# Impedance Control Algorithm for Physical Rehabilitation Robots

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Sai Praneeth Cheedella

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Dr.Arshia Khan

August 2018

Acknowledgements

I would first like to express my gratitude to my advisor Dr.Arshia Khan for her constant support throughout my Master's program and my Thesis. This research wouldn't have been possible without her.

I would like to thank Dr.Haiyang Wang and Dr. Desineni Subbaram Naidu for serving on my thesis defense committee and taking time to evaluate my work. I am thankful to all my professors during my course period at UMD for teaching me some valuable insights and fun courses.

I would also like to thank the Computer Science Department staff for their constant reminders and their timely help.

I would also like to thank my friend Arshia Hassan for her constant talks of inspiration.

Lastly, I would like to thank my family and friends for their encouragement.

# Dedication

I would like to dedicate this thesis to my parents Rajendra Prasad and Padmaja Kiran who made it possible for me to pursue my masters and my brother Pruthvi for his constant and nagging support. I would also like to mention my Late Grandfather Prakash Rao who would have been very happy to see this day. I also thank all my family members for their everlasting support.

Abstract

Cardiac care recovery has evolved over a period of time; automation in cardiac rehabilitation is gaining momentum in an attempt to achieve the goal of shorter hospital stay and reduction in hospitalizations. Care Robots are one of the solutions that have the potential to increase the efficiency of care. Patients who have undergone a major surgery are fragile and require constant care around the clock.Tasks such as lifting the patient or getting the patient out of bed require physical strength and create complications not only for the patients but also for the caregivers. We have chosen Baxter(Buddy) research robot as our first care-bot prototype to attempt to assist in tasks that can be strenuous on human care givers. We are focusing on the physical strengths of Baxter in helping patients get out of bed. In this thesis we have designed a control mechanism that adjusts to patients physical metrics with minimal human intervention.Robotic motor joints are programmed to manipulate environment around the patient.On-board Torque control mechanisms have been modified to accommodate individual human needs based on their body type. In addition to Torque control we have also incorporated Baxter with a Object Recognition technique that employs the headboard camera in identifying the position of the subject on the bed.

# Contents

# List of Figures

# 1 Introduction

Open heart surgery is a procedure which puts a person under a lot of trauma. Rehabilitation after this surgery helps to regain the patient's mobility and return to usual living situation. It involves several multifaceted and multidisciplinary core components such as nutritional counseling, adherence to pharmacological therapies, psycho social interventions, physical activity counseling and exercise training.Basic daily tasks like getting out of bed also puts a lot of stress on the chest and will require a physical support. Patients might need devoted attention and physical support of a caregiver throughout the rehabilitation stage.

Robotic Home Assistants are the technical aids that provide support and help in a person's daily life. They can help you with basic tasks like medication reminders, physical support and also constant supervision patient's health without human need. In this research we hypothesize a human like robotic assistance to assist in physical Cardiac Rehabilitation after open heart surgery. Hence, The calculated impedance control in Baxter will counter the varying weights of individuals getting out of bed.

This research encompasses around the managing of by improving the stability and control in the delivery system. The existing care robots like PARO[5] deal with the emotional and social companionship but not with the physical help. Our research mainly focuses on the physical help and support required for the rehabilitating patients after their surgery. The algorithm used in this scenario regulates the framework of robotic movements and output force applied. Hence the algorithm used for our research is named as Adaptive (impedance) control algorithm.

The adaptive control algorithm designed for this research will help enhance the robotic interaction with the patient. It factors in parameters like the position and post-surgical condition of the patient to support him with gently whenever needed.

This research impacts health care, CS and Robotic fields. In the perspective of health care, the robots would be much more stable, provide better care and will be able to keep constant track of the patient's activities without the need of an actual person. From the robotics and CS perspective, the adaptive impedance algorithm can be altered so that it is compatible with various humanoid devices, which can further improve the human robot interaction. These robots can be used to not only support the patients, but can lift the patients from wheelchairs and place them on bed. If this research is successful we will be able to utilize this technology to other recovery and patient support needs

# 2 Background

## 2.1 Background

This background chapter gives you a insight of various topics and technologies that help you better understand the chapters further ahead.

### 2.1.1 Human Robot Interaction

Human Robot Interaction(HRI) is mainly about the design, understanding and evaluation of robotic systems, which involve humans and robots interacting with each other[13]. Direct interaction with computers have reached a point where usability and usefulness including the risk factor are widely accepted in computing.

Advancements in the computer technology, Artificial Intelligence(AI), Speech Simulation and Understanding lead to a huge break in Robotic technologies and HRI. These Autonomous robots can identify and track user position, display text and travel on command while avoiding obstacles. These robots will be assisting humans in tasks which are unsafe and boring to people. A robotic assistant could do some basic chores for the elderly people like holding a cup, bringing a newspaper. Robots can even execute medical and welfare applications, home and office automation, which are very difficult to be automated but could be executed with human assistance. In most of the HRI systems robots move passively based on the force applied to the system by a human to accomplish tasks together with a human. If the robots could work not only based on the passive input but also work actively based on human intentions, the knowledge of the tasks more effective Human Robot coordination can be achieved.

### 2.1.2 Automation in Health Care Industry

Automation of a process deals with using information technology and control mechanisms which reduce human intervention and increase productivity while cutting the costs. Automation helps in decreasing the labor costs and increasing the accuracy of the output product. Starting with FORD car company to the ATM's(Automated Teller Machines) in the recent age automation has spread to our everyday life scenarios. There has been an influx of research and effort to bring the same level of automation and accuracy to the health care industry. Main goal of automation in health industry is not to replace doctors or nurses but work along with them to increase the efficiency in the care delivery process and patient engagement. Automation in health care industry will help reducing the costs, eliminating waste and improving the health care delivery.

Robots are faster and far more accurate than humans, this saves time and reduces waste. Human involvement can be reduced in areas like radiation therapy where exposure to radiation is great risk. Automation in medical field can range from a robotic pharmacist to robotic surgeon. Robots have been used in the surgeries since 2000 a famous example for a surgical robot is DA Vinci, which performed numerous minimal surgeries.

The biggest sector in health care industry where automation is very much needed are care facilities. A care facility is a place where patients are admitted for round the clock care for trauma recovery or due to an illness. These facilities are often understaffed and the employee satisfaction is very low due demanding working hours and very robust work.

### 2.1.3 Care Facilities and Care Robots

Care facilities / Aid homes are homes for elderly , disabled ,people who need assistance after due to a mental illness like dementia and people recovering from a recent medical trauma. These rehabilitation centers employ care givers who take care of people/patients in round the clock. Most of these care givers have to work untimely hours and have to take care of multiple patients at the same time. Both the patients and employees are physically or mentally stressed at some point of time. Their tasks are often physically demanding and are plagued by job insecurity.[11]

With the help of automation, we can assist this care workers so that they don't have to undergo robust tasks like 24 hour tracking of patients, alerts and reminders for medicine and physical assistance of carrying patients around. Robotic Home Assistants are the technical aids that provide support and help in a persons daily life. A robotic home assistant would be performing basic tasks like fetching and carrying the objects like medicines and keys, provide support in holding, lifting things and getting up from a bed or a chair. Instead of replacing care givers we assist them in the daily tasks and improve their care delivery.

Hobbit and Care-O-Bot are the examples for the robotic home assistants. They also are equipped with Media Management, Communication with Medical and Emergency facilities, Supervision of vital signs. These robots can be further improved by adding new applications and integrating additional sensor data, being a mobility aid and supporting communication of elderly people.

### 2.1.4 Open Heart Surgery and Cardiac Rehabilitation

Open Heart Surgery(OHS) is a type of surgery where chest of the patient is cut open and performed on the valve or muscles of the heart[1]. OHS has a lot of potential risks like lung failure, respiratory difficulties, chest pain etc. Taking care of the body after the surgery is very essential for the patient to recover. There are certain institutions such as hospital or a care facility which provide professional and disciplined care to the recovering patients after the surgery.

Cardiac Rehab is a program designed with regulated exercise and education which help patients recovering from the open heart surgery. This Rehabilitation program generally ranges from 4-6 weeks and is even longer in some cases, deals with regulated bodily exercise, pain management and constant help.Rehab period makes sure that a patient is on track to his recovery and reduces risks such as infections and readmissions into the hospital. Access to improper rehab program leads to slow recovery time, depending on additional medicines to handle pain and possible re-admission into the hospital . The readmission rate after a cardiac surgery due to an infection or other risk factors is 18.7%[2]. This means approximately 1 out of 5 patients is in the risk of readmission due to lack of knowledge of proper care. Due to improper handling of self care, increasing population of elderly people and limited care givers it is really hard to provide the qualitative and effective care in the near future. At least 15% of the patients admitted in the care facility complain that their physical needs are unmet or have been ignored by the care givers and 22% people have complained that their are not accurately informed during their stay at the hospital or rehab center[3]. These numbers are going to go up in the near future unless there is an alternative

[1] https://www.healthline.com/health/open-heart-surgery
[2] https://www.ncbi.nlm.nih.gov/pubmed
[3] https://www.ncbi.nlm.nih.gov/pubmed/12102434


Figure 2.1: Example of a Basic Control System

method to provide effective care. Rehabilitation robots might be the answer in the near future.

### 2.1.5 Control Systems

The word Control Systems refers to a framework used to design a system or group of systems whose performance can be controlled, regulated and tracked[6]. Most of our daily use devices are designed with the principle of the control. Basic example of an control system are the traffic lights. They are designed to regulate and track the flow of traffic. Many of the household devices like toaster, oven, washing machine use the principle of control systems.

Fig 2.1 is an example of a simple first order control system.

Traditional Control Systems used Fourier and Laplace transforms for the design of an mechanical system.The calculations were complex and lengthy. Now a days, people use differential equations and state variables to design a system. Control systems are further divided into many parts, but the major categories are open loop and closed loop systems. The major difference between a open and a closed loop systems is the usage of the feedback network.

Figure 2.2 is an example for an open looped system where as the Figure 2.3 is an

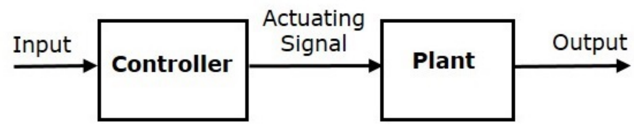



Figure 2.2: Example of a Open Loop Control System

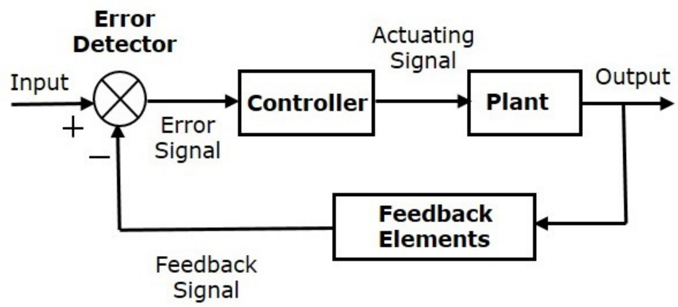



Figure 2.3: Example of a Closed Loop Control System

example for a closed loop system.

The design of controllers which can modify themselves to produce the same output regardless of changes to the input are called as Adaptive controllers. Adaptive control systems are dynamic and change with the requirements and this type of systems are much more useful in the real-world situations.

### 2.1.6 Impedance Control

Impedance control is a active force control which is used to manipulate the mechanical interaction between the environment and manipulator. This type of technique is highly used in robots where the position is of the subject is taken as input and regulated force is desired as an output. Impedance control is suitable for environment interaction and object manipulation tasks. Robot applications, such as versatile use of tools or close cooperation with humans, may be enabled by improved control of mechanical interaction.

It requires a position as input and has a resulting force as output. Impedance control in the robotic hands helps to control the resulting force and can be used to handle even the delicate tasks like stitching wounds and assisting in surgeries.

### 2.1.7 Collaborative Robots

Collaborative Robots (Co-bots) are robots which are specifically designed to work alongside humans without causing them any harm. Major applications of co-bots is seen in manufacturing sector and packaging sector. These robots are deployed across the assembly lines to work alongside the employees. These robots are easily trainable and perfect for the repetitive and ergonomically challenging tasks. Offices, homes, labs, warehouses, farms, distribution centers, hospitals and health care facilities are all enabling service robots to help them do their jobs better.

It is observed that teams made of humans and robots collaborating efficiently can be more productive than teams made of either humans or robots alone. These co-bots are easier to train and deploy than a industrial robot. Few examples of co-bots currently used in industries are Baxter Robot by Rethink Robotics, UR3 by Universal Robots, MRK Systeme .Companies of all types and sizes are finding strategic reasons to acquire or invest in robotics and robotic ventures to add to their arsenal of products and services.

Fig 2.4 shows the Co-bot developed by Universal Robots[4]. UR3 is considered as most flexible table top cobot designed to help and assist human with automation.

[4] https://www.universal-robots.com/products/ur3-robot/

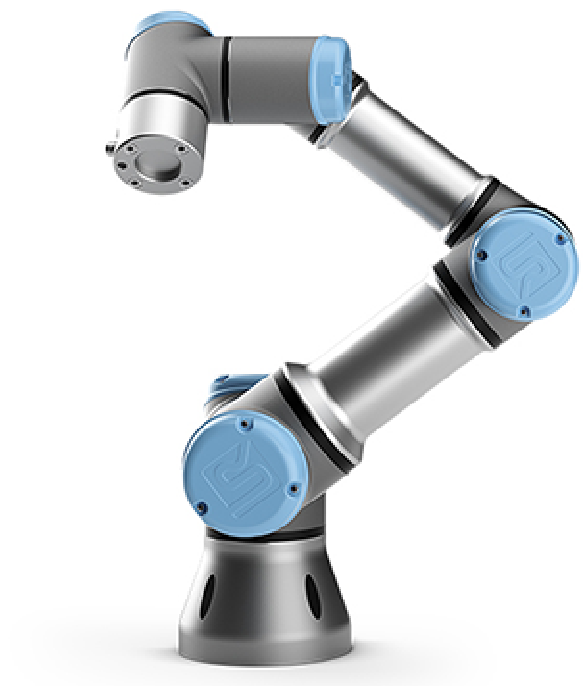


Figure 2.4: UR3 Universal Robot

### 2.1.8 BAXTER

Baxter is a humanoid, anthropomorphic and collaborative robot equipped with torque sensing capabilities at every joint and cameras to support computer vision applications[2]. It also has inbuilt interaction elements such as a head-mounted display, buttons, knobs. It was designed by Rethink Robotics to assist people in manufacturing sector for assembly line jobs. Baxter's Series Elastic Actuators which provide force feedback at each joint, and safety features like collision avoidance which improve handling scenarios with close proximity to humans make it an ideal piece of hardware on which to test different control systems in a controlled setting.

Baxter Research Robot comes with an Software Development Kit(SDK) which enables the researches to develop custom programs or software for the Baxter.The

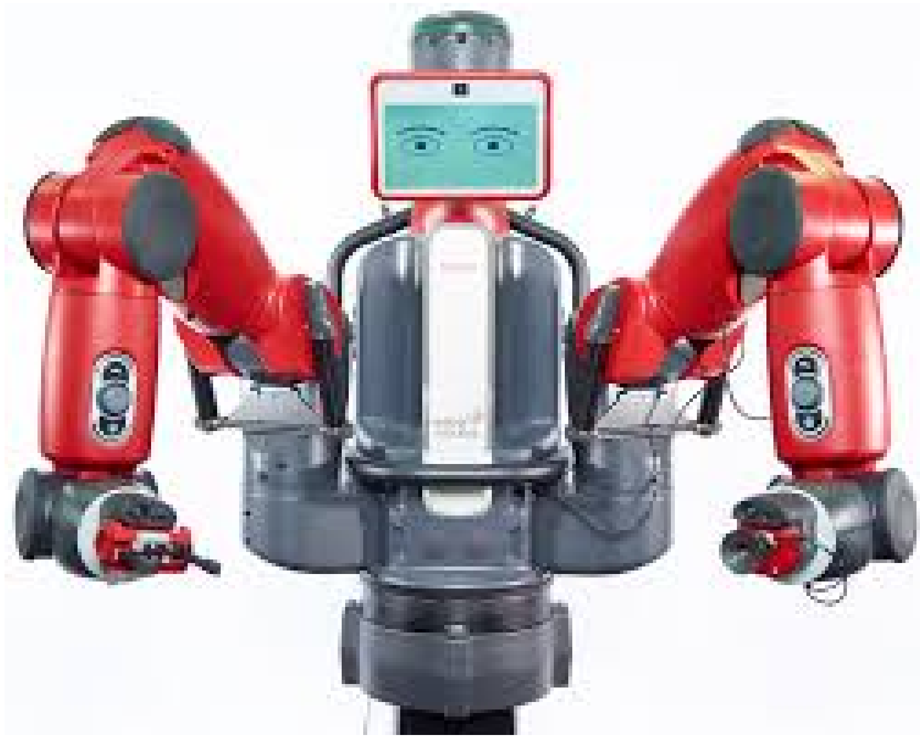


Figure 2.5: Baxter Research Robot
[2]

SDK interfaces with the Baxter Research Robot via ROS (Robot Operating System). It allows the researchers to write and test new custom programs on the robot. With the help of ROS

Baxter comes under a new set of robots famous for their collaborative nature called as Co-bots. Cobots are complex set of robots which are specifically designed to work along hand in hand with people. They range from autonomous robots in the offices to the industrial robots that can be deployed with workers. Baxter is designed in such a way that it can never harm an human and can be completely trained with out the need of an engineer. The belt of sonar sensors around its head along with arms that are power and force limited compliant with series elastic actuators and embedded sensor. These help Baxter to be constantly aware of its surroundings.

### 2.1.9 Robotic Operating System(ROS)

ROS is an open-source operating system for a robot. It provides the services you would expect from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers.

The primary goal of ROS is to support code reuse in robotics research and development. ROS is a distributed framework of processes (aka Nodes) that enables executables to be individually designed and loosely coupled at run time. These processes can be grouped into Packages and Stacks, which can be easily shared and distributed. ROS also supports a federated system of code Repositories that enable collaboration to be distributed as well.

The SDK provides interfaces that control the robot hardware layer via the Robot Operating System (ROS) to any other computers on the robot's network. By using the foundation ROS interface layer, the robot can be controlled and programmed in any Programming Language that supports ROS. The Python Baxter Interface provides a Python Class-based interface library, which wraps many of the ROS interfaces in Python Classes.

### 2.1.10 Pepper

Pepper is an humanoid robot which was designed to perceive human emotions. Pepper is not as robust as Baxter, but has a major advantage to move on its own. Pepper is the first humanoid robot capable of recognizing the principal human emotions and adapting his behavior to the mood of the person with whom she is communicating.

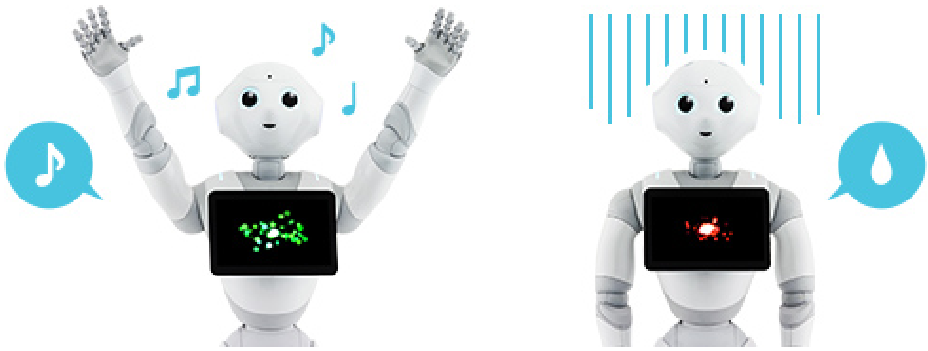


Figure 2.6: Pepper Robot
[16]

As the main ability of pepper is detecting human emotions, it is much more useful in cases where the patients are suffering with emotional dilemma and loneliness. Pepper can move around the house autonomously hence it can also be used to keep track of patients constantly. Pepper is much more of a companion than a care robot, because it can detect your face and wants to learn more about users tastes and habits.It gradually memorizes your personality ,your preferences and adapts himself to your needs.

Pepper comes under a category called Sobot(Social Robot) where as Baxter is a Cobot(Collaborative Robot).

### 2.1.11 Object Detection and OpenCV

Object Detection is a image processing technique used to detect the objects such as person or a vehicle in a given image or a video stream[1]. Histogram of Oriented Gradient (HOG)[7], Scale Invariant Feature Transform[12] and Convolutional Neural Networks[8] are few of the popular image processing techniques that can be used to detect, classify and recognize an object.

Image data sets are used to train and test the model.Training data plays an important role in the performance of the model, better the training data better the accuracy of prediction. Hence annotated datasets are mostly used to train the model to improve the detection accuracy of the model. Generally a bounding box is used to annotate the object in the training data. With the help of this annotations the detection model can isolate the pixel data of the required object and training the classifier becomes more efficient.

OpenCV (Open Source Computer Vision Library) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms[5]. The software library of OpenCV contains about 2500 optimized computer vision and machine learning algorithms[6]. They are used for various tasks ike face detection and recognition, 3d model extraction, human classification in videos etc.

## 2.2 Previous Work

In this section we are going to go through some methods developed in the past which tried to achieve impedance similar to human arm. The following methods are brief description of the models developed in [10] , [20] and [9].

### 2.2.1 Tele-Operance using Electromyography Signals

[10]

Electromyography is a procedure that tracks and records the electric signals in the human muscle tissue. This data is further converted into an signal and is transmitted to the robot to replicate the hand movements.Human arms can easily adapt to the various force interactive situation like handling delicate objects, using tools

[5]https://docs.opencv.org/3.4.2/d1/dfb/intro.html
[6]https://opencv.org/about.html

and cutting down a tree. One possibility of making robots achieve this adaptability is to transfer the nature of human arm flexibility to the robotic arm. This can be done using the EMG signals which can be used to decode the impedance in real time. Advantage of using EMG signal is to eliminate the wait time for the feedback and low cost tracking of these signals. The main objective of this method is to maintaining the control of the robotic arm when it is subjected to an external force. This requires the hand to detect and adapt to the external force which can be attenuated using the EMG signal data.

For this to happen, the authors proposed an tele-operated where the arm is commanded into a position and the robot has to maintain that position using the adaptive impedance control and the EMG signals. The output torque function used to adapt the impedance is given by

$$\tau_u = -t_{ff} - \tau_{fb} - L_\epsilon + \tau_r$$

where $\tau_{ff}$is feed-forward torque, $\tau_u$ is output torque, $\tau_{fb}$ is the feedback torque, $L_\epsilon$ is the stability margin and $\tau_r$ is the torque loss for the robot disturbances.

The experimental setup showed in Figure 2.7 was used to extract the EMG signals and evaluate the impedance control.

After extracting the EMG signals a comparison was made between human operated impedance control vs human like impedance control. Examining the graphs between their stiffness values it is concluded that human operated tele-operance gave better results than the human like impedance.A tele-control disturbance attenuation experiment was designed to compare this improved algorithm with automatic

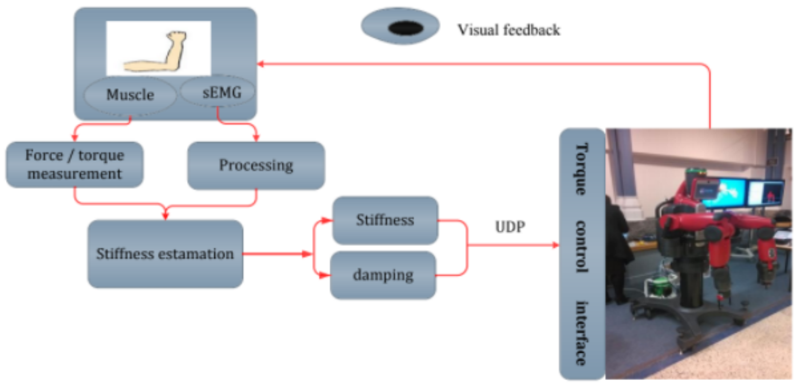



Figure 2.7: Torque function using EMG signals

human-like adaptation of impedance. Results show that the proposed methodology is efficient in tele-operation with human skills transferring to robot in unknown environment. human operated impedance control brings more flexibility than automatic disturbance attenuation, but requires appropriate visual feedback and attention from a human operator.

### 2.2.2 Joint Passivity Control

[20]

Joint passivity control is a framework to control the flexible robot joints. A inner torque feedback loop is incorporated by interpreting the torque feedback as shaping of inertia. Based on motor angles a function is designed incorporating the gravity compensation and Cartesian stiffness. A robot control is said to be successful when

1. Give out the required Cartesian compliant behavior of the manipulator

2. Enable strong and quick adaptation when in contact with unknown environments

3. Safe and dependable activity around humans

The joint torque sensors equipped in the modern robotic arms play an important role to remove damping during the movement and provide safe interaction control. For lightweight robots, passivity based approach is suggested because the only parameters used to change the impedance are the joint angles and torque signals. This framework is highly robust even when the interaction environment is unknown. A full scale feed back controller is designed here by incubating the torque signal feedback and gravity compensation torques.

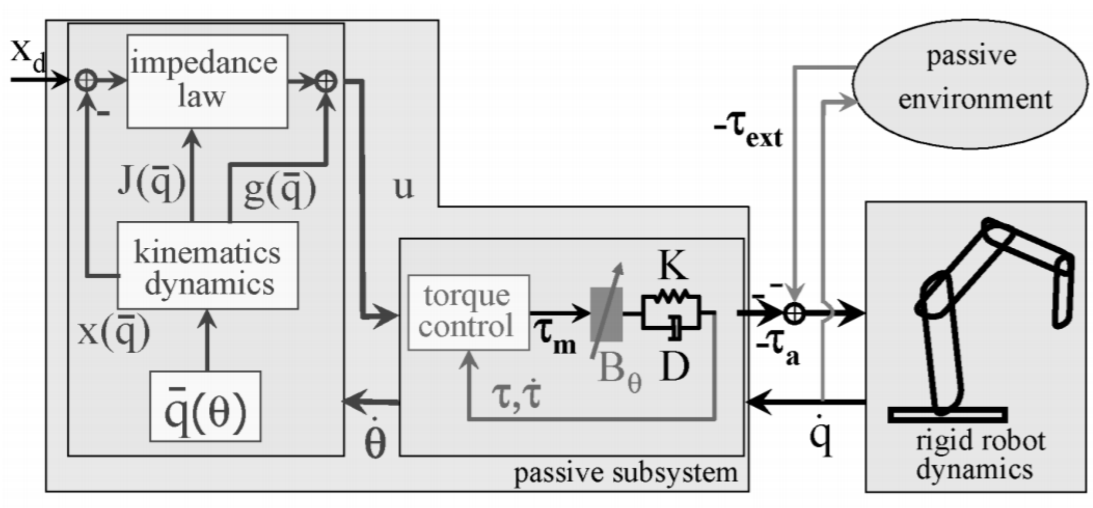
The framework of this model uses the torque feed back as shaping of inertia and motor position as the shaping of the potential energy. Each joint is considered as a fourth order system ad will follow the dynamic model proposed by Richard Spong[14]. The Euler-lagrange equations for a dynamic model is given by

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) &= \boldsymbol{\tau} + \boldsymbol{D}\boldsymbol{K}^{-1}\dot{\boldsymbol{\tau}} + \boldsymbol{\tau}_{\text{ext}} \\ \boldsymbol{B}\ddot{\boldsymbol{\theta}} + \boldsymbol{\tau} + \boldsymbol{D}\boldsymbol{K}^{-1}\dot{\boldsymbol{\tau}} &= \boldsymbol{\tau}_m - \boldsymbol{\tau}_f \\ \boldsymbol{\tau} &= \boldsymbol{K}(\boldsymbol{\theta} - \boldsymbol{q}) \end{aligned}$$

The above equations are the basis of the dynamic modeling for a robotic manipulator of elastic joints. Based on this equations the new feedback controlled based on the torque feedback, joint angles and gravity compensation is given by

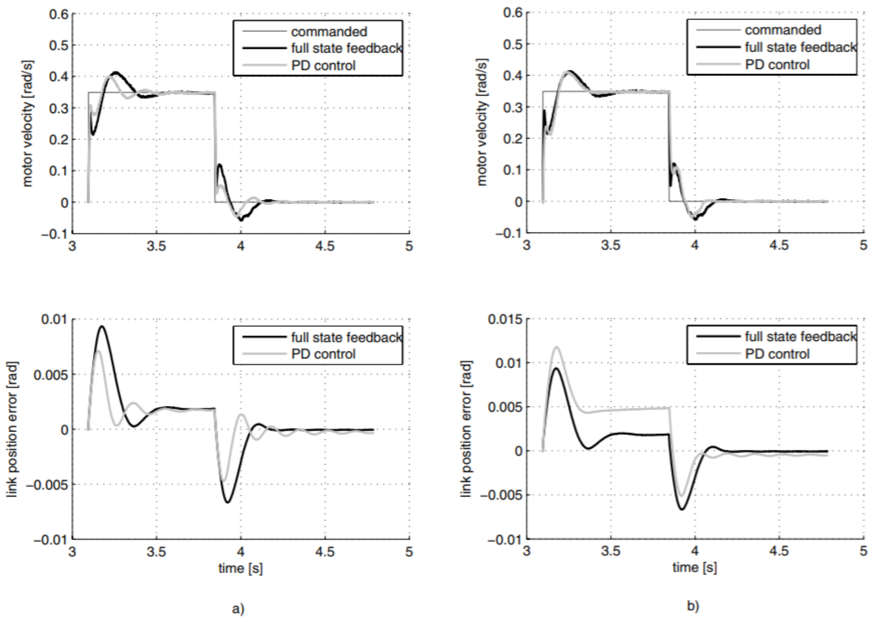
$$\boldsymbol{K}_q = \frac{\partial \boldsymbol{\tau}_{\text{ext}}}{\partial \boldsymbol{q}} = \frac{\partial \boldsymbol{g}(\boldsymbol{q})}{\partial \boldsymbol{q}} + \boldsymbol{K}(\boldsymbol{K} + \boldsymbol{K}_\theta)^{-1}\boldsymbol{K}_\theta,$$

where $K_q$ is the output stiffness, g(q) is gravity vector, K, $k_\theta$ are the controller gains. The function designed above is a position dependent function and the output stiffness is the partial derivative of external torque. The model showed below is used

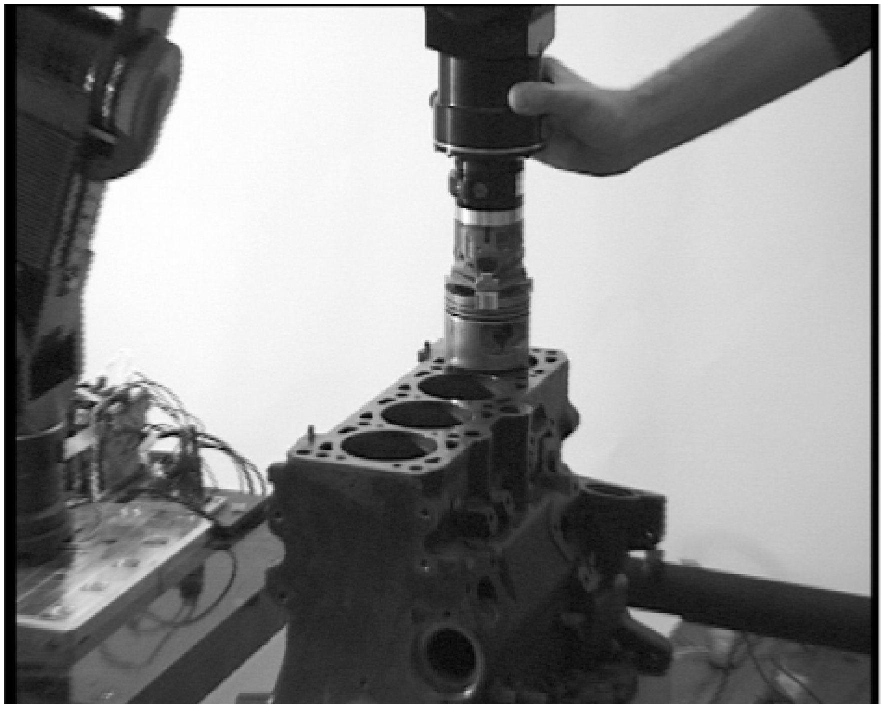



This feedback controller will help regulate the output impedance which in-turn will maintain a safe relation between the manipulator and the end effector.

To evaluate the system the new Position controller(PD) is tested against the state feedback system. The graphs below show the evaluation results.[20]




We observe that PD controller still has higher position error and some oscillations on the torque symbol. Even though the response times are similar the position error is higher in PD controller.

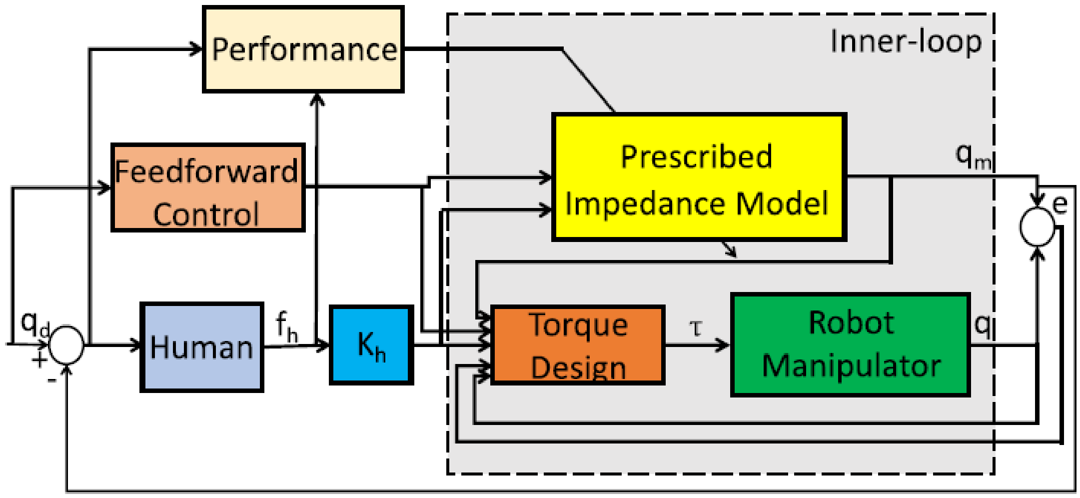


The PD controller was used in some real life applications like Piston insertion, Wiping the table and opening a door. The robotic arm was trained by teaching it how to operate for every task. Each task is split into 2 phases, the teaching phase and Impedance control phase.Piston insertion dealt with the task of teaching and inserting a piston into motor block.Despite a well tuned Cartesian force controller, the insertion process had to be performed much slower, because of the well known control problems which occur in the case of hard contacts with conventional robots. Thus, the advantage of a compliant manipulator with stiffness control in assembly tasks is obvious. Table wiping is a task where the human interaction occurs. Contact with the table went smoothly due to the soft cloth.While opening the door the arm was trained on how to manipulate the door handle by a sequence of impedance controlled movements. Measurements of the joint torques provide an estimate for the contact force and state.

### 2.2.3 Human-Robot Cooperation using Reinforced Learning

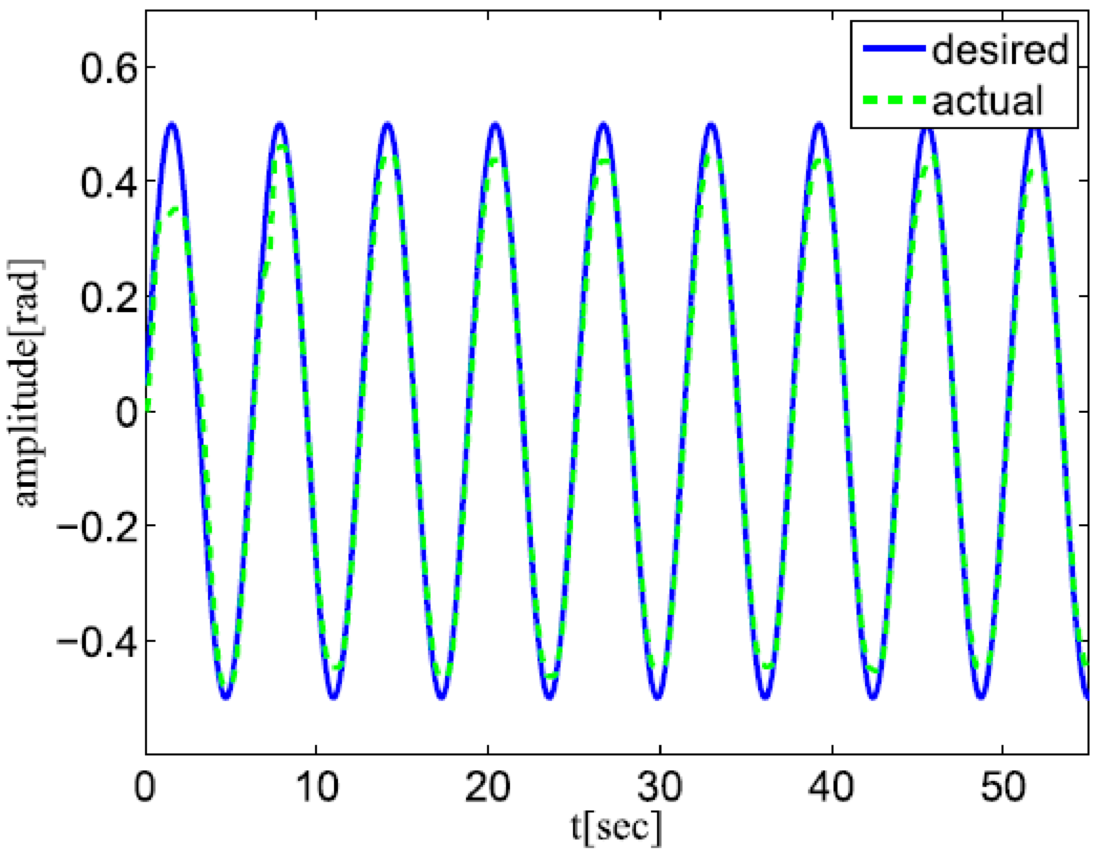
[9]

In this approach a new impedance controller is proposed whose effector is limited by the human movements.A Linear Quadratic Regulation(LQR) with an Integral Reinforcement Learning(IRL) has been proposed to reduce the motion tracking errors and obtain the optimal impedance resembling the human arms.




This model uses a dual loop system where the outer loop is used to control the robot and the inner loop is used to effectively handle the task. Inner loop is for robot orientated control and the Outer loop is for task orientated control. The LQR mechanism was used in the task loop to obtain an adequate impedance to replicate an human arm and then the IRL was used to solve the LQR functions. The inner loop uses an BLF based adaptive control algorithm for robot exoskeleton with the position constraints in place.

This model is tested on 3 different human beings who had no idea of human robot interaction and the trajectories are mapped against the desired trajectories. Extensive testing of these model shows better results than normal impedance controllers as the task specific loop and robot specific loop were used because one loop ignores the parameters of the another loop which separates the designs which in-turn increases the accuracy.




Impedance controllers are being designed with to accommodate fields like Deep Learning and AI. Deep learning methods are used to train the robotic to become fully autonomous without the human intervention.

Our first model uses tele-operance technique where the position of the arm is controlled by the human but the force is adjusted by the impedance controller. Second and third model are more autonomous as they are trained and tested in an unmapped environment and humans. The third model has the novice approach of using task and robot oriented loops to increase the efficiency. At this rate, developing and using of an fully autonomous care robot is not that far away.

# 3 Implementation

## 3.1 Impedance Control

In a dynamic mechanical system the word Impedance is the opposition force acting on the robotic motor joints on its movement. It gives out the relation between the displacement(Kinematics) and Torque(Dynamics). Mechanical impedance is very much similar to the electrical impedance is defined as the ratio of force to the displacement of the system. Manipulating this Impedance in order to regulate the output force and stimulate the environment is called Impedance control.

Impedance control is a framework which is used to indirectly control the force applied by the manipulator(Robotic Arm) on the environment(patient).The position of the environment that needs to be manipulated is the input and the dynamic force reaction is the output.This type of control is widely used in robotic frameworks where there is a need for dynamic and safe interaction between the robot and a human and also because most the time the position of the environment is known.

With advancement in robotics and increase in the availability of sensors,Human Robot Interaction has gained a lot of traction. The scope for a robot to dynamically interact, assist and learn from a human has been improving drastically. With these advancements new type of robots are called Rehabilitation Robots are being developed. The main goal of this robots is to assist recovering patients support human care providers.

## 3.2 Experimentation

Our research deals with assisting the cardiac rehabilitation patients in their daily tasks with minimal human intervention and reduce the load on human care givers. We have chosen a very common activity "getting out of the bed" which causes stress on the upper body of a person recovering from Open Heart Surgery. For this we have to identify the subjects body on the bed and then gently assist them by lifting them up at their own pace without causing any further harm.

For this, we have come up with 2 problem statements

1. Developing a Control Algorithm to control the output torque applied on the patient
2. Identifying the patients position using Baxter's Headboard camera

### 3.2.1 Control Modes and Observations

This section deals with the working of the impedance control algorithm implemented on the Baxter robot. The arm of the Baxter Robot consist of 7 joints from its shoulder to the wrist, The joints labeling is given as S0, S1, E0,E1,W0,W1,W2 where as S0 and S1 represent shoulder joint , E0 and E1 represent elbow joints , W0,W1 and W2 represent wrist joints. All these joints can be manipulated by accessing them using the Software Development kit in the ROS.This arm was designed to work along with humans in a manufacturing sector for picking things up and placing them down. Peak torque payload by the shoulder joints is 50NM[19] and the wrist joints is 15NM[19].

The main component of Baxter's arm used to control and measure force and other parameters is the Serial Elastic Actuators(SEA)[18]. These SEA is the hardware

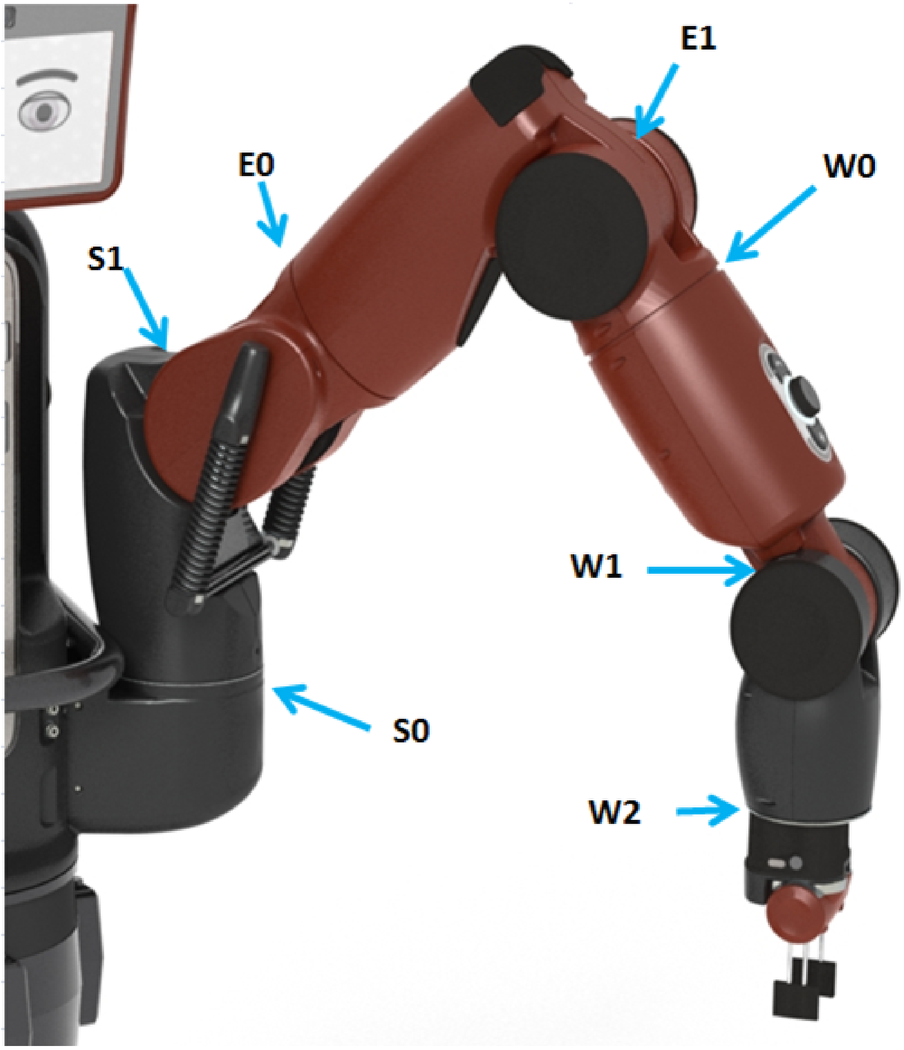



Figure 3.1: Description of Joints in Baxter's Arm.
[19]

which provides elasticity to the robotic joints, reduce friction, act as shock absorber[1] and improve the force accuracy of the robot[2]. This SEA helps in deploying and testing new and different control systems on the Baxter Robot. This SEA introduces a new feature in Baxter called "compliance" which helps the robot in dealing with

[1]https://spectrum.ieee.org/robotics/industrial-robots/rethink-robotics-baxter-robot-factory-worker

[2]https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=525827

unstructured environment by sensing and controlling the forces.[3]

By default Baxter has 4 Control modes[4] which we can access using the ROS. They are Joint Position Control, Joint Velocity Control, Joint Torque Control and RAW Joint Position Control.

Position Control is used to manipulate the position of the arm in the 3-Dimensional space, Velocity Control is used to manipulate the output velocity of the arms and Torque control is to regulate the output torque. To develop a new impedance control algorithm for the robot, we have used the Joint torque control as our base.

**Joint Control Mode**

Joint Control mode is an advanced control mode which allows the user to explicitly modify the torque acting on each and every joint of an arm[4]. Figure 3.2 show the working flow of how the joint torque control works in the robot. The command to modify the torques is sent to the motor controller, where the controller process this request to change the torque. The modified torque is added to the new gravitational compensation forces and then the final torque is sent to the Joint Control Board to enact the changes in the robot torque state.

Also we have to note that in this mode Baxter's collision detection and avoidance are disabled, so the person commanding the toques should be extra careful. Gravity Compensation Vectors(G(q)) are the forces applied on the Baxter's arms to nullify the activity gravitational forces acting on it. This generally removes the loss in forces caused due to gravity.

[3]https://spectrum.ieee.org/robotics/industrial-robots/rethink-robotics-baxter-robot-factory-worker

[4]http://sdk.rethinkrobotics.com/wiki/Arm_Control_Modes

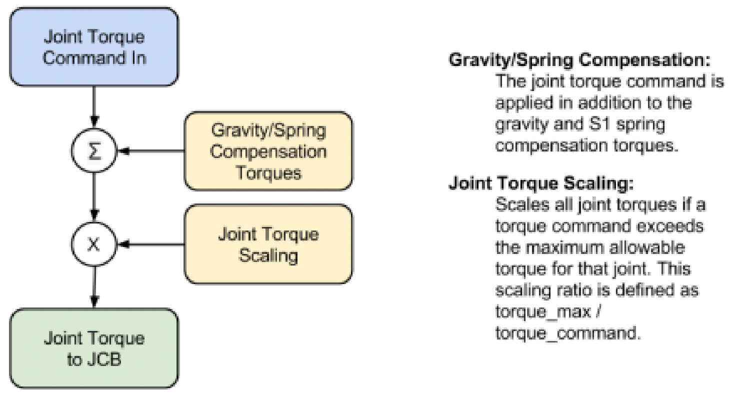



Figure 3.2: Flowchart of Baxter Joint Torque Control.
[4]

Torque scaling in this mode is to make sure the torque given by the user doesn't exceed the maximum payload delivered by the joints.(50NM for shoulder and 15NM for elbow).

### 3.2.2 Impedance Control Algorithm

Our goal in this research was to develop a new algorithm for Baxter so that it could assist people with physical help. We can up with a new impedance control logic which uses the robots internal functionality of the joint control boards to modify the torque on the arm of the Baxter.

We have used Baxter's torque control method as out basic outline to access the joint control boards. We have made some changes to the Baxter's torque control by adding 2 additional feedback loops across the joint control boards. The joint control boards are responsible for calculating the torque values and then SEA moves applies the torques to the arm. The space around Baxter is mapped as a 3 Dimensional co-ordinate space with x,y and z axes and origin located at the center of the robot

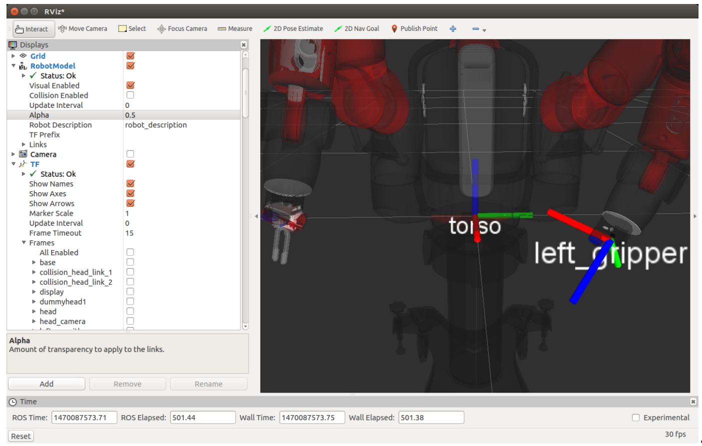



Figure 3.3: Co-ordinate Space around Baxter.[15]

and the stand.

Figure 3.3 shows the origin and the co-ordinate space around the Baxter Robot.

Figure 3.4 show the functioning of control loop using the additional feedback loops for the Baxter robot. The initial torque is represented by $\tau_m$ and the feedback torque is represented by $\tau_{fb}$. $x_f$ represents the position feedback and $tau_o$ represents the final output torque. For a given $x_i$(position co-ordinate), the torque function is

The torque control equation can be expressed as,

$\tau_o = \tau_m + \tau_{fb} + g(q)$

where G(q) is gravity vector with respect to position q and $\tau_k$ is the spring compensation

Once the arm comes into contact with the subject in question, the control loop starts working. The initial torque has been set to a value of 10NM based on the experiments conducted. Once the torque is set, the arm tries to move forward and starts reporting the position vector back to the joint control board. If the position vector stays remains unchanged for 10 cycles(2 seconds;each cycle is set at 0.2 seconds)

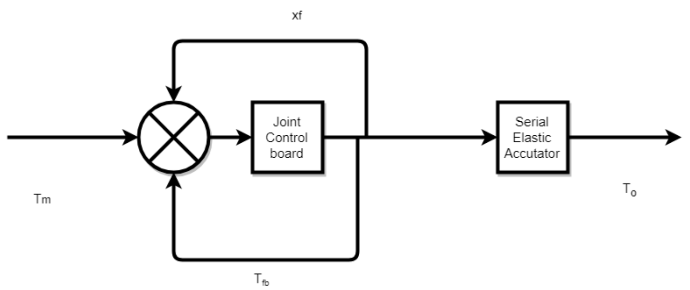



Figure 3.4: Modified Control Diagram

then the torque values are increased by 15% of the feedback torque. This process of checking the position feedback and updating the torque feedback will be repeated until the patient is in upright position.

Using Baxter's SDK, we have implemented the algorithm in python on ROS indigo. We can see the position vector is obtained and compared to its previous vector and then it the pos value is constant then the torque is updated and new torque value is sent to the limb

---

```
pos = self._limb.joint_angles()
          # The joint angles command gives out the
          #current co-ordinates of
          #all the joints in baxter arm
          tor = self._limb.joint_torques()
      #gets the current torque values
          for joint in self._start_angles.keys():
               # checking the current position vector to the
               #previous angles and executes the code
               #below if its true
```

```
            if pos = joint
                cmd[joint] = self._springs[joint] *
                    (self._start_angles[joint] -
                        pos[joint])
                cmd[joint] = cmd[joint] -
                    self._damping[joint] +
                    15/100*tor

        updated_joint_torques =  tor+cmd

    self._limb.set_joint_torques(updated_joint_torques)
```

The above experiments have been carried out on a test subject moving him from point A to point B mimicking the pace and resistance of a person getting out the bed. We have collected the data from the torque sensors during the arm movements to compare and plot the values. In the results section we have included the results of 3 test case scenarios with varying resistance from the test subject. Case 1 with minimal resistance and Case 3 with maximum resistance. During case 3, the robot reaches the maximum output payload and stops moving the patient forward and slowly tries to put him down.

### 3.2.3 Object detection from Baxter's live Feed

In order to help the subject out physically the first task is to locate the subject in question. In this section we explain how we try to identify the position of the patient by using a object detection technique to detect and locate the patients upper body. We have used Baxter's headboard camera as our primary feed and tried to detect the

| Description | Spec |
| --- | --- |
| Max Resolution | 1280 x 800 pixels |
| Effective Resolution | 640 x 400 pixels |
| Frame Rate | 30 frames per second |
| Focal Length | 1.2mm |

Figure 3.5: Baxter Camera Specifications
[3]

person from this live feed.

Object detection algorithm scans the live video feed to detect upper half of the human body. Once the person is found, the robot tries to access the patient using its arms. For the particular object detection task, we have used a model developed by Adam Pollack[17]. The classifier part of the code remains mostly unchanged where as the part of code dealing with Baxter's approach to the object has been changed.

The detection model uses a bag of words approach to identify the specific object from the live video. The process is divided into 4 parts. The first part is feature extraction,To extract the features from the frame of the video, we are using SIFT algorithm through OpenCV. The reason for using SIFT is that it is robust and is invariant to orientation and uniform scaling[12]. Second part deals with visual words and for a bag of words model to work understanding Visual words is very important. To study the features extracted from previous stage, K-means clustering has been used to determine the relations between the vectors. K-means model from scikit learn has been used. After clustering is done, features are quantized and a histogram is created. Then for the final part a SVM classifier is trained on the features extracted and the classifier is tested on the test data.

Our first choice for training and testing data, was a annotated data set called Buffy Stickmen[5]. This data set is made up of 748 annotated images from the TV show Buffy the vampire slayer. For each human in the image, they provide line segments indicating location, size and orientation of six body parts (head, torso, upper/lower right/left arms). But the accuracy using this dataset was not good enough. Hence for this task we created a dataset using the same Baxter's headboard camera. we took around 300 images of the subject and split this data set into training and testing data set. 270 images were used for training the classifier and 30 images were used as the testing data.

After training the classifier, it is interfaced with Baxter using the ROS in python. After the person is detected, the location of the person is estimated using inverse kinematics and the cameras located on Baxter's arms to estimate the position of the subject.

Pseudo code for Position estimation from using Baxter's cameras

---

```
zsafe = 0.00333971663214


    camera_model = image_geometry.PinholeCameraModel()
    camera_model.fromCameraInfo(camera_info)
    gc = GripperClient()

    camera_x = camera_state_info.pose.position.x
    camera_y = camera_state_info.pose.position.y
    camera_z = camera_state_info.pose.position.z
    zoffset = -0.28 # offset on Z axis for Baxter's View
    h = camera_z-zoffset # height from table to camera
    x0 = camera_x # Position of X camera
```

[5]http://www.robots.ox.ac.uk/ vgg/data/stickmen/

```
y0 = camera_y # Position of Y camera
height = 400 # image frame dimensions
width = 640
cx = object_location[0]
cy = object_location[1]


# Convert pixel coordinates to baxter coordinates
xb = (cy - (height/2))*.0025*h + x0 # pixel_size = .0025
yb = (cx - (width/2))*.0025*h + y0
```

## 3.3 Programs implemented on Baxter

### 3.3.1 Rethink Robotics Copyright Notice

UTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### 3.3.2 Impedance Control

```
import argparse
import sys

import rospy

import baxter_interface

class newtorque(self):

    #Function to move the arms back to neutral pose
    #after task execution
    def neutral_pose(self):

        print("neutral pose")
        self._control_arm.move_to_neutral()
        self._puppet_arm.move_to_neutral()
    #Function to shutdown the impedance control mode
    def shutdown(self):

        self._reset_control_modes()
        self.set_neutral()
        return True
    #Function to update the joint dampning loss and
    #spring compensation values
    def _update_parameters(self):
```

```
        for joint in self._limb.joint_names():
            self._springs[joint] = self._dyn.config[joint[-2:]]
            self._damping[joint] = self._dyn.config[joint[-2:]]

    def _update_forces(self):

            self._update_parameters()
            pos = self._limb.joint_angles()
            #  The joint angles command gives out the current
            #  co-ordinates of all the joiints in baxter arm


            tor = self._limb.joint_torques()
        #gets the current torque values
            for joint in self._start_angles.keys():
                if pos = joint # checks the current position
              #vector to the previous angles and executes
              #the code below if it does.
                    cmd[joint] = self._springs[joint] *
                (self._start_angles[joint] -
                        pos[joint]) # manipulates the
                                            # spring stiffness

                    cmd[joint] = cmd[joint] - self._damping[joint] +
                15/100*tor #maipulates the damping co-efficient

                updated_joint_torques = - tor+cmd

        self._limb.set_joint_torques(updated_joint_torques)

   def __init__(self, limb, reconfig_server):

                #refresh rate set to 0.2 seconds
        self._rate = 200.0
        #Time limit for time out set to 20 secs
        self._missed_cmds = 20.0
        #accessing Baxter's limbs
        self._limb = baxter_interface.Limb(limb)
        self._init_state = self._rs.state().enabled

def main():
```

```
    arg_fmt = argparse.RawDescriptionHelpFormatter
    parser = argparse.ArgumentParser(formatter_class=arg_fmt,
                                     description=main.__doc__)
    parser.add_argument(
        '-r', '--limb', dest='limb', required=True )
    js = JointSprings(args.limb, dynamic_cfg_srv)
    control_rate = rospy.Rate(self._rate)
```

### 3.3.3 Feature extraction using SIFT

The following code has been obtained from [17].

```
 #!/usr/local/bin/python2.7
import cv2
import numpy as np
from sklearn.externals import joblib
from scipy.cluster.vq import vq, kmeans, whiten

# Load the classifier, class names,
#scaler, number of clusters and vocabulary
classifier, class_names, std_slr, k, vocabulary =
        joblib.load("trained_variables.pkl")

cap = cv2.VideoCapture(0)

# Create SIFT object
sift = cv2.SIFT()

# Variables for filtering results
confidences = [0,0,0,0,0]
counter = 0
needResizing = False
while(True):
    # Used for filtering results
    if counter > 4:
        counter = 0

    # Capture frame-by-frame
```

```python
    ret, frame = cap.read()

    # descriptor storage list
    descriptor_list = []
    kp, des = sift.detectAndCompute(frame, None)

    # To draw SIFT points
    frame = cv2.drawKeypoints(frame, kp,
            flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
    descriptor_list.append(('curFrame', des))

    # Checking that there are elements in the descriptor_list
    if descriptor_list[0][1] is not None and len(kp) > 15:
        test_features = np.zeros((1, k), "float32")
        words, distance = vq(whiten(
                descriptor_list[0][1]), vocabulary)
        for w in words:
            if w >= 0 and w < 100:
                test_features[0][w] += 1

        # Feature Scaling
        test_features = std_slr.transform(test_features)

        # predictions based on classifier (more than 2)
        predictions = [class_names[i] for i in
                classifier.predict(test_features)]

        # Add label to half of the image
        font = cv2.FONT_HERSHEY_SIMPLEX
        cv2.putText(frame, predictions[0], (225,100),
                        font, 1, (255,255,255), 2)

    if needResizing:
        frame = cv2.resize(frame, (1920, 1080),
                        interpolation = cv2.INTER_CUBIC)
    cv2.imshow('frame', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Stop capturing and close
cap.release()
cv2.destroyAllWindows()
```

# 4 Results

## 4.1 Results from control Algorithm

In this chapter, we discuss the results obtained by running the control algorithm. I have tested the control method on myself and mapped out the results during the process. I have collected the changes in torque values while the arm tried to move my body from point A to point B. We have collected results in 3 cases of subject's behavior in order to analyze the performance of the control loop under varied levels of resistance.

1. Case 1: Minimum Resistance by the subject.
2. Case 2: Moderate Resistance by the subject.
3. Case 3: High Resistance by the subject.

### 4.1.1 Case 1

In case 1, the subject exhibits minimum resistance to arm movement and goes along with the robot to move from point A to point B. In this case we see an increase in torque until the subject starts moving and then the torque value is saturated ones he starts moving freely.

In this graph X axis is the time in seconds once the program starts working and Y axis is the values of torque obtained from the arm. Once the arm applies 25 units

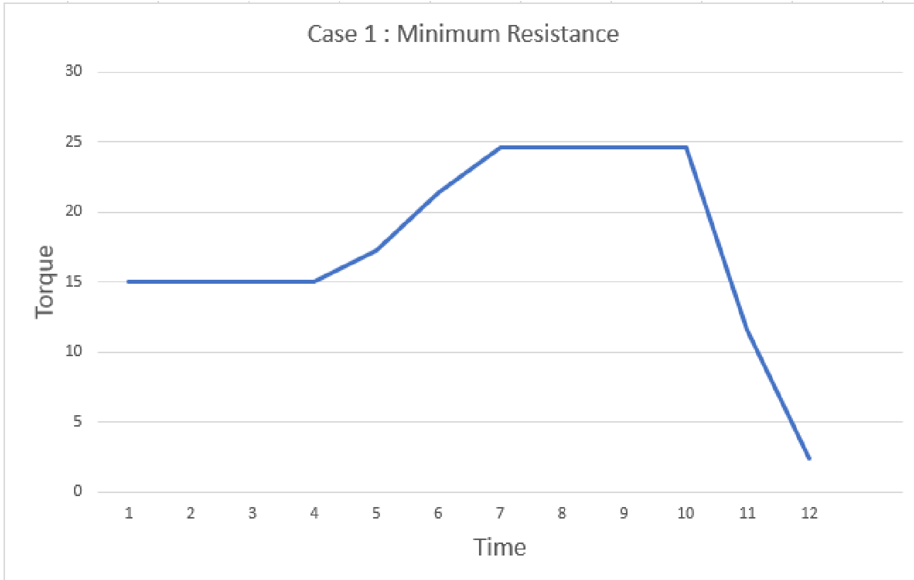



Figure 4.1: Torque values plotted against time

of force, the subject starts moving in pace with the hand hence it does not increase its torque value.

### 4.1.2 Case 2

In case 2, the subject is not freely moving but he exhibits some resistance when the arms comes in contact with him. So it takes more time and energy to push the test subject from point A to point B.

Figure 4.2 shows the torque plot when subject applies moderate resistance.Here we can see the torque output almost doubled from case 1 and it took longer for the arm to reach from point A to point B. This is because the arm couldn't move the body due to increased resistance.

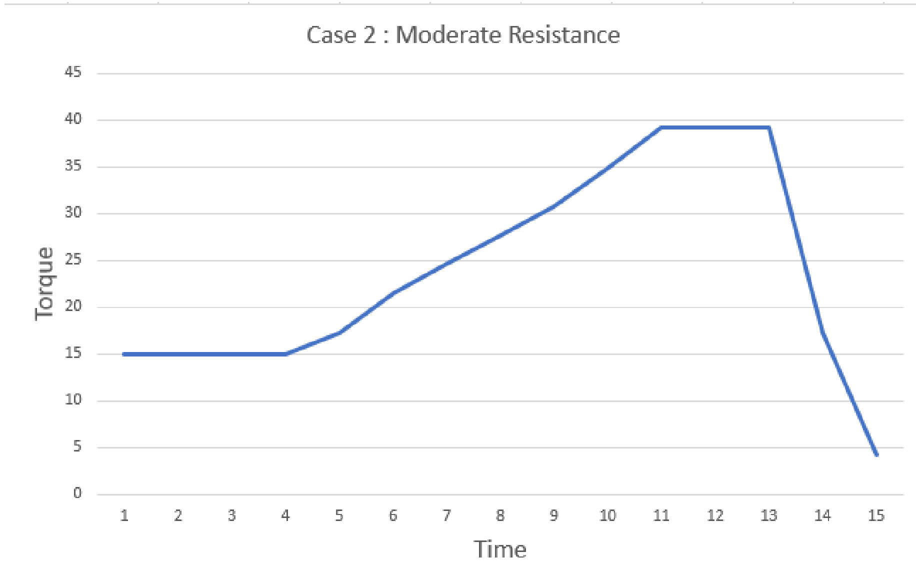



Figure 4.2: Torque values plotted against time

### 4.1.3 Case 3

In case 3, when the test subject shows high resistance to the movement of the robotic arm when it tries to move him.This makes the output payload greater than 50NM and the hand comes to a halt because of torque scaling.

### 4.1.4 Sample Torque Outputs

Sample output show the output obtained from the Baxter when the impedance control mode is on. We map the torque forces in Y axis as the test subject is being pushed across y axis.

header:

seq: 850382

stamp:

secs: 1512500470

nsecs: 852727520

frame_id: ”

pose:

position:

x: 1.19536012266

y: 0.00138443875258

z: 0.214595042714

orientation:

x: 0.503110674444

y: 0.629223930397

z: 0.465920059776

w: 0.365889863994

twist:

linear: x: -0.0111982931996

y: 0.0578988416267

z: -0.00341872668224

angular:

x: -0.0466875541786

y: -0.0164600658146

z: 0.0439125883894

wrench:

force:

x: -6.63067245483

y: -5.25642538071

z: -2.35385274887

torque:

x: -1.42629623413

y: 23.8029937744

z: -7.79895877838

### 4.1.5 Sample position outputs with respect to joint

header:

seq: 285836

stamp:

secs: 1512500858

nsecs: 728039513

frame_id: base

name: $['right_s0','right_s1','right_e0','right_e1','right_w0','right_w1','right_w2']$

commanded_position: $[0.8782040010643993, -0.2814854745769734, 0.39078160571380915, 0.6316165894118191, 1.2137622984143337, -0.051004861197190006, -0.006902913545484362]$

commanded_velocity: $[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]$

commanded_acceleration: $[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]$

commanded_effort: $[5.969435432801388, -0.32529463499784955, 5.729331806449507, -0.192007654523781, 0.03804761833906041, 0.3688865621657698, 0.18826320069822786]$

actual_position: $[0.884339924215941, -0.2814854745769734, 0.39576704327443674, 0.6331505701997046, 1.2106943368385628, -0.05445631796993219]$

actual_velocity: $[0.05654866775400875, 0.01531526418337737, 0.05183627877450802, 0.011388273367126762, -0.056941366835633805, -0.12134401622214377, -0.006675884387626033]$

actual_effort: $[6.108, -22.836, 9.46, -15.28, -0.072, 0.092, -0.008]$

gravity_model_effort: $[-7.070846431234009e-16, -21.328552835580012,$
$5.531813728260596, -15.371705580615043, 0.18401881471378354,$
$0.09406816845290024, 0.0259660035937445]$

gravity_only: $[-7.070846431234009e-16, -54.1775276186751,$
$5.531813728260596, -15.371705580615043, 0.18401881471378354,$
$0.09406816845290024, 0.0259660035937445]$

hysteresis_model_effort: $[0.0, -32.04505504616459, 0.0, 0.0, 0.0, 0.0, 0.0]$

crosstalk_model_effort: $[0.0, 0.8039197369304952, 0.0, 0.0, 0.0, 0.0, 0.0]$

hystState: -0.00273225146709

## 4.2 Results from Object recognition

After training the classifier, we tried the image detection technique on me multiple times. Out of 20 scenarios the hand was able to reach me approximately 14 times, giving out an accuracy of 70%. This accuracy can be further improved by using a CNN for model and better camera than Baxter's.

The camera in the Baxter's headboard provides the live feed to the Object detection model which in turn detects and estimates the position of the subject and tries to reach the Upper half of the subjects body. The position estimation can be further improved by training the robotic arm and using Reinforced Learning methods. The accuracy calculated above is the correct interactions where the Baxter detects my position and reaches me.

Figure 4.3: Object Detection Output

Figure 4.3 shows the snapshot of Baxter trying to reach me

# 5 Conclusions and Future Work

Autonomous Physical Rehabilitation Robots are the next goal in the Human-Robot Interaction. With the the development of the Collaborative Robots Rethink Robotics and many other companies have made it possible working beside and along with a robot with many safety features enabled.

In our thesis we tried to come up with a novice approach of using impedance control to help the OHS patients. We were able to deploy an impedance controller with a position and a torque feedbacks on Baxter Research Robot to assist the person physically and have used a object recognition technique to locate their initial position.

## 5.1 Future Work

The accuracy of object recognition can be further improved by using a Microsoft kinect sensor instead of Baxter's head board camera. Kinect's motion control and body tracking which can perfectly identify the upper half of the human body and keep track of them. The control design can be further improved by using Reinforced learning methods along with the impedance control to further improve the accuracy as the robots gets trained more.

# References


[1] Y. Amit and P. Felzenszwalb. "Object Detection". In: *Computer Vision: A Reference Guide*. Ed. by K. Ikeuchi. Boston, MA: Springer US, 2014, pp. 537–542. ISBN: 978-0-387-31439-6. DOI: 10.1007/978-0-387-31439-6_660. URL: https://doi.org/10.1007/978-0-387-31439-6_660 (cit. on p. 13).

[2] "Baxter Robot". In: (). URL: https://www.rethinkrobotics.com/baxter/ (cit. on pp. 10, 11).

[3] "Baxter Robot Camera Specs". In: (). URL: http://sdk.rethinkrobotics.com/wiki/Hardware_Specifications#Joint_Sensor_Resolution (cit. on p. 32).

[4] "Baxter's Torque control mode". In: (). URL: http://sdk.rethinkrobotics.com/wiki/Arm_Control_Modes#Joint_Torque_Control (cit. on pp. 27, 28).

[5] M. Birks, M. Bodak, J. Barlas, J. Harwood, and M. Pether. "Robotic Seals as Therapeutic Tools in an Aged Care Facility: A Qualitative Study". In: 2016 (Jan. 2016), pp. 1–7 (cit. on p. 1).

[6] "Control Systems". In: (). URL: https://en.wikipedia.org/wiki/Control_system (cit. on p. 7).

[7] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern*

*Recognition (CVPR'05)*. Vol. 1. June 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177 (cit. on p. 13).

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf (cit. on p. 13).

[9] Z. Li, J. Liu, Z. Huang, Y. Peng, H. Pu, and L. Ding. "Adaptive Impedance Control of Human–Robot Cooperation Using Reinforcement Learning". In: *IEEE Transactions on Industrial Electronics* 64.10 (Oct. 2017), pp. 8013–8022. ISSN: 0278-0046. DOI: 10.1109/TIE.2017.2694391 (cit. on pp. 14, 21).

[10] P. Liang, C. Yang, N. Wang, Z. Li, R. Li, and E. Burdet. "Implementation and Test of Human-Operated and Human-Like Adaptive Impedance Controls on Baxter Robot". In: *Advances in Autonomous Robotics Systems: 15th Annual Conference, TAROS 2014, Birmingham, UK, September 1-3, 2014. Proceedings*. Ed. by M. Mistry, A. Leonardis, M. Witkowski, and C. Melhuish. Cham: Springer International Publishing, 2014, pp. 109–119. ISBN: 978-3-319-10401-0. DOI: 10.1007/978-3-319-10401-0_10. URL: https://doi.org/10.1007/978-3-319-10401-0_10 (cit. on p. 14).

[11] D. Linda, W. S. P., G.-B. Jeanne, and M. Carles. "Job Stress and Job Satisfaction: Home Care Workers in a Consumer-Directed Model of Care". In: *Health Services Research* 45.4 (), pp. 922–940. DOI: 10.1111/j.1475-6773.2010.01112.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.

1475-6773.2010.01112.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-6773.2010.01112.x (cit. on p. 5).

[12] D. G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 1573-1405. DOI: 10.1023/B:VISI.0000029664.99615.94. URL: https://doi.org/10.1023/B:VISI.0000029664.99615.94 (cit. on pp. 13, 32).

[13] R. R. Murphy, T. Nomura, A. Billard, and J. L. Burke. "Human–Robot Interaction". In: *IEEE Robotics Automation Magazine* 17.2 (June 2010), pp. 85–89. ISSN: 1070-9932. DOI: 10.1109/MRA.2010.936953 (cit. on p. 3).

[14] S. MW. "Modeling and Control of Elastic Joint Robots." In: (). DOI: 10.1115/1.3143860. URL: http://dx.doi.org/10.1115/1.3143860 (cit. on p. 17).

[15] "Origin and Co-ordinate space around Baxter". In: (). URL: https://groups.google.com/a/rethinkrobotics.com/forum/#!topic/brr-users/yYBjaG2ga5M (cit. on p. 29).

[16] "Pepper Robot". In: (). URL: https://www.softbankrobotics.com/emea/en/robots/pepper (cit. on p. 13).

[17] A. Pollack. "Object Recognition using Baxter". In: (). URL: https://apollack11.github.io/object-recognition.html (cit. on pp. 32, 37).

[18] G. A. Pratt and M. M. Williamson. "Series elastic actuators". In: *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*. Vol. 1. Aug. 1995, 399–406 vol.1. DOI: 10.1109/IROS.1995.525827 (cit. on p. 25).

[19] R. Robotics. "Labeled diagram of Baxter's Arm". In: (). URL: http://sdk.rethinkrobotics.com/wiki/Hardware_Specifications#Joint_Sensor_Resolution (cit. on pp. 25, 26).

[20] A. A. Schaffer, C. Ott, and G. Hirzinger. "A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots". In: *The International Journal of Robotics Research* 26.1 (2007), pp. 23–39. DOI: 10.1177/0278364907073776. eprint: http://dx.doi.org/10.1177/0278364907073776. URL: http://dx.doi.org/10.1177/0278364907073776 (cit. on pp. 14, 16, 18).