification of a modified optimal control model for the human operator," *Automatica*, vol. 12, no. 1, pp. 31 – 41, 1976.
- [98] Fu-Chuang 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.
- [99] F L Lewis, K Liu, and A Yesildirek, "Neural net robot controller with guaranteed tracking performance.," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 6, no. 3, pp. 703–15, jan 1995.
- [100] Shuzhi Sam Ge, Tong Heng Lee, and C J Harris, *Adaptive Neural Network Control of Robotic Manipulators*, vol. 19 of *World Scientific Series in Robotics and Intelligent Systems*, World Scientific, dec 1998.
- [101] F. L. Lewis, S. Jagannathan, and A. Yesildirak, *Neural Network Control Of Robot Manipulators And Nonlinear Systems*, CRC Press, 1998.
- [102] Neville Hogan and Dagmar Sternad, "Sensitivity of smoothness measures to movement duration, amplitude, and arrests.," *Journal of motor behavior*, vol. 41, no. 6, pp. 529–34, nov 2009.
- [103] Duane T McRuer and Ezra S Krendel, "Mathematical models of human pilot behavior," Tech. Rep., Advisory group for aerospace research and Development, 1974.
- [104] Alessandro De Luca and Fabrizio Flacco, "Integrated control for phri: Collision avoidance, detection, reaction and collaboration," in *Biomedical Robotics and Biomechanics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*. IEEE, 2012, pp. 288–295.
- [105] Perla Maiolino, Marco Maggiali, Giorgio Cannata, Giorgio Metta, and Lorenzo Natale, "A flexible and robust large scale capacitive tactile system for robots," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3910–3917, 2013.
- [106] Philipp Mittendorf, Emmanuel Dean, and Gordon Cheng, "3d spatial self-organization of a modular artificial skin," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 3969–3974.
- [107] Luca Maiolo, Alessandro Pecora, Francesco Maita, Antonio Minotti, Emiliano Zampetti, Simone Pantalei, Antonella Macagnano, Andrea Bearzotti, Davide Ricci, and Guglielmo Fortunato, "Flexible sensing systems based on polysilicon thin film transistors technology," *Sensors and Actuators B: Chemical*, vol. 179, pp. 114–124, 2013.
- [108] Jartuwat Rajruangrabin and Dan O Popa, "Enhancement of manipulator interactivity through compliant skin and extended kalman filtering," in *IEEE International Conference on Automation Science and Engineering, 2007. CASE 2007*. IEEE, 2007, pp. 1111–1116.
- [109] Sven Cremer, Isura Ranatunga, Sumit K Das, Indika B Wijayasinghe, and Dan O Popa, "Neuroadaptive calibration of tactile sensors for robot skin," in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1079–1085.

- [110] Harry West, Evangelos Papadopoulos, Steven Dubowsky, and Hanson Cheah, "A method for estimating the mass properties of a manipulator by measuring the reaction moments at its base," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on.* IEEE, 1989, pp. 1510–1516.
- [111] Guangjun Liu, Karl Iagnemma, Steven Dubowsky, and Guillaume Morel, "A base force/torque sensor approach to robot manipulator inertial parameter estimation," in *IEEE International Conference on Robotics and Automation, 1998. Proceedings.* IEEE, 1998, vol. 4, pp. 3316–3321.
- [112] Martin Grotjahn and Bodo Heimann, "Determination of dynamic parameters of robots by base sensor measurements," *IFAC Proceedings Volumes*, vol. 33, no. 27, pp. 279–284, 2000.
- [113] Guillaume Morel and Steven Dubowsky, "The precise control of manipulators with joint friction: A base force/torque sensor method," in *1996 IEEE International Conference on Robotics and Automation, 1996. Proceedings.,* IEEE, 1996, vol. 1, pp. 360–365.
- [114] Christian Ott and Yoshihiko Nakamura, "Admittance control using a base force/torque sensor.," *IFAC Proceedings Volumes*, vol. 42, no. 16, pp. 467–472, 2009.
- [115] Christian Ott and Yoshihiko Nakamura, "Base force/torque sensing for position based cartesian impedance control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009.* IEEE, 2009, pp. 3244–3250.
- [116] Franck Geffard, Claude Andriot, Alain Micaelli, and Guillaume Morel, "On the use of a base force/torque sensor in teleoperation," in *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA '00.* IEEE, 2000, vol. 3, pp. 2677–2683.
- [117] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry, *A mathematical introduction to robotic manipulation*, CRC press, 1994.
- [118] RPW Duin, P Juszczak, P Paclik, E Pekalska, D De Ridder, DMJ Tax, and S Verzaikov, "A matlab toolbox for pattern recognition," *PRTTools version*, vol. 3, pp. 109–111, 2000.
- [119] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo, *Robotics: modelling, planning and control*, Springer Science & Business Media, 2010.
- [120] Dieter Vischer and Oussama Khatib, "Design and development of high-performance torque-controlled joints," *IEEE Transactions on robotics and automation*, vol. 11, no. 4, pp. 537–544, 1995.
- [121] MH Lee and HR Nicholls, "Tactile sensing for mechatronics-a state of the art survey, mechatronics, vol. 9," 1999.
- [122] JC Hake and J Farah, "Design of a joint torque sensor for the unimation puma 500 arm," *Final Report, ME210, University of Stanford, CA*, 1984.
- [123] Guillaume Morel, Karl Iagnemma, and Steven Dubowsky, "The precise control of manipulators with high joint-friction using base force/torque sensing," *Automatica*, vol. 36, no. 7, pp. 931–941, 2000.
- [124] Eftychios G Christoforou, "On-line parameter identification and adaptive control of rigid robots using base reaction forces/torques," in *Proceedings 2007 IEEE International Conference on Robotics and Automation.* IEEE, 2007, pp. 4956–4961.

- [125] Gabriele Buondonno and Alessandro De Luca, "Combining real and virtual sensors for measuring interaction forces and moments acting on a robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 794–800.
- [126] Fred D Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS quarterly*, pp. 319–340, 1989.
- [127] I Ju Chen, Kuei-Feng Yang, Fu-In Tang, Chun-Hsia Huang, and Shu Yu, "Applying the technology acceptance model to explore public health nurses intentions towards web-based learning: A cross-sectional questionnaire survey," *International journal of nursing studies*, vol. 45, no. 6, pp. 869–878, 2008.
- [128] Martin Fishbein and Icek Ajzen, "Belief, attitude, intention, and behavior: An introduction to theory and research," 1977.
- [129] Indika B Wijayasinghe, Sumit K Das, Haylie L Miller, Nicolta L Bugnariu, and Dan O Popa, "Head-eye coordination of humanoid robot with potential controller," *Journal of intelligent and robotic systems*, 2018.
- [130] Mohammad Nasser Saadatzi, Sumit K Das, Indika B Wijayasinghe, and Dan O Popa, "Precision grasp control with a pneumatic gripper and a novel fingertip force sensor," in *Automation Science and Engineering (CASE), 2018 IEEE International Conference on*. IEEE, 2018.
- [131] Indika B Wijayasinghe, Haylie L Miller, Sumit K Das, Nicoleta L Bugnariu, and Dan O Popa, "Human-like object tracking and gaze estimation with pkd android," in *Sensors for Next-Generation Robotics III*. International Society for Optics and Photonics, 2016, vol. 9859, p. 985906.
- [132] Indika B Wijayasinghe, Joseph D Sanford, Shamsudeen Abubakar, Mohammad Nasser Saadatzi, Sumit K Das, and Dan O Popa, "Optimal accelerometer placement on a robot arm for pose estimation," in *Smart Biomedical and Physiological Sensor Technology XIV*. International Society for Optics and Photonics, 2017, vol. 10216, p. 102160B.
- [133] Maryam Moosaei, Sumit K Das, Dan O Popa, and Laurel D Riek, "Using facially expressive robots to calibrate clinical pain perception," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2017, pp. 32–41.
- [134] Oguz Yetkin, Joe Sanford, Fahad Mirza, Roopak Karulkar, Sumit K Das, and Dan O Popa, "Control of a powered prosthetic hand via a tracked glove," *Journal of Medical Devices*, vol. 9, no. 2, pp. 020920, 2015.
- [135] Sven Cremer, Indika B Wijayasinghe, Sumit K Das, and Dan O Popa, "Skinsim 2.0: A design and simulation tool for robot skin with closed-loop phri controllers," TASE 2016 - Under Review.
- [136] Sumit K Das, S. Abubakar, Indika B Wijayasinghe, and Dan O Popa, "Human-robot interaction with nursing assistant robot," .
- [137] Sumit K Das, S. Abubakar, Indika B Wijayasinghe, and Dan O Popa, "Model-free inverse kinematics using imu sensors," .
- [138] Sumit K Das, Indika B Wijayasinghe, and Dan O Popa, "Enhanced hri using base-sensor," .

- [139] Oguz Yetkin, Joseph D Sanford, Dan O Popa, Lena C Wallace, Fahad Mirza, Roopak Karulkar, Sumit Kumar Das, Joshua R Baptist, and Paul J Carpenter, “Systems, apparatuses and methods for controlling prosthetic devices by gestures and other modalities,” Oct. 19 2017, US Patent App. 15/488,500.
- [140] Mohammad Nasser Saadatzi, Sumit Kumar Das, Indika B. Wijayasinghe, and Dan O. Popa, “Adaptive robotic nursing assistant,” Apr. 16 2019, US Patent App. 62/834,689.

## APPENDIX A

### LIST OF PUBLICATIONS

#### A.1 Published Papers

1. Sumit K Das, Indika B Wijayasinghe, Mohammad Nasser Saadatzi, and Dan O Popa, “Whole body human-robot collision detection using base-sensor neuroadaptive interaction,” in *Automation Science and Engineering (CASE), 2018 IEEE International Conference on*. IEEE, 2018
2. Sumit K Das, Mohammad Nasser Saadatzi, Shamsudeen Abubakar, and Dan O Popa, “Joint torque estimation using base force-torque sensor to facilitate physical human-robot interaction (phri),” in *2019 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019
3. Sumit Kumar Das, Ankita Sahu, and Dan O Popa, “Mobile app for human-interaction with sitter robots,” in *Smart Biomedical and Physiological Sensor Technology XIV*. International Society for Optics and Photonics, 2017, vol. 10216, p. 102160D
4. Sumit Kumar Das, Joshua R Baptist, Ritvij Sahasrabuddhe, Woo H Lee, and Dan O Popa, “Package analysis of 3d-printed piezoresistive strain gauge sensors,” in *Sensors for Next-Generation Robotics III*. International Society for Optics and Photonics, 2016, vol. 9859, p. 985905
5. Indika B Wijayasinghe, Sumit K Das, Haylie L Miller, Nicolta L Bugnariu, and Dan O Popa, “Head-eye coordination of humanoid robot with potential controller,” *Journal of intelligent and robotic systems*, 2018
6. Sven Cremer, Isura Ranatunga, Sumit K Das, Indika B Wijayasinghe, and Dan O

- Popa, “Neuroadaptive calibration of tactile sensors for robot skin,” in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1079–1085
7. Mohammad Nasser Saadatzi, Sumit K Das, Indika B Wijayasinghe, and Dan O Popa, “Precision grasp control with a pneumatic gripper and a novel fingertip force sensor,” in *Automation Science and Engineering (CASE), 2018 IEEE International Conference on*. IEEE, 2018
  8. Indika B Wijayasinghe, Haylie L Miller, Sumit K Das, Nicoleta L Bugnariu, and Dan O Popa, “Human-like object tracking and gaze estimation with pkd android,” in *Sensors for Next-Generation Robotics III*. International Society for Optics and Photonics, 2016, vol. 9859, p. 985906
  9. Indika B Wijayasinghe, Joseph D Sanford, Shamsudeen Abubakar, Mohammad Nasser Saadatzi, Sumit K Das, and Dan O Popa, “Optimal accelerometer placement on a robot arm for pose estimation,” in *Smart Biomedical and Physiological Sensor Technology XIV*. International Society for Optics and Photonics, 2017, vol. 10216, p. 102160B
  10. Maryam Moosaei, Sumit K Das, Dan O Popa, and Laurel D Riek, “Using facially expressive robots to calibrate clinical pain perception,” in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2017, pp. 32–41
  11. Oguz Yetkin, Joe Sanford, Fahad Mirza, Roopak Karulkar, Sumit K Das, and Dan O Popa, “Control of a powered prosthetic hand via a tracked glove,” *Journal of Medical Devices*, vol. 9, no. 2, pp. 020920, 2015

## **A.2 Conditionally Accepted**

1. Sven Cremer, Sumit K Das, Indika B Wijayasinghe, Dan O Popa, and Frank L

Lewis, “Model-free online neuroadaptive controller with intent estimation for physical human-robot interaction,” TRO Journal - Conditionally Accepted

### **A.3 Papers Under Review**

1. Sven Cremer, Indika B Wijayasinghe, Sumit K Das, and Dan O Popa, “Skinsim 2.0: A design and simulation tool for robot skin with closed-loop phri controllers,” TASE 2016 - Under Review

### **A.4 Papers in Preparation**

1. Sumit K Das, S. Abubakar, Indika B Wijayasinghe, and Dan O Popa, “Human-robot interaction with nursing assistant robot,”
2. Sumit K Das, S. Abubakar, Indika B Wijayasinghe, and Dan O Popa, “Model-free inverse kinematics using imu sensors,”
3. Sumit K Das, Indika B Wijayasinghe, and Dan O Popa, “Enhanced hri using base-sensor,”

### **A.5 Patent Pending**

1. Oguz Yetkin, Joseph D Sanford, Dan O Popa, Lena C Wallace, Fahad Mirza, Roopak Karulkar, Sumit Kumar Das, Joshua R Baptist, and Paul J Carpenter, “Systems, apparatuses and methods for controlling prosthetic devices by gestures and other modalities,” Oct. 19 2017, US Patent App. 15/488,500
2. Mohammad Nasser Saadatzi, Sumit Kumar Das, Indika B. Wijayasinghe, and Dan O. Popa, “Adaptive robotic nursing assistant,” Apr. 16 2019, US Patent App. 62/834,689

APPENDIX B

QUESTIONNAIRES

TABLE B.1: Walker Scenario Questionnaire

1	Did you lose your balance?
2	How responsive was the robot?
3	How often did you feel like you'd lose your balance?
4	Did you find the robot helpful?
5	How did you find it to operate the walker?
6	How was it keeping the robot straight?
7	How was turning the robot?
8	How intuitive was the robot?
9	Is the robot heavy going straight?
10	Is it heavy to turn the walker?
11	How easy is it learning to use the walker?
12	How easy is it remembering how to use the robot?
13	Do you sufficient support from the walker?
14	Are you satisfied with the walker as a mobility aid?
15	How easy to use it in crowded situations?
16	How easy is it to go through doorways?
17	How easy is it to avoid bumping into objects?
18	How comfortable walking it in front of people?
19	How big do you think the walker is?
20	How do you find the speed of the walker?



TABLE B.2: Technology Acceptance Questionnaire

1	My job would be difficult without the robot
2	Using the robot gives me greater control over work
3	Using the robot improves my performance
4	The robot addresses my work-related needs
5	The robot saves me time
6	Robot enables me to accomplish tasks more quickly
7	Robot supports critical aspects of my job
8	Robot accomplishes more work than possible otherwise
9	Robot reduces time I spend on unproductive activities
10	Robot enhances effectiveness of job
11	Improves quality of work I do
12	Increases Productivity
13	Makes it easier to do my job
14	I find ARNA useful overall
15	I am often confused when I use ARNA
16	I make frequent errors when using ARNA
17	Interacting with ARNA is often frustrating
18	I need to consult the user manual often when using ARNA
19	Interacting with ARNA requires a lot of my mental effort
20	I find it easy to recover from errors using ARNA
21	The system is rigid and inflexible to work with
22	I find it easy to get ARNA to do what I want it to do
23	The robot often behaves in unexpected ways
24	I find it cumbersome to use ARNA
25	My interaction with ARNA is easy to understand
26	It is easy for me to remember how to perform tasks while using ARNA

27	ARNA provides helpful guidance in performing tasks
28	Overall, I find ARNA easy to use

TABLE B.3: Human-Robot Interaction Questionnaire

1	Fake/Natural
2	Machinelike/Humanlike
3	Unconscious/Conscious
4	Artificial/Lifelike
5	Rigid/Elegant movement
6	Dead/Alive
7	Stagnant/Lively
8	Mechanical/Organic
9	Artificial/Lifelike
10	Inert/Interactive
11	Apathetic/Responsive
12	Dislike/Like
13	Unfriendly/Friendly
14	Unkind/Kind
15	Unpleasant/Pleasant
16	Awful/Nice
17	Incompetent/Competent
18	Ignorant/Knowledgeable
19	Irresponsible/Responsible
20	Unintelligent/Intelligent
21	Foolish/Sensible
22	Anxious/Relaxed
23	Agitated/Calm

24	Quiescent/Surprised
25	Easy to learn/hard to learn
26	Efficient/Inefficient
27	Memorable/Forgettable
28	Accurate/Inaccurate
29	Satisfied/Dissatisfied
30	Using ARNA would enable me to accomplish tasks
31	Using ARNA would improve my performance
32	Using the system would increase my productivity
33	Using ARNA would enhance my effectiveness
34	Using ARNA would make it easier to perform tasks
35	I would find ARNA useful
36	Learning to operate ARNA would be easier
37	I would find it easy to get ARNA to do what I want
38	My interaction would be clear and understandable
39	I would find ARNA to be flexible to interact with
40	It would be easy to become skillful with ARNA
41	I would find ARNA easy to use

TABLE B.4: Human-Robot Interaction Questionnaire

1	How convenient is it to drive the robot with the tablet interface?
2	How quickly does the robot arrive at its destination using the tablet interface?
3	How safe do you think the robot is while you are controlling it with the tablet interface?
4	How much attention does it take to drive the robot to the desired place while avoiding obstacles?

5	How easy is it to drive the robot to the desired place while avoiding obstacles?
6	How convenient is it to tell the robot where to go using the interface?
7	What would you say the speed of the robot is when moving around the room?
8	How safe do you think the robot is when navigating to the desired place?
9	How easy is it to grab items with the robot arm using the tablet interface?
10	How stably did the robot gripper grasp the item?
11	How safe do you think the robot arm is when it hands over the fetched items?

# Sumit Kumar Das

Electrical Engineer and  
Robotist



11201 Sabine Way, Louisville,  
KY, USA 40229



+1 (817) 744 0214



sumitkdas89



sumit.k.das@gmail.com



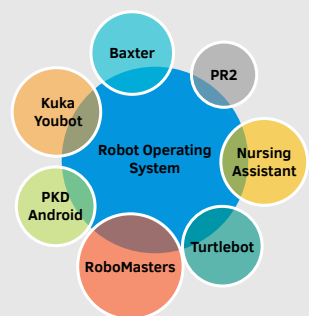
/in/sumitkdas89



sumitdas89

## Skills

### Robotics



### Technical skills

ROS • Gazebo • C • Embedded C /  
Assembly Language programming • C++  
• C# • Java • Eigen • PCL • .NET  
Framework • Matlab • SQL • HTML •  
Solidworks • NI Labview • COMSOL •  
Microsoft Kinect API • VB

Realtime Programming • EtherCAT •  
Networking • Distributed Computing  
• Rapid Prototyping • Google Cartog-  
rapher

## CURRICULUM VITAE

### Education

- 2015 - Present **PhD. Candidate, Electrical Engineering** University of Louisville, USA  
Dissertation: *Adaptive Interaction With A Nursing Robot Using Direc-  
tive Observer and Lead Assistance(DOLA) Framework*  
Supervisor: Dr. Dan O. Popa
- 2013 - 2015 **MSc., Electrical Engineering** University of Texas at Arlington, USA  
Thesis: *Realistic interaction with social robots via facial expressions  
and neck-eye coordination*  
Supervisor: Dr. Dan O. Popa
- 2006 - 2010 **BTech., Electronics and Telecommunication** Biju Patnaik University of  
Technology, India

### Experience

- Jan 2015 - Present **Graduate Research Assistant** Next Generation Systems Lab  
Nursing Assistant Robot  
• Managed a team of graduate students and undergraduate students  
for building the robot.  
• Autonomous navigation module in development now.  
• Developed the ROS framework for the whole robot.  
• Built control structure for a 6-DOF arm on a mobile platform  
• Developed a networking system for realtime application using  
EtherCAT and Ethernet.  
Adaptive Physical Human-Robot Interaction Controllers  
• Developed and implemented a Neuro-Adaptive Controller on a  
robotics manipulator.  
• Formulated a controller for whole body collision sensing for robots.  
Sensor Package Simulation and FEA Analysis  
• Used COMSOL for FEA analysis of tactile sensor package.  
Robomasters Mobile Manipulation Challenge  
• Lead a team of 4 students for this challenge.  
• Implemented Google Cartographer and ROS Nav packages for au-  
tonomous navigation.  
• Developed OpenCV based custom pattern recognition.  
• Used embedded C to program under-laying micro-controllers for  
motion control programming.  
Micro-robotics Challenge  
• Placed 2nd in the Mobile Micro robotics Assembly challenge held  
at International Conference on Robotics and Automation (ICRA)  
2017, Singapore.  
• Placed 2nd in the Mobile Micro robotics Mobility challenge held  
at International Conference on Robotics and Automation (ICRA)  
2017, Singapore.
- Sep 2013 - Dec 2015 **Graduate Research Assistant** Next Generation Systems Lab  
Humanoid Robot - Philip K. Dick  
• Research in Neck-Eye Coordination during target tracking.  
• Mimicking facial expressions of human user.  
• Lip synchronization during speech.  
PR2 Robot  
• Implemented autonomous navigation for the robot.  
• Developed code base for autonomous lift operation by the robot.  
Prosthetic Robotic Arm  
• Created code-base for driving a robotic prosthetic arm using ges-  
tures.  
• The system was implemented using both tactile sensors as well as  
IR sensors.  
Micro-robotics Challenge  
• Placed 1st in the Mobile Micro robotics Assembly challenge held  
at International Conference on Robotics and Automation (ICRA)  
2014, Hong Kong.

- Placed 2nd in the Mobile Micro robotics Mobility challenge held at International Conference on Robotics and Automation (ICRA) 2014, Hong Kong.

#### Amazon Picking Challenge

- Part of the team which represented UTA at the challenge.
- Developed object identification modules for pick and place tasks.
- Implemented the developed modules on Baxter and PR2 robot.

Nov 2010 - **Senior Programmer** Accenture Services Pvt. Ltd.  
Aug 2013

- Managed a team consisting of 5 programmers
- Application Tester certification from Massachusetts Institute of Technology (MIT).
- Information Technology Infrastructure Library V3 Certified

## Publications

- Sumit K Das, Indika B Wijayasinghe, Mohammad Nasser Saadatzi, and Dan O Popa, "Whole body human-robot collision detection using base-sensor neuroadaptive interaction," in *Automation Science and Engineering (CASE), 2018 IEEE International Conference on*. IEEE, 2018
- Sumit Kumar Das, Ankita Sahu, and Dan O Popa, "Mobile app for human-interaction with sitter robots," in *Smart Biomedical and Physiological Sensor Technology XIV*. International Society for Optics and Photonics, 2017, vol. 10216, p. 102160D
- Sumit Kumar Das, Joshua R Baptist, Ritvij Sahasrabuddhe, Woo H Lee, and Dan O Popa, "Package analysis of 3d-printed piezoresistive strain gauge sensors," in *Sensors for Next-Generation Robotics III*. International Society for Optics and Photonics, 2016, vol. 9859, p. 985905
- Sven Cremer, Isura Ranatunga, Sumit K Das, Indika B Wijayasinghe, and Dan O Popa, "Neuroadaptive calibration of tactile sensors for robot skin," in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1079–1085
- Mohammad Nasser Saadatzi, Sumit K Das, Indika B Wijayasinghe, and Dan O Popa, "Precision grasp control with a pneumatic gripper and a novel fingertip force sensor," in *Automation Science and Engineering (CASE), 2018 IEEE International Conference on*. IEEE, 2018
- Indika B Wijayasinghe, Haylie L Miller, Sumit K Das, Nicoleta L Bugnariu, and Dan O Popa, "Human-like object tracking and gaze estimation with pkd android," in *Sensors for Next-Generation Robotics III*. International Society for Optics and Photonics, 2016, vol. 9859, p. 985906
- Indika B Wijayasinghe, Joseph D Sanford, Shamsudeen Abubakar, Mohammad Nasser Saadatzi, Sumit K Das, and Dan O Popa, "Optimal accelerometer placement on a robot arm for pose estimation," in *Smart Biomedical and Physiological Sensor Technology XIV*. International Society for Optics and Photonics, 2017, vol. 10216, p. 102160B
- Maryam Moosaei, Sumit K Das, Dan O Popa, and Laurel D Riek, "Using facially expressive robots to calibrate clinical pain perception," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2017, pp. 32–41
- Oguz Yetkin, Joe Sanford, Fahad Mirza, Roopak Karulkar, Sumit K Das, and Dan O Popa, "Control of a powered prosthetic hand via a tracked glove," *Journal of Medical Devices*, vol. 9, no. 2, pp. 020920, 2015
- Sven Cremer, Sumit K Das, Indika B Wijayasinghe, Dan O Popa, and Frank L Lewis, "Model-free online neuroadaptive controller with intent estimation for physical human-robot interaction," *TRO Journal* - Under Review
- Sven Cremer, Indika B Wijayasinghe, Sumit K Das, and Dan O Popa, "Skinsim 2.0: A design and simulation tool for robot skin with closed-loop phri controllers," *TASE 2016* - Under Review

## Patent Pending

- Oguz Yetkin, Joseph D Sanford, Dan O Popa, Lena C Wallace, Fahad Mirza, Roopak Karulkar, Sumit Kumar Das, Joshua R Baptist, and Paul J Carpenter, "Systems, apparatuses and methods for controlling prosthetic devices by gestures and other modalities," Oct. 19 2017, US Patent App. 15/488,500