User Experience Enchanced Interface ad Controller Design for Human-Robot Interaction

Chen, Junshen

User Experience Enhanced Interface and Controller Design for Human-Robot Interaction

Prifysgol Abertawe
Swansea University

Junshen Chen

Supervised by Professor Chenguang Yang

Submitted to Swansea University in fulfilment of

the requirements for the Degree of

Doctor of Philosophy

in Electronic and Electrical Engineering

Swansea University

July 2019

Abstract

The robotic technologies have been well developed recently in various fields, such as medical services, industrial manufacture and aerospace. Despite their rapid development, how to deal with the uncertain environment during human-robot interactions effectively still remains unresolved. The current artificial intelligence (AI) technology does not support robots to fulfil complex tasks without human's guidance. Thus, teleoperation, which means remotely controlling a robot by a human operator, is indispensable in many scenarios. It is an important and useful tool in research fields. This thesis focuses on the study of designing a user experience (UX) enhanced robot controller, and human-robot interaction interfaces that try providing human operators an immersion perception of teleoperation. Several works have been done to achieve the goal.

First, to control a telerobot smoothly, a customised variable gain control method is proposed where the stiffness of the telerobot varies with the muscle activation level extracted from signals collected by the surface electromyograph(sEMG) devices. Second, two main works are conducted to improve the user-friendliness of the interaction interfaces. One is that force feedback is incorporated into the framework providing operators with haptic feedback to remotely manipulate target objects. Given the high cost of force sensor, in this part of work, a haptic force estimation algorithm is proposed where force sensor is no longer needed. The other main work is developing a visual servo control system, where a stereo camera is mounted on the head of a dual arm robots offering operators real-time working situations. In order to compensate the internal and external uncertainties and accurately track the stereo camera's view angles along planned trajectories, a deterministic learning techniques is utilised, which enables reusing the learnt knowledge before current dynamics changes and thus features increasing the learning efficiency. Third, instead of sending commands to the telerobts by joy-sticks, keyboards or demonstrations, the telerobts are controlled directly by the upper limb motion of the human operator in this thesis. Algorithm that utilised the motion signals from inertial measurement unit (IMU) sensor to captures humans' upper limb motion is designed. The skeleton of the operator is detected by Kinect V2 and then transformed and mapped into the joint positions of the controlled robot arm. In this way, the upper limb motion signals from the operator is able to act as reference trajectories to the telerobts. A more superior neural networks (NN) based trajectory controller is also designed to track the generated reference trajectory. Fourth, to further enhance the human immersion perception of teleoperation, the virtual reality (VR) technique is incorporated such that the operator can make interaction and adjustment of robots easier and more accurate from a robot's perspective.

Comparative experiments have been performed to demonstrate the effectiveness of the proposed design scheme. Tests with human subjects were also carried out for evaluating the interface design.

Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

This thesis is the result of my own investigations, except where otherwise stated. Where correction services have been used, the extent and nature of the correction is clearly marked in a footnote(s). Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Junshen Chen

December 2018

Contents

Acknowledgements

First of all, I would like to express my sincere gratitude to my first supervisor, Prof Chenguang Yang. His patience, motivation and immense knowledge helped me in all the time for research and writing of this thesis. Without his advice, guidance and support, I would not have made any of what I have achieved. I would also like to thank my second supervisor, Dr Christian Griffiths, not only for his insightful comments and encouragement, but also for the hard question which incented me to widen my research visual angle from various perspectives.

I would also like to show my appreciation to Dave and Sue Bickford. They offered me a comfortable place to stay when I was in difficulty. The long-time friendship with irreplaceable memories I had with them is the greatest treasure I have had since I came to the UK. My sincere thanks also go to Dr Zhangfeng Ju, Dr Peidong Liang, and Dr Xinyu Wang for their support and help during my early stage of study in Plymouth. I would also thank Dr Yang Chen, Dr Zheng Cui, Dr Chen Wang, and Dr Runxian Yang for the discussions and all the fun we have had during my time in Swansea. Special thanks to Dr Yali Gao for her encouragement and treatment.

Last but not least, I would like to thank my parents for their unquestioning support.

List of Figures

List of Tables

List of Publications & Awards

1. Yang, C., Chen, J., Ju, Z. and Annamalai, A.S., 2018. Visual servoing of humanoid dual-arm robot with neural learning enhanced skill transferring control. *International Journal of Humanoid Robotics*, 15(02), 1750023.

2. Carter-Davies, D., Chen, J., Yang, C. and Chen, F., 2018. Mechatronic Design and Control of a 3D Printed Low Cost Robotic Upper Limb. In *11th International Workshop on Human-Friendly Robotics (HFR 2018).* **- Best Paper Award in HFR 2018**

3. Oqua, O.I., Chen, J., Annamalai, A. and Yang, C., 2018. 3D Printed Data Glove Design for VR Based Hand Gesture Recognition. In *11th International Workshop on Human-Friendly Robotics (HFR 2018).*

4. Chen, J., Glover, M., Yang, C., Li, C., Li, Z. and Cangelosi, A., 2017. Development of an immersive interface for robot teleoperation. In *Conference Towards Autonomous Robotic Systems (TAROS)*, pp. 1-15.

5. Cousins, M., Yang, C., Chen, J., He, W. and Ju, Z., 2017, July. Development of a mixed reality based interface for human robot interaction. In *Machine Learning and Cybernetics (ICMLC), 2017 International Conference on*, Vol. 1, pp. 27-34.

6. Yang, C., Chen, J. and Chen, F., 2016, September. Neural learning enhanced teleoperation control of Baxter robot using IMU based motion capture. In *Automation and Computing (ICAC), 2016 22nd International Conference on*, pp. 389-394.

7. Chen, J., Glover, M., Li, C. and Yang, C., 2016, August. Development of a user experience enhanced teleoperation approach. In *Advanced Robotics and Mechatronics (ICARM), International Conference on*, pp. 171-177. **- Best Student Paper Award in ICARM 2016**

8. Yang, C., Chen, J. and Cheng, L., 2016, July. Neural learning enhanced teleoperation control of robots with uncertainties. In *9th Int. Conf. Human System Interactions (HSI)*, pp. 223-228. **- Best Paper Award in HSI 2016**

9. Wang, X., Yang, C., Chen, J., Ma, H. and Liu, F., 2015. Obstacle Avoidance for Kinematically Redundant Robot. In *IFAC-PapersOnLine*, 48(28), pp.490-495.

10. Yang, C., Chen, J., Li, Z., He, W. and Su, C.Y., 2015, August. Development of a physiological signals enhanced teleoperation strategy. In *Information and Automation, 2015 IEEE International Conference on*, pp. 13-19. **- Best Conference Paper Award in ICIA 2015**

Acronyms

AI - Artificial Intelligence

ANN - Adaptive Neural Network

APs - Action Potentials

AR - Augmented Reality

CLIK - Closed-Loop Inverse Kinematics

CNS - Central Neural System

DH - Denavit-Hartenberg

DOF - Degree of Freedom

sEMG - Surface Electromyography

FSR - Force Sensing Resister

HRI - Human-Robot Interaction

HSV - Hue-Saturation-Value

IMU - Inertial Measurement Unit

IR - Infrared

LCD - Liquid Crystal Display

LTV - Linear Time-Varying

MDOF - Multi-Degree of Freedom

MUAPs - Motor-Unit Action Potentials

NN - Neural Networks

PbD - Programming by Demonstration

RBF - Radial Basis Fuctions

RMSE - Root Mean Square Error

ROV - Remotely Operated Vehicle

SAD - Sum of Absolute Differences

UX - User Experience

VR - Virtual Reality

VS - Visual Servoing

1 Introduction

With the development of advanced materials, sensors, and actuators, robots are playing an essential role in industrial manufacturing and daily life. There are many industries, such as nuclear energy, deep sea exploration, and space and deep mining, requiring robots to complete complicated tasks under high temperature, high pressure, strong radiation, and some other extreme environments. Given the complexity of tasks and the uncertainty of environment, general robots, including the existing intelligent robots, are powerless. From long-term practice, considering the possibility and limitation of technology and economy, human operators participate in the control and decision-making of robots is currently the most efficient way. Therefore, there has emerged an important branch in the field of robotics - robot teleoperation, also known as telebots.

Robot teleoperation system is essentially an intelligent system of Human-robot Interaction (HRI). As the application environment of telebots becomes more and more extensive, the traditional operation method controlled by operators has been unable to meet the requirements of high-precision operations. A good illustration is to perform work cooperatively with coordinate activities. Generally speaking, the advantage of human lies in the strong ability of perception, decision-making, and planning. Robots have their advantages in achieving stable and high-precision operations.

For an intelligent system, its perception, execution and other functions are distributed at different intelligent levels (such as decision-making level and execution level). Therefore, the problem of HRI is to assign different levels of intelligence between the operator and the robot. The specific assignment depends on the complexity of the task, the complexity of the environment, the operator's skills and the autonomous abilities of the robot. For example, human beings get tired easily and have physical and even psychological limitations because of emotional interference. While, robots can perform complex and accurate calculations rapidly. They can complete repetitive and significantly precise operations. Furthermore, robots have no emotional awareness and are not affected by emotions.

With the development of automation and intelligence of the robot, human-robot collaborative control becomes an important research direction of the robot teleoperation. Therefore, how to harmonise the advantages of human and robots, and combine the strong decision-making ability of human with the high precision operation ability of robots, becomes an important role in improving the user experience (UX) of remote operation of a robot. On the other hand, accurate high-precision operations can be carried out to improve the success rate of completing tasks. Friendly HRI is an important bridge between human and robot, and an important tool to improve the UX while working remotely. The research carried out in friendly HRI is of critical value in both theoretical and application aspects.

There are a variety of control methods and control interfaces for operating multi-degree of freedom (MDOF) robots. New demands for HRI methods emerge as the widespread use of MDOF robots. In order to improve UX for current HRI of MDOF robots, attention should be paid to the challenges stated below:

1) The current remote control methods for MDOF robots requires physical contact. Human operator's motion is easily restrained by hardware devices, such that operator cannot control robot intuitively. It implies the potential chances of reduction in robots' efficiency, and poor performance for telebots.

2) Most of the research on multi-dof robot interaction focuses on master - slave teleoperation and shared teleoperation. The former mainly reflects the human as the dominant, and the robot moves according to the human command. The latter mainly divides the control right to a certain extent, with each part controlled by human and robot. Concerning the current interaction mode, both

master-slave and shared-control robot do not effectively handle assisting tasks. Human operator and robot are hard-wired together physically, resulting in a lack of flexibility in completing tasks.

3) The multi-dof robot exists because it can replace human to reach the environment that human cannot reach, such as high temperature, high pressure and high radiation. In these environments, human beings cannot predict the situation of the environment, which is called environmental uncertainty. Furthermore, most of the tasks that a multi-dof robot needs to complete are high-precision operations, such as bomb removal by an explosive removal robot, assembly and maintenance of a space robot, and operation of a telesurgery robot. This becomes a research challenge for using robots to assist human in high-precision operations under both internal and external uncertainties.

This thesis is a research of designing UX enhanced controller and interface for HRI during teleoperation, including sensor data processing, using artificial intelligence (AI) and computer applications. The research carried out can help to improve UX during teleoperation in both intuitive and immersive ways.

The thesis is organised as follows. The introduction is given in Chapter 1. The motivation and innovation of this research are presented with research challenges. After the introduction, background information is described in Chapter 2, including methods of robot control and HRI interfaces. The robot used in this research, Baxter robot, is introduced, together with its system dynamics. In Chapter 3, we developed a physiological signal enhanced teleoperation strategy. Muscle activation was extracted for improving user's ability to complete delicate task during teleoperation. Chapter 4 studies visual servoing of a humanoid dual-arm robot with neural learning enhanced skill transferring control. A visual servoing (VS) control system is built by using stereo vision. In order to compensate for the effect of uncertain payload and other internal and external uncertainties during teleoperation, a neural networks (NN) controller is designed with knowledge-reuse and skill transfer features. Chapter 5 focused on utilising human operator's motion as reference trajectories and the robot is controlled using NN technique. In Chapter 6, an immersive interface for robot teleoperation has been developed. A virtual reality environment has been designed for controlling a telerobot arm. Extensive tests have been performed with human subjects to evaluate the interface. Finally, Chapter 7 concludes the research works and the contributions of the thesis.

2 Background Research

2.1 Methods of Robot Control

2.1.1 Baxter robot arm

Baxter is a two-arm robot built by Rethink Robotics [3] shown in Fig. 1. It is 5 to 6 feet tall, 306 lbs weight with pedestal and 7 DOFs per arm, and it is equipped with 360 degree sonar for human presence detection and changeable end-effectors. It is used in simple industrial tasks such as loading, unloading, packaging and kitting, machine tending or material handling. Different accessories can be used on Baxter platform such as vacuum cup gripper, electric parallel gripper and mobile pedestal.

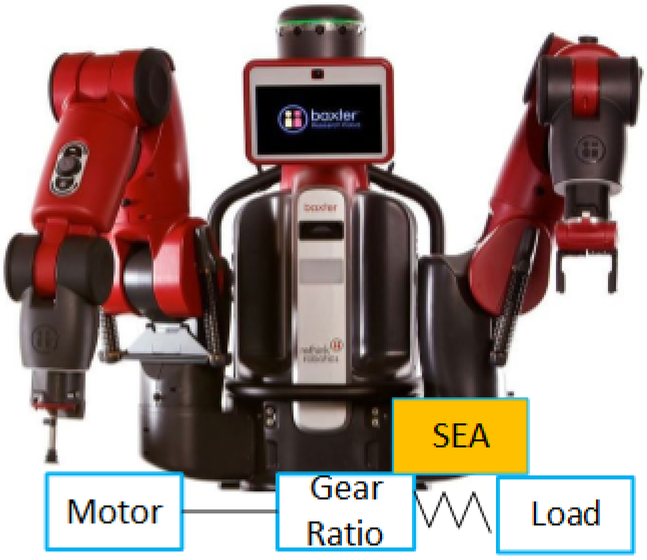


Figure 1: Baxter robot profile [1]: Baxter robot has 7 joints with each arm and each joint is constructed as serial elastic actuator (SEA).

The feature of the Baxter lies in its intrinsic safety. Many existing robots used in industry are built to complete tasks rapidly, most of the components are moving in very fast speed in order to increase efficiency, and lack of safety sensors to reduce costs, these make them not safe for workers to interact with.

Compliant joints are used in Baxter, its arms are driven by springs that driven by motor, this allow Baxter reduce the impact when its arm hit something. Force sensors built in the arms, adjustable torque force and 12 sonar sensors around the head of Baxter can perform detection of human movement and proximity also make Baxter safer to be work nearby.

One another feature of the Baxter is that it is cost effectively. And software engineers are not necessary to use Baxter, because it can be trained to complete its tasks, by moving its arms to required position and letting Baxter to memorise it. Therefore, Baxter is able to work without complicated programming, such that the cost of software development can be reduced [3].

Each Baxter's arm has seven rotational joints and eight links, as well as an interchangeable gripper (such as an electric gripper or vacuum cup) which can be installed at the end of each arm. A head-pan with a screen, located on the top of the torso, can rotate in horizontal plane. Baxter's kinematics model based on Denavit-Hartenberg (DH) parameters has been built in our previous work [4]. The slave Baxter robot has two identical 7-DOF manipulators, as shown in Fig. 2.

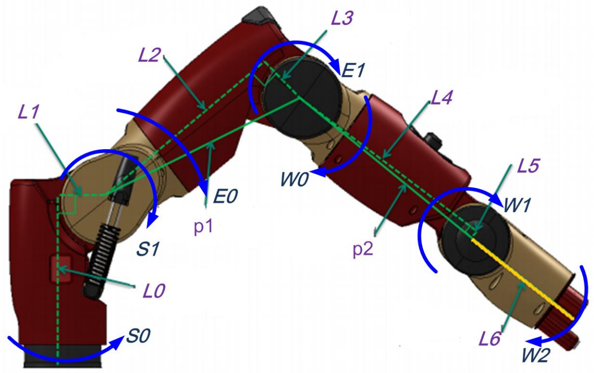


Figure 2: The kinematic model of Baxter's arm

2.1.2 Robot Operating System and Rospy

The Robot Operating System (ROS) is an open source operation system for robot programming. A variety of built-in tools, libraries, and conventions are provided to simplify the task which demands complex and robust robot behaviours to complete.

Rospy is a pure Python client library for ROS. The rospy client API enables Python programmers to quickly interface with ROS Topics, Services, and Parameters. The design of rospy favours implementation speed (i.e. developer time) over runtime performance so that algorithms can be quickly prototyped and tested within ROS. It is also ideal for non-critical-path code, such as configuration and initialisation code. Many of the ROS tools are written in rospy to take advantage of the type introspection capabilities. Many of the ROS tools, such as 'rostopic' and 'rosservice', are built on top of rospy.

2.1.3 Robot System Dynamics

The dynamics of Baxter robot manipulator is described below [5]:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tau_g + G(q) = \tau_u \tag{1}$$

where $D(q)$ is the inertia matrix of the manipulator, $C(q)$ is the Coriolis matrix for the manipulator, $G(q)$ denotes the gravity terms, τ_u denotes the control input and τ_g denotes the payload gravity.

The payload gravity can be described as

$$\tau_g = J^{\mathrm{T}}(q) f_g \tag{2}$$

where $J \in R^{6\times 7}$ is the manipulator's Jacobian matrix, and f_g is the gravity of the payload.

2.1.4 Teleoperation trajectory generation

During the teleoperation, the end effector would not be able to follow exactly the reference trajectory set at the master side. Let us introduce tracking error defined as

$$e_q(t) = q(t) - q^d(t) \tag{3}$$

where $e_q = [e_1, e_2, \cdots, e_n]^T, n = 7$ which equals Baxter robot arm's joint numbers.

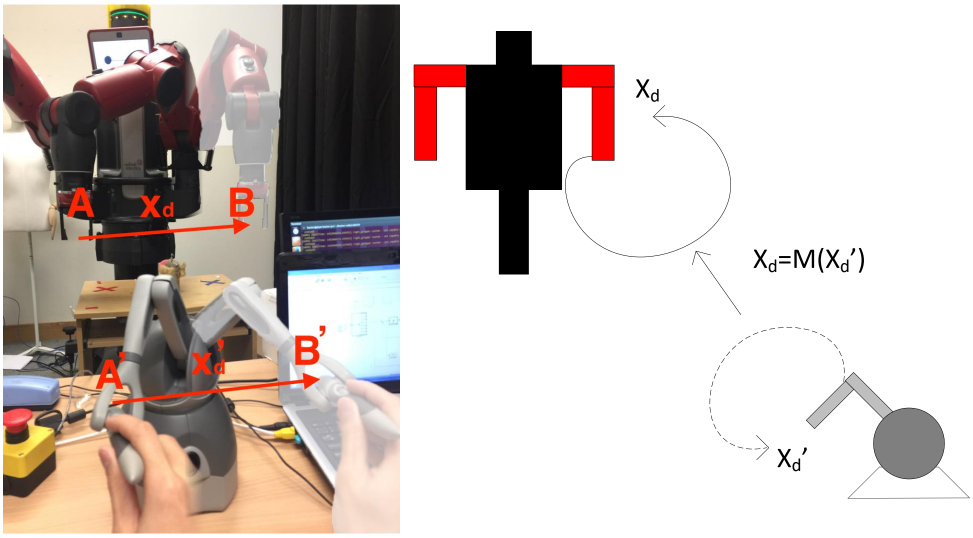


Figure 3: Trajectory and reference trajectory generated by the Omni and the Baxter.

As shown in Fig. 3, $x_d{}'$ is the trajectory generated by the master device and x_d is the end-effector's reference trajectory obtained by applying the transformation on $x_d{}'$. Then, the reference trajectory in joint space, i.e., q_d, can be obtained by using the closed-loop inverse kinematics (CLIK) method on x_d [6]. However, typically there is undesired transient performance such as overshoot or steady-state tracking error, which need to overcome during teleoperation.

2.1.5 User Datagram Protocol

The User Datagram Protocol (UDP) is one of the core members of the Internet protocol suite (the set of network protocols used for the Internet). With UDP, computer applications can send messages, in this case referred to as datagrams, to other hosts on an Internet Protocol (IP) network without prior communications to set up special transmission channels or data paths.

UDP uses a simple transmission model with a minimum of protocol mechanism. It has no handshaking dialogues, and thus exposes any unreliability of the underlying network protocol to the user's program. As this is normally IP over unreliable media, there is no guarantee of delivery, ordering, or duplicate protection. UDP provides checksums for data integrity, and port numbers for addressing different functions at the source and destination of the datagram. UDP is suitable for cases where error checking and correction are either not necessary or performed in the application, avoiding the overhead of such processing at the network interface level. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, according to [7]. If error correction facilities are needed at the network interface level, an application may use the Transmission Control Protocol (TCP) or Stream Control Transmission Protocol (SCTP) which are designed for this purpose.

2.1.6 Neural Networks Control

Neural networks (NN) have been a useful tool for robotic control design to deal with the dynamic uncertainties. It has been proved effective and efficient to approximate the unknown dynamics, for instance, uncertain payloads. However, NN usually does not have the competence to learn the actual system dynamics even during stable closed-loop control. Furthermore, neural learning suffers the the lack of reusing previously learned knowledge, such that computational load may bring high pressure to the processor. How to design a neural learning control, which is able not only to perform storage and reuse of the optimal neural weight values during control, but also guarantee steady-state tracking performance and predefined transient performance, is an important topic to be investigated.

2.1.7 Localized RBF NNs

Radial basis functions have their origins in the study of multivariate approximation theory, particularly in the area of strict multivariable interpolation [8]. By introducing a number of modifications to overcome the restrictions in exact RBF interpolation, RBF neural network models were developed by Broomhead and Lowe [9] and Poggio and Girosi [10], in the late 1980s. In this thesis, RBF NN is used to design the robot teleoperation controller in Chapter 4 and 5.
A linear-in-the-parameter RBF NN can be used to approximate a continuous function, i.e., $\phi(z) : R^m \to R$, over a compact set $\Omega_z \subset \mathbf{R}^m$, can be formulated as

$$\phi(z) = W^{*\mathrm{T}} S(z) + \epsilon_z \ \forall_z \in \Omega_z \tag{4}$$

where $W^* = [w_1^*, w_2^*, \cdots, w_l^*]^{\mathrm{T}} \in R^l$ is the weight vector, $z \in \Omega_z$ is the input vector with $\Omega_z \subset \mathbf{R}^m$ being a compact set, l is the number of NN nodes, and ϵ_z is the approximation error. $S(z) = [S_1(\|z - \mu_1\|), \cdots, S_l(\|z - \mu_l\|)]^T$, is the regressor vector with $S_i(\cdot)$ being a radial basis function, and μ_i $(i = 1, \cdots, l)$ the centre. The Gaussian functions are chosen as

$$S_i(\|z - \mu_i\|) = \exp\left[\frac{-(z - \mu_i)^T (z - \mu_i)}{\varsigma^2}\right] \tag{5}$$

where $\mu_i = [\mu_{i1}, \mu_{i2}, \cdots, \mu_{im}]^T \in R^m$ represents the centre of each receptive field and ς is the variance.

For a large enough number of neurons, there exists a weight matrix W^* such that

$$F(z) = W^{*\mathrm{T}} S(z) + \epsilon(z) \tag{6}$$

where $W^* = [W_1^*, W_2^*, \cdots, W_n^*] \in R^{N \times n}$ is the weight matrix of RBFNN. The estimated weight $\hat{W}$ will be used in practice to replace W^* for the approximation of a continuous function in this manner $F(z) = \hat{W} S(z)$, where $\hat{W}$ is the estimate of W to be specified later in Section 4.7.

The ideal weight vector W^* is defined as the optimal value of W that could minimise the approximation $|\epsilon_z|$ for all $z \in \Omega_z$ as

$$W \overset{\text{def}}{=} \arg \min_{W' \in R^l} \left\{ \sup |h(z) = W^{'\mathrm{T}} S(z) = b| \right\}, \ z \in \Omega_z. \tag{7}$$

When the number of NN nodes l increases, the approximation error $|\epsilon_z|$ reduces. Given sufficiently large node number l and appropriately placed node centres μ_i and variances η_i, it has been proved that the RBFNN can approximate any continuous function arbitrarily close over a compact set. It should be noted that the ideal NN weight W^* is an artificial quantity, which is unknown. In practice, we use estimated weight $\hat{W}$ in replacement of W^* to approximate a continuous target function as

$$\phi(z) \approx \hat{W}^{\mathrm{T}} S(z) \tag{8}$$

where $\hat{W}$ is obtained by a learning law.

2.1.8 Predefined Transient Tracking Performance

To guarantee the desired tracking performance of teleoperation, we now consider shaping the dynamics of the transient response (e.g., overshoot, convergence rate and convergence accuracy) of motion tracking error e_q.

Let us define a performance function γ_i, $i = 1, 2, \ldots, n$ which is a smooth, bounded, strictly positive, and decreasing function with $\lim_{t\to\infty} \gamma_i(t) = \gamma_{i\infty}$. $\gamma_i(t)$ is chosen as [11]

$$\gamma_i(t) = (\gamma_{i0} - \gamma_{i\infty})e_q^{-\kappa_i t} + \gamma_{i\infty} \tag{9}$$

where γ_{i0}, $\gamma_{i\infty}$ and κ_i are positive constants. Then the mathematical expression of predefined tracking performance can be described by equalities

$$-\delta_{1i}\gamma_i(t) < e_{qi}(t) < \delta_{2i}\gamma_i(t), \quad i = 1,\ 2, \cdots, n \tag{10}$$

where δ_{1i}, δ_{2i} are positive design constants. The constants $\delta_{2i}\gamma_{i\infty}$ and $-\delta_{1i}\gamma_{i\infty}$ represent, respectively, the upper bound and low bound of the maximum allowable magnitude of the tracking error e_i at the steady state and they can be designed arbitrarily small. Furthermore, the decreasing rate κ_i of $\gamma_i(t)$ introduces a lower bound on the required convergence rate of e_i, and $-\delta_{1i}\gamma_{i0}$, $\delta_{2i}\gamma_{i0}$ represent the upper bound of the maximum overshoot and the lower bound of undershoot, respectively. Therefore, the desired transient and steady-state tracking performance can be determined by the appropriate selection of the design constants δ_{1i}, δ_{2i}, γ_{i0}, $\gamma_{i\infty}$ and κ_i.

2.2 Methods of Human-robot Interaction

2.2.1 Motion Capture

Many teleoperation applications have been advanced in recent years. In [12], a shared control method of Baxter robot manipulator has been developed. While the user is teleoperating the motion of the end-effector of the manipulator, the robot manipulator can avoid obstacle automatically with the original performance of the end effector motion. A method for imitating human writing skills and transferring them to a Baxter robot manipulator has been present in [13]. By using electromyography (EMG) signals and a haptic device, human operators can teleoperate a robot manipulator doing a fine calligraphy. Physiological studies have shown that our central neural system (CNS) can adapt to muscle force, stiffness and damping to perform different tasks in various environments. However, there has been very little research in the robotics communities to incorporate human motor skills of muscle force, stiffness and damping adaptation into teleoperation. Muscle activities regulated by CNS can be represented by surface electromyography (sEMG) measured by electrodes attached to the skin of human limbs. Therefore, sEMG based tele-impedance control has been recently developed to transfer the variable stiffness from a human operator to a robot for flexible manipulation [14–17]. In [18], a surface electromyography (sEMG) signals enhanced teleoperation strategy has been presented. The human operators can sense the circumstance in a haptic manner and adapt muscle contraction subconsciously as if they are directly interacting with the environment. In [19], it describes the development of a virtual robot teleoperation platform based on hand gesture recognition incorporating visual information.

Motion capture system is one of the teleoperation methods. A motion capture system mainly includes two interfaces, vision system based sensing interface and wearable device/joystick remotely input interface. Several sensors have been used for the vision-based system such as Kinect and Leap Motion. In [20], human motions are captured by a Kinect sensor and the joint angles of Baxter robot has been calculated to teleoperate the robot based on vector approach and inverse kinematics approach. In [21], human welder movement is captured by the Leap Motion sensor, and then the learnt skill has been transferred to a welding robot via a teleoperation system. For the remote input interface, wearable devices such as exoskeleton [22] or joystick such as Omni haptic device [6] are commonly used.

2.2.2 VR and AR User Interface

VR is a technology that presents digitally created information to our senses via various pieces of equipment the combination of which gives the user the feeling that they are in the virtually created environment. It can be achieved using many different types of hardware to cover each of the human senses, e.g. headsets, omnidirectional treadmills, and special gloves. In [23], by wearing a visual headset, operators can intuitively control the pose of a camera on the head of the robot, and the operator can perceive from the robot's perspective. Combining these pieces of technology simulates senses in unison to create the feeling of reality. A VR-based interface was created for Underwater Robots that utilises immersive technologies to reduce user faults and mental fatigue [24].

Augmented Reality (AR) is 'technology to superimpose information on the world we see'. AR is different to VR; instead of using entirely computer-generated environments to let the user thinking they are somewhere else, and immerses them in a new environment, AR just adds more visual information into the environment they are already in. There have been various attempts over the past few years to bring AR to mainstream consumers. For example, a novel AR system was developed for defining virtual obstacles, specifying tool positions, and specifying robot tasks [25]. An augmented reality visualisation interface was presented in [26] to simultaneously present visual and laser sensors information further enhanced by stereoscopic viewing and 3-D graphics.

There have also been applications in entertainment with devices such as the Nintendo 3DS using AR to create games that interacted with the player's surroundings.

2.2.3 Programming by Demonstration

Programming by demonstration (PbD) is a similar technique to the record and replay technique, in which a robot is shown a set of movements and then repeats them exactly multiple times, but PbD has an aspect of learning integrated into it, making it a more effective system. The main aim of programming by demonstration is that the end user of the robot can teach the robot to perform a task without the need for programming [27]. Before this technology became widespread, every function a robot would undertake needed to be meticulously broken down and analysed so a programmer would be able to code in each step individually. In this traditional case, a programmer would have had to programme the robot so that it can safely and correctly response to every possible case within its working environment. To do this effectively, the task has to be broken down into numerous steps, and each step tested thoroughly with many different scenarios to ensure the robot can cope with these changes [27] [28].

In contrast to conventional coding procedure, PbD allows this process to be streamlined by showing the robot its task, while its position, joint rotations and any other required pieces of data are recorded. This allows it to repeat the task by following this data and no coding is required. It is also possible for the robot to learn how to deal with varying circumstances by showing it through multiple different but similar scenarios and the robot will be able to generalise its task [27] [28]. Combining cutting-edge technology, VR, with PbD is an effective way of teaching a robot. In [29], the author created a cleaning robot with virtual agents in a virtual environment. The proposed VR based PbD method can help to reduce the labour, time and cost associated with the interactive learning of robot. A VR approach of learning by imitation and programming by demonstration was also presented in [30].

2.3 Objective and Scope

The general objectives of this thesis are to develop systems including designing different kinds of robot controllers and building user interfaces for teleoperation purposes. For the user interfaces, we will use different types of sensors to study how different level of visual feedback and haptic feedback will effect the user experience during teleoperation. For the controller's design, we will put focus on developing neural network control to increase the autonomy of the telebot. The challenge of compensating both internal and external uncertainties, and reuse the knowledge after learning are both included into consideration.

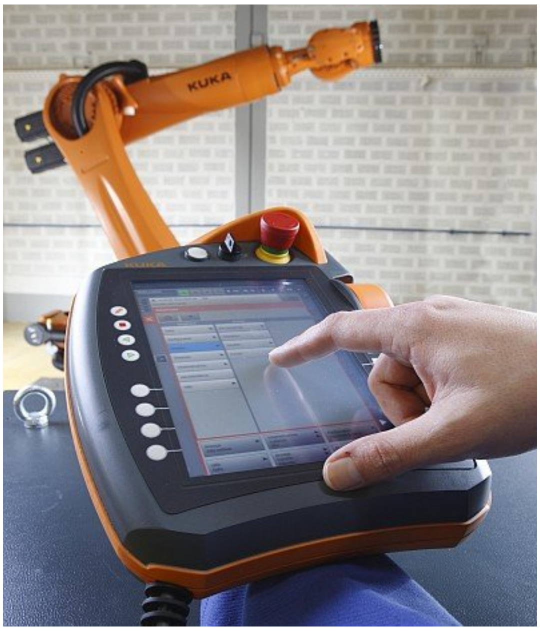


Figure 4: Teaching-pendant programming interface for a KUKA industrial robot.

During teleoperation, the most important challenge is to let human user collaborate with a robot. However, teleoperating the robot through every motion is slow and tiresome, especially on difficult tasks. It is important to take human factors into consideration while designing controllers and user interfaces for teleopration applications. Conventionally, industrial robot were mostly operated by a teaching pendant. In Figure 4, it shows a typical user interface

(teach pad) for human workers to control an industrial robot. A typical teaching pendant has its limitations, such as it only has simple button interface, and it can only leave a certain distance away from the robot due to cable connection. With the call of Industry 4.0, robots have marched into every types on production line. It brings new challenges for creating user friendly interfaces for human operators to operate these machines, especially for unskilled workers.
In this thesis, more immersive and intuitive teleoperation methods has been proposed and investigated, aiming to improve user experience during teleoperation.

3 Development of a Physiological Signals Enhanced Teleoperation Strategy

In this Chapter, we have developed a teleoperation method to control a telerobot manipulator with novel features, such that the human operator could manoeuvre the robot with both physical and physiological signals.

We have developed a personalised variable gain control where the control gain is set according to muscle activation level extracted from sEMG signals, and force feedback is employed so that the human operator is able to sense the circumstance and adapt to muscle contraction subconsciously as if they are directly interacting with the environment.

3.1 Introduction

Telerobots have a wide range of applications in space, subsea exploration and manipulating dangerous and hazardous objects, e.g., disabling a bomb. Nowadays, telerobots have been marching into medical fields, e.g., teleoperated microsurgery and health care, such as tele-assisted rehabilitation for people who suffer from immobility of upper limb [31] due to neurological injuries. In contrast to the automatically controlled manipulator, which can deal with fixed tasks under structure environments, teleoperated robots can work in more complex ones. A teleoperated micro-hand of two rotational fingers was proposed in [32]. A new control method was introduced to control the Engineering Test Satellite VII manipulator [33]. An Internet-based teleoperation system has been developed for robot manipulator Thermo CRS A465 [34].

Most teleoperation systems employ a master-slave framework, in which the necessary feedback information sampled at the remote slave end is transmitted back to the master end, such that the local human operator can manipulate the robot according to these feedback information in real time as if he/she is manipulating the robot in person. As a combination of a local human operator and a remote robot, a telerobot system should merge the intelligence of both robot and operator to achieve an optimal performance [35]. However, most existing techniques ignore this point and fail to consider adapting robot controller in accordance to motor behaviour of each human operator. Despite plausible mathematical models of human motor control built by physiologists [36, 37] and advanced control techniques developed by robotics [38, 39], little effort has been made to combine expertise from both disciplines to develop a matching control technology that perfectly fuses respective advantages of human and a robot.

Studies of human motor skills reveal that human arm can adjust muscle groups co-contraction to yield desired skeleton mechanical impedance during interaction with dynamic environment [40] for the sake of minimising the interaction force and performance errors without endpoint force changes. There also exist selective changes by CNS [41, 42] for activations in individual muscles to generate variable forces for different tasks. Indeed, humans have superior motor capabilities and skills which allows them to effectively modulate their motion/impedance in a smooth and efficient manner according to the task needs. Therefore, it is ideal to transfer human's adaptive impedance and force to a telerobot manipulator of geographical similarity. Many other advantages of this human-robot dynamic transfer can be emerged such as safety, compliant interaction with human and the environment [43].

There are various ways of transferring motion of a human operator to a telerobot using different sensors and techniques, such as motion capture technique based on a visual sensor [44, 45], a joystick, or an exoskeleton device. However, these traditional means only collect physical signals such as position, orientation, force and torque, therefore, these approaches are incapable of capturing the muscle activations or motor skills of human operators. The surface electromyography (sEMG) signals collected by non-invasive electrodes attached on human skins are ideal physiological means to incorporate human skills into robots. They reflect human muscles activations that represent human joint motion, force, and stiffness [46–49]. sEMG signals have been widely used for a robot to understand human motion intention during implementing tasks and have also been recently used in tele-impedance control [16, 43, 50]. The role of impedance regulation in increasing stability, accuracy and task readiness has been illustrated in many research work [40, 41, 51]. Study results have reflected that bringing muscle impedance adaptation into teleoperation provides great flexibility of telerobot manipulation.

Human motor control experiment shows the independent adaptation of force and impedance when interacting with external force field [52]. Our daily life experience also reveals that our dexterous skills in muscle stiffness can adapt in the presence of interactive force. Therefore, the presence of force feedback is important to ensure the natural performance of motor skills, and it is also necessary to bring into tele-impedance control system the haptic interface, which provides operator's tactile feeling of the remote environment through force feedback. It is reported that the introduction of force feedback into teleoperated systems can facilitate the reduction of energy consumption, task completion time and the magnitude of errors [53, 54].

3.2 Teleoperation System Description

In this work, the teleoperation system employs a master-slave control structure. A haptic device (SensAble® haptic Omni) is used at the master end, and the slave system mainly consists of a 7-DOF Baxter robot arm. The position information of the robot hand is sampled and sent to a central processing computer, while the feedback force is exerted on the stylus of the haptic device. In this way, a human operator can operate a distant object according to the haptic feedback and visual feedback. The overall teleoperation structure is illustrated in Fig. 5, where clearly shows that in our proposed method, both physiological signal (sEMG) collected from MYO armband shown in Fig. 6 and physical signal (position) sampled by Omni shown in Fig. 7(a) are integrated to generate commands controlling the robot arm.

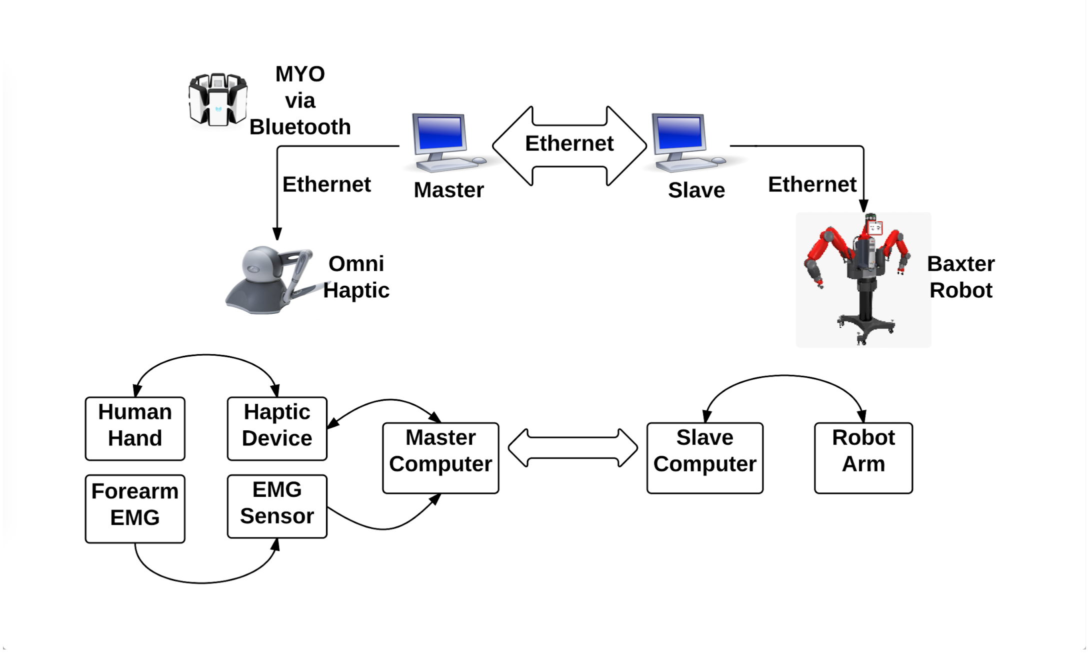


Figure 5: The illustration of the overall system

3.2.1 MYO armband

sEMG signals can be regarded as a linear summation of a compound of motor-unit action potentials (MUAPs). In this work, a wireless 8-channels EMG collecting device MYO is employed with a default sampling frequency, 200Hz.

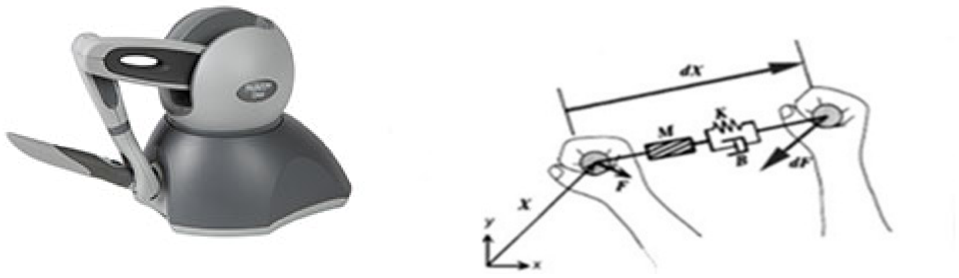
Figure 6: Myo armband

As shown in Fig. 6, the MYO armband can be easily worn in comparison to conventional electrodes. It consists of 8 EMG sensors and 9-axis IMU(Inertial Measurement Unit). The MYO armband needs to be calibrated for each different users before using. It can be done with the official MYO connect app by following the instructions. Also, the size of the MYO armband can be adjusted with clippers provided from the original package to fit the width of each user's arm.

3.2.2 SensAble®Omni

In this work, a SensAble® Omni haptic joystick as shown in Fig. 7(a) is utilised to produce force feedback. This haptic device is of 6 DOFs, in which the first three DOFs contribute to position while the last three form a gimbal contributing to orientation. A stylus equipped with two buttons is also attached to the end-effector. The kinematics of Omni device including forward kinematics, inverse kinematics, and Jacobian matrix has been well studied in the literature [55, 56]. A high-performance 32/64bit driver on MATLAB/Simulink platform has been developed in our previous work [6].

In this section, we incorporate force feedback in the control scheme without using a force sensor for measuring the interaction force. Instead, a haptic rendering algorithm based on the motion tracking error information is employed in



(a) Omni haptic joystick

(b) Force feedback generation

Figure 7: Haptic rendering using Omni joystick.

such a manner that the force feedback is generated using an admittance model as illustrated in Fig 7(b) following the law specified below:

$$F = M(d\ddot{X}) + D(d\dot{X}) + K(dX) \tag{11}$$

where K is the stiffness of a virtual spring, D is the damping ratio of a virtual damper, M is the virtual mass and dX is the difference between Baxter end-effector's actual position and commanded reference position set by Omni joystick. For simplicity, in this work, we only consider 3D translational movement, and $dX = X_s - X_m$, where $X_s \in R^3$ is the salve Baxter robot end-effector's actual translational position and $X_m \in R^3$ is its reference position from the master Omni device.

The feedback force generated by the haptic rendering algorithm Eq. 11 enhances the tracking performance awareness of the human operator, i.e., integrate the tactile sensing with visual sensing. Most importantly, to compensate the feedback force which tends to pull the operator's hand counter the direction of movement towards the lagged robot end effector position, operator's muscle activation is subconsciously enlarged to produce a resistant force. Subsequently, the control gain increases accordingly to the growing EMG signals, such that the robot will speed up to follow the operator's motion.

3.2.3 Workspace Matching

When operating a telerobot which is kinematically dissimilar to the master device, it should be bear in mind that the telerobot manipulator works in its own workspace which might be quite different from that of the master device. Evaluating whether or not a given location is reachable is thus a fundamental problem. Analytical methods can determine closed-form descriptions of workspace boundary, but these methods are usually complicated with nonlinear equations and matrix inversion involved. Numerical methods, on the other hand, are relatively more efficient. Rastegar and Perel [57] introduced the Monte Carlo random sampling numerical method to generate the workspace boundary of some simple manipulators using only forward kinematics. The method is relatively simple to apply and by which we create the workspace mapping model.

The definition of coordinate frame axis of the Omni joystick is different from that of the Baxter, as illustrated in Fig. 8. Thus, the Cartesian coordinate of Omni, $[x_m' y_m' z_m']^T$, needs to be transformed as followings:

$$A_o' = R_z(\frac{\pi}{2})R_x(\frac{\pi}{2})A_o R_y(\frac{\pi}{2})R_z(\frac{\pi}{2}) \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{12}$$

where R_x, R_y, and R_z are the basic rotational matrix, A_o is the transform matrix of the Omni and A_o' is the correspondent modified matrix.

According to the forward kinematics and joint rotate limit of the master device and the slave robot, the Monte Carlo method of random sampling is applied to the joint space of the manipulator to approximate the workspaces of both master and slave [6]. The homogeneous radial distribution was employed to generate 8000 points in the joint space of master and slave separately.

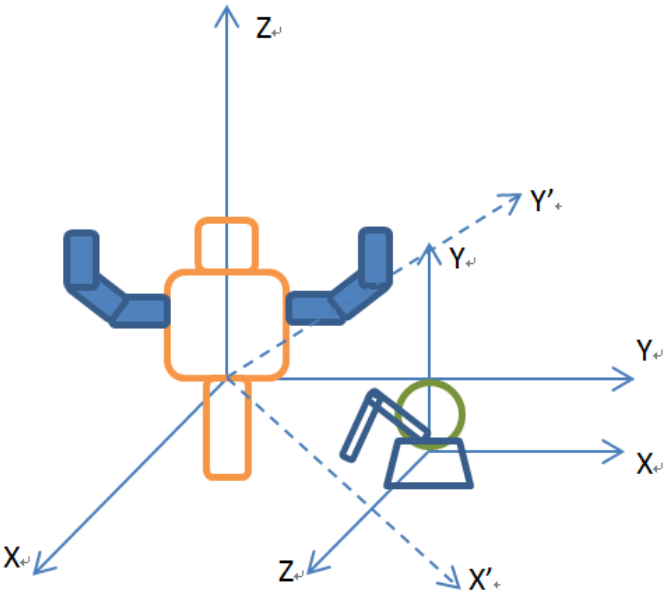


Figure 8: Frame axis direction of the Omni and the Baxter.

To enable the workspace of the master Omni joystick and that of the slave Baxter robot arm overlapping each other as much as possible to improve the manoeuvrability, the point cloud matching method is utilised due to its convenience of considering the position of the end effector instead of its structure. The mapping process can be performed as in equation (13)

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} cos\delta & -sin\delta & 0 \\ sin\delta & cos\delta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \left(\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \right) \tag{13}$$

where $[x_s y_s z_s]^T$, $[x_m y_m z_m]^T$ are the Cartesian coordinate of end effector of the Baxter and the Omni respectively, δ is the revolute angle about Z-axis of the Baxter base frame, $[S_x S_y S_z]^T$ and $[T_x T_y T_z]^T$ are the scaling factor and translation about X, Y, Z axis. According to [6], for the left arm of Baxter robot, the mapping parameters in the equation (13) can be calculated and given as

$$\delta = \frac{\pi}{4}, \quad \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} 0.041 \\ 0.040 \\ 0.041 \end{bmatrix}, \quad \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} 0.701 \\ 0.210 \\ 0.129 \end{bmatrix}$$

The result of workspace matching is represented in 3D space as in Fig. 9. Note that, the direction of the axis of the haptic joystick is different from the axis of telerobot manipulator. With the help of the proposed method, a three axis of the end-effector has been aligned to match the sight of the operator.

In equation (13), only the revolution about Z-axis is done. The master, Omni joystick is placed on a horizontal platform and consequently the Z-axis of master is perpendicular to horizontal plane.

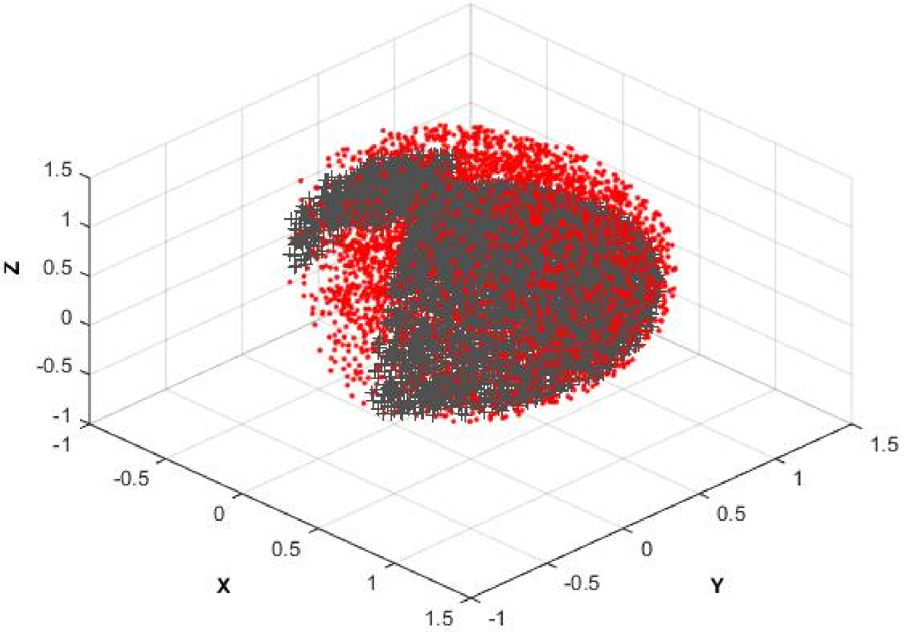


Figure 9: Workspace matching result. The red point cloud is the workspace of slave and grey point cloud represents the master workspace.

Likewise, the slave, Baxter robot is adjusted carefully to make sure that the Z-axis is upward vertical and also perpendicular to horizontal plane. In this way, the Z-axis of master is parallel to Z-axis of slave. The revolution about X-axis and Y-axis are neglected due to a human-in-loop control approach is employed. That is to say, as long as the X-axis and Y-axis of master and slave are alignment approximately, operator can adjust the arm of robot according to the posture depicted on the master computer's screen instead of to control the slave directly use the accurate coordinate value.

It is noted that the aforementioned workspace matching causes motion amplification because the workspace of Baxter robot arm is physically much larger than that of the Omni joystick. This amplification would increase the difficulty of delicate manipulation, e.g., tiny motion error of a user would lead to large undesired motion deviation of the telerobot.

3.3 Extraction of Muscle Activation

The transformation from sEMG to muscle activation is an essential process. How to estimate the influence of the activation level from sEMG in terms of the performance of variable gain control was proposed in this work. Firstly, the raw sEMG from all channels, namely, $u_i(k)$, where k is the current sampling instant and $i = 1, 2, \cdots, N$, are integrated by computing root mean square of the $u_i(k)$, with a similar concept to those proposed in [58, 59] as below:

$$\bar{u}(k) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} u_i^2(k)} \tag{14}$$

Next, the moving average filter is applied as below:

$$\bar{u}_f(k) = \frac{1}{\max\{k, N\}}\sum_{i=k}^{k+M} \bar{u}_f(k) \tag{15}$$

following experimental studies, $M = 20$ is chosen in the experiment to gain the best filtered signals from sEMG data.

The study on the single motor unit shows that the multiple action potentials (APs) cause multiple twitch responses. If the time between APs decreases, the twitches start to merge into each other, and thus the muscle force increases steadily. However, at high frequency, the twitches get closer to tetanus, where no further force is produced even if the frequency increases.

Therefore, a nonlinear mapping between frequency and force for single motor units can be assumed. The nonlinear mapping from neural activation $u(k)$ to muscle activation $a(k)$ proposed in [60] is employed in this work:

$$a(k) = \frac{e^{Au(k)} - 1}{e^{A} - 1} \tag{16}$$

where A is the nonlinear shape factor which is allowed to vary between -3 and

0, with $A = -3$ being highly exponential and $A = 0$ being linear.

3.4 sEMG based Variable Gain Control

3.4.1 Control gain calculation

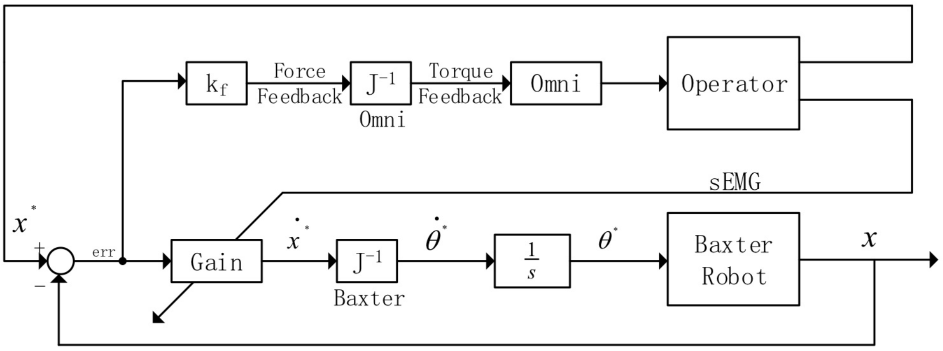
The control gain can be set proportionally to the muscle activation. However, it is of importance to normalise the control gain within the specified range of stable motion. Otherwise, it may cause instability due to incompatibility with human stiffness variations. The following Eq. (17) is used to generate control gain at the kth sampling instant by incorporating specified range:

$$\mathrm{Gain}(k) = (\mathrm{Gain}^{max} - \mathrm{Gain}^{min})\frac{(a(k) - a^{min})}{(a^{max} - a^{min})} + \mathrm{Gain}^{min} \tag{17}$$

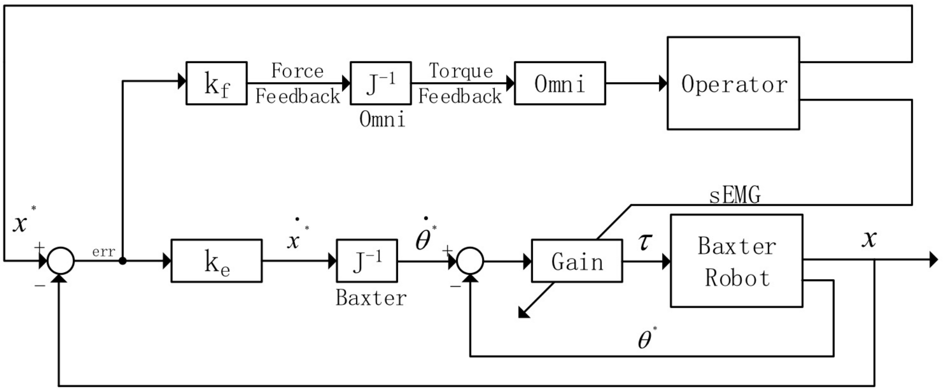
where the gain range of stable robot motion Gain^{max} and Gain^{min} as well as the maximum and minimum muscle activation a^{max} and a^{min} can be obtained experimentally beforehand.

3.4.2 Structure of control implementation

The variable gain control proposed in this work is realised in both position control mode and torque control mode provided by the Baxter robot. The control diagrams are shown in Fig. 10, where x^* is the reference trajectory commanded by the operator, x is the actual trajectory of the robot, J is the Jacobian of the robot arm, and J^{-1} is the pseudo-inverse of J. In position control mode shown in Fig. 10(a), the control gain generated from sEMG reflecting muscle activation directly affect the commanded speed sent to the position controller of the Baxter robot. Therefore, in this control mode, an operator can adjust the moving speed of the robot manipulator by muscle contractions. On the Baxter platform, the position and torque modes are built-in control modes. During the remote control, the operator remotely merely controls the movement of end-



(a) Diagram of position control mode



(b) Diagram of torque control mode

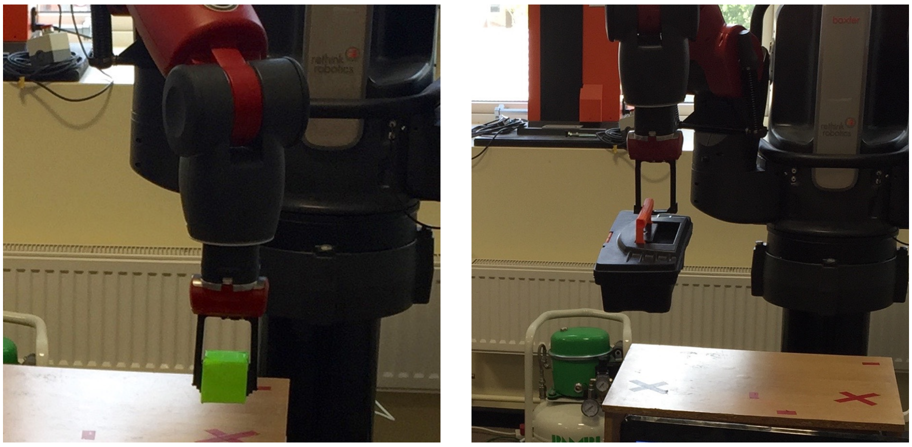
Figure 10: Diagrams of two control modes: position control (a) and torque control (b) modes.

effector providing positions to be tracked, and the target placing position is not known by the arm manipulator. Therefore, the overshoot is brought about mainly due to the inaccurate operations, which relates to the settings of teleoperation system, instead of the selected control mode.

Fig. 10(b) shows the implementation in torque control mode, whereas the specified control gains proportionally affect the stiffness of the controlled robot arm.

3.5 Experimental Studies

Two sets of experiments have been designed and carried out to verify the effectiveness and efficiency of the proposed sEMG enhanced teleoperation system.



(a) Pick-up and Drop Task

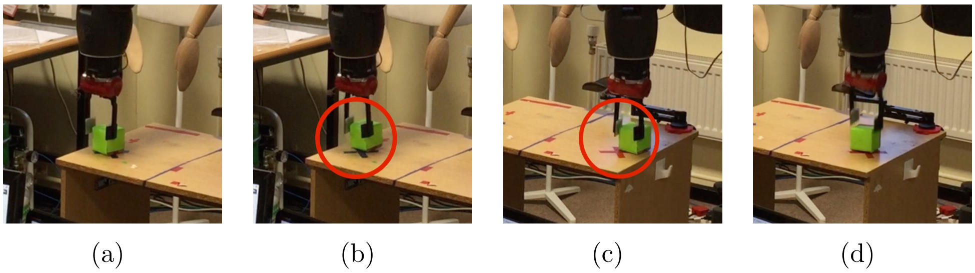
(b) Lift-up and Move Task

Figure 11: Two designed experimental tasks to test the developed variable gain control strategy under both position control mode and torque control mode.

3.5.1 Test in position control mode

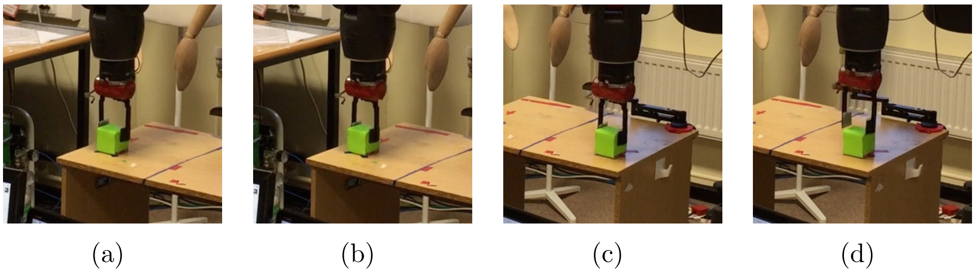
In the position control method, the control gain can be set in order to change the following speed of the telerobot manipulator. A pick-up and drop task was designed to test the performance of the EMG based variable gain control method. In this experiment, the operator is to pick up the target object from a start position labelled with the blue cross on the table to a target position labelled red cross on the table, as shown in Fig. 11(b).

The green object showed in Fig. 11(a) is the object to pick-up and drop. Figs. 12, 13, 14 are captured from the video clips which were recorded during the test. Figs. 12(a), 13(a), 14(a) show the scenarios when the manipulator reached the object. Figs. 12(b), 13(b), 14(b) show the status when the manipulator was gripping the object and starting the movement to the target dropping position, as labelled with the blue cross on the table. Figs. 12(c), 13(c), 14(c)



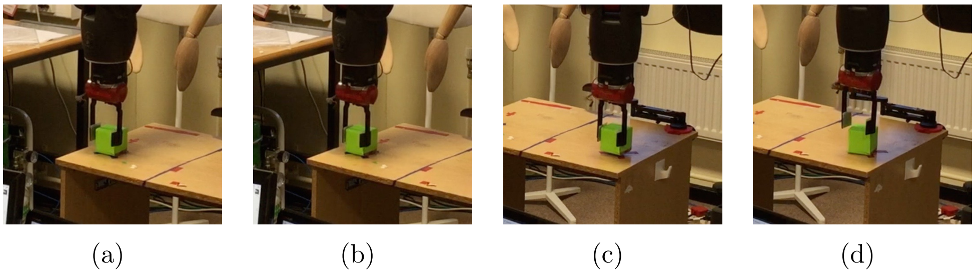
(a) (b) (c) (d)

Figure 12: High gain mode



(a) (b) (c) (d)

Figure 13: Low gain mode

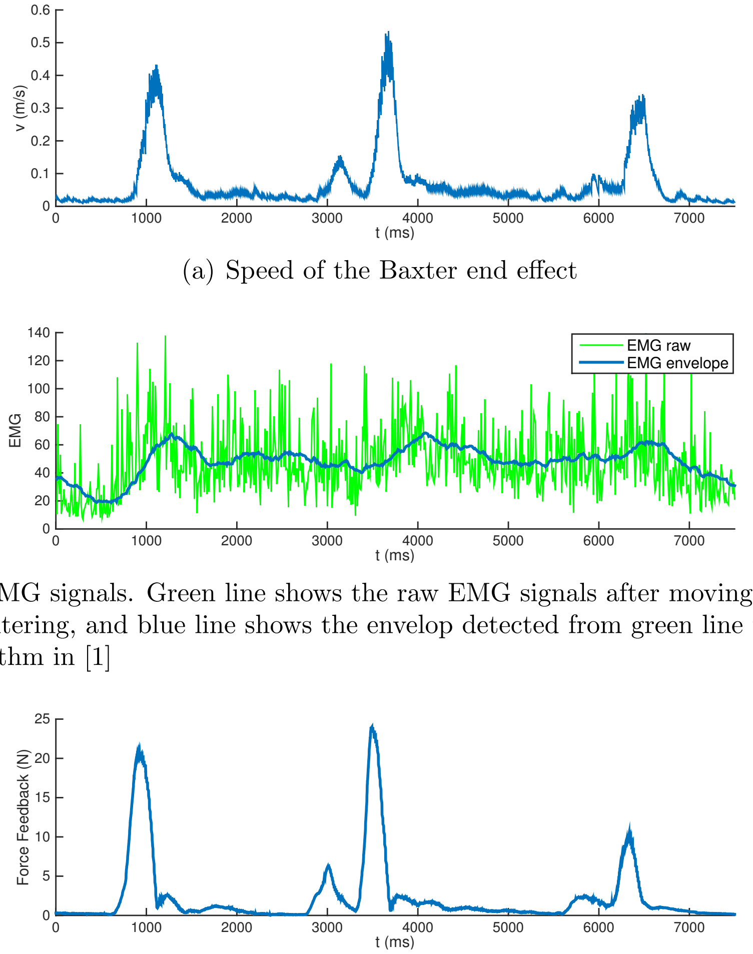


(a) (b) (c) (d)

Figure 14: sEMG based variable gain mode

show the scenarios when the manipulator reached the dropping position. Figs. 12(d), 13(d), 14(d) show the scenarios when the manipulator dropped the object and start moving back to the original position. Operator in both low gain mode and the EMG based variable gain mode could reach and drop the target smoothly and accurately. While under the high gain control mode, it is very hard for an untrained operator to grip the object accurately in the first time, for any inaccurate operation will be rapidly followed and amplified. The red circles in Figs. 12(b), 12(c) illustrate that the object had been placed a distance from the target position, i.e., the red cross labelled on the table, by the manipulator. Generally, motion amplification exists when the workspace of the master device is much smaller than the workspace of the slave robot.

The time consumption of the entire pickup and drop task is relatively low in high gain mode and EMG based variable gain mode, while it is much higher in the low gain mode. Comparative experiments illustrate that the sEMG based variable gain in position control model ensures efficient, smooth and accurate manipulation. Moreover, it has better user experience than the high gain and the low gain modes, especially for the untrained, unskilful operators.



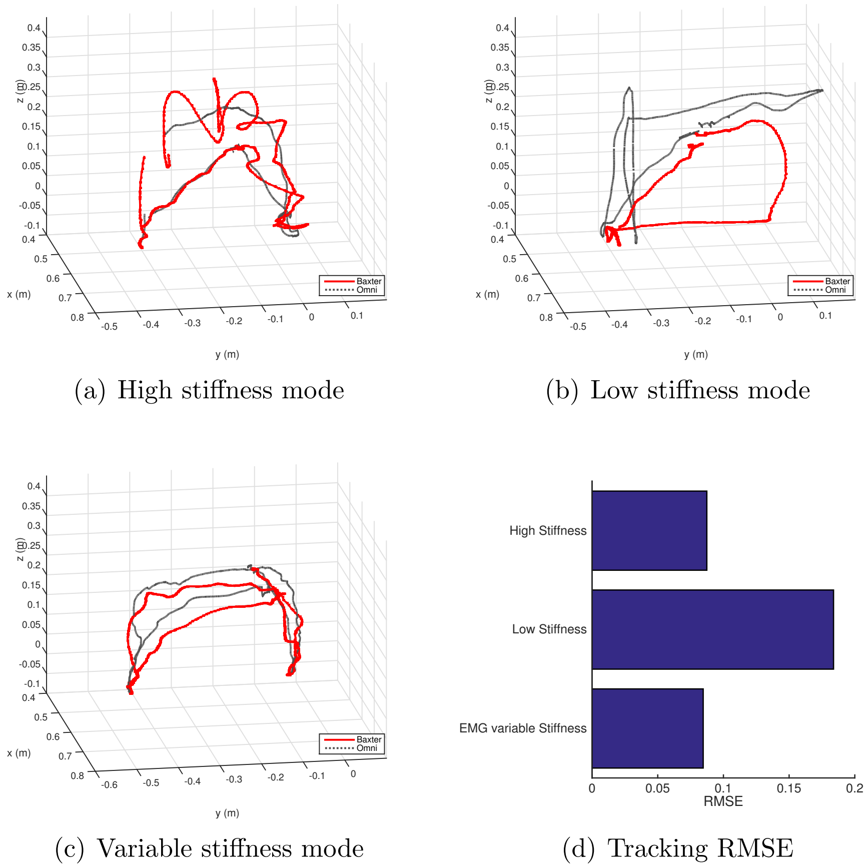
(a) Speed of the Baxter end effect

(b) EMG signals. Green line shows the raw EMG signals after moving average filtering, and blue line shows the envelop detected from green line using algorithm in [1]

(c) Force feedback

Figure 15: Relationship between force feedback and robot tracking speed, under sEMG based variable gain mode

In Fig. 15(a), the blue curve represents the speed of the manipulator's end effect. In Fig. 15(b), the green one represents the raw data of the EMG signals, and the blue curve stands for the time series of muscle action extracted from the EMG signals. Force feedback of the Omni is shown in Fig. 15(c). Fig. 15 demonstrates that when the sEMG based variable gain control mode is selected, the gain of the manipulator's speed will increase when the operator is subject to a large feedback force forced on his/her forearm and vice versa.



(a) High stiffness mode

(b) Low stiffness mode

(c) Variable stiffness mode

(d) Tracking RMSE

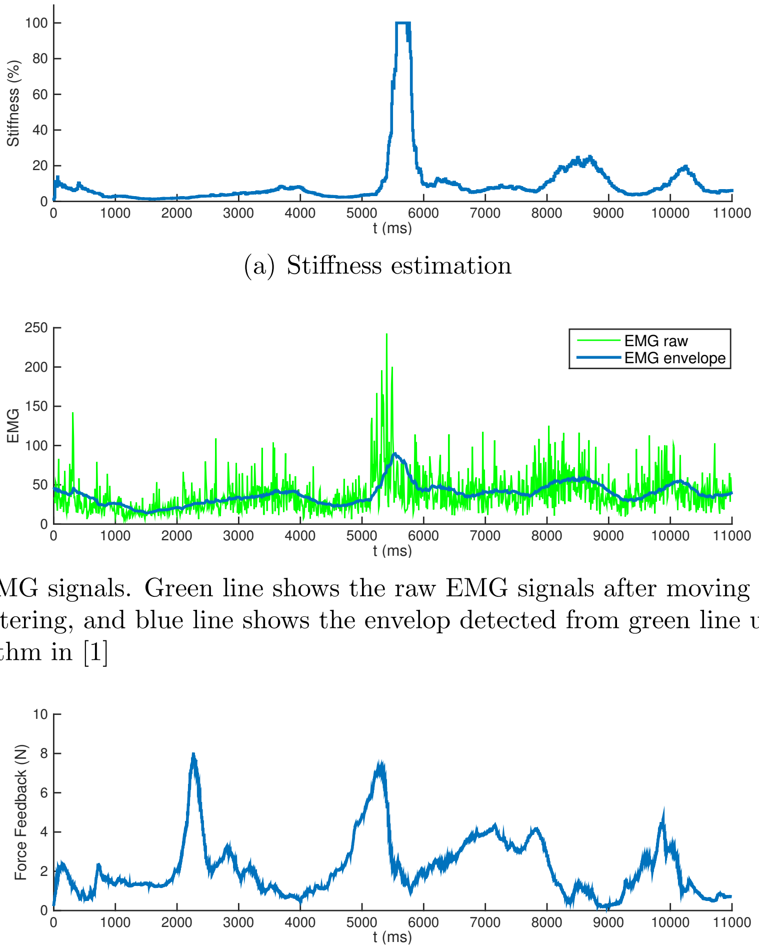
Figure 16: Tracking performance. Solid red lines: actual position trajectories of robot manipulator. Solid black lines: commanded position trajectories set by Omni joystick.

3.5.2 Test in torque control mode

In the stiffness control method, the manipulator's stiffness can be set in order to adjust the torque of the manipulator. A lift-up and move task is designed to verify whether the EMG based variable stiffness can improve the performance of the task regarding efficiency and accuracy. In this experiment, the operator is asked to lift up the target object from a start position labelled with the blue cross on the table to a target position labelled red cross on the table, as shown in Fig.11(b).

As shown in Fig. 11(b), a one kilogram toolbox is an object to be lift-up and moved. When high stiffness is applied, the toolbox can be lifted successfully. However, the stiffness of the manipulator is set according to the unloaded condition. When the heavy box has been lifted, the dynamic of the manipulator is changed, and the manipulator begins to shake, which can be seen from the trajctory in Fig. 16(a). When low stiffness is applied, as demonstrated in Fig. 16(b), the object could not be lifted at all and could be barely dragged to the target position. While when sEMG based variable stiffness is applied, in Fig. 16(c), the manipulator was able to lift up the object as well as to keep itself stable. Fig. 16(d) demonstrates the Root Mean Square Error (RMSE) of tracking performance in the three different modes. The sEMG based variable stiffness mode presents the lowest RMSE among these three tests. Furthermore, better user experience than the high stiffness and the low stiffness modes has been achieved, especially for the untrained, unskilful operators.

In Fig. 17(a), the blue curve represents the stiffness percentage of the manipulator. In Fig. 17(b), the green one represents the raw data of the EMG signal. The blue curve is the filtered EMG signal. Force feedback of the Omni is shown in Fig. 17(c). Fig. 17 demonstrates that when the sEMG based variable stiffness control mode is used, the stiffness percentage of the manipulator will increase when the operator is subject to a large feedback force forced on his/her forearm and vice versa.



(a) Stiffness estimation

(b) EMG signals. Green line shows the raw EMG signals after moving average filtering, and blue line shows the envelop detected from green line using algorithm in [1]

(c) Force feedback

Figure 17: Relationship between force feedback and robot tracking speed, under sEMG based variable stiffness mode

3.6 Conclusion

This chapter which has been published in [61], and awarded as Best Conference Paper Award on ICIA 2015. This chapter presents a novel motion control strategy for telerobots, using variable control gain which can be adjusted in real time according to muscles activation by a human operator, whereas the muscle activation is extracted from sEMG measured from a human operator. Both haptic feedback and visual feedback (from human's eye) are combined in order to fully exploit human motor skills in stiffness and force adaptation when interacting with external forces. A simple yet effective haptic rendering algorithm generating force feedback proportional to position tracking error is used. The sEMG based variable gain control strategy has been implemented in both position control mode and torque control mode, and a master-slave framework is employed in teleoperation. Workspace between the master device and the slave telerobot manipulator has been mapped in order to match the researchable spaces. Comparative experimental studies have demonstrated that the proposed method can improve teleoperation performance in both position control mode and torque control mode, in terms of manipulation performance as well as user experience.

4 Visual Servoing of Humanoid Dual-arm Robot with Neural Learning Enhanced Skill Transferring Control

4.1 Introduction

The issues pertaining to robot control has gained increasing research attention, recently. VS is a technique of control using computer vision information to control the motion of a robot. It mainly consists of techniques relating to computer vision, image processing and control theory [62, 63]. VS is one of the most important way in improving the flexibility of robot control systems [64] and has been widely applied. There are two central setups of the camera and the robot end-effector: Eye-in-hand, also known as end-point open-loop control, where the position of the object is captured by the camera appended to the robot hand; Eye-to-hand, or end-point closed-loop control, where the movement of the end-effector and the object are both acquired by a camera settled on the world frame [65]. Several approaches aims to provide better observation of target objects by increasing the number of cameras, such as a system using multiple cameras in [66]. A combination of eye-in-hand camera and eye-to-hand camera system in [67]. However, they have low adaptability to a changing environment. In this section, a control method of a Baxter robot arm end-effector using a stereo visual camera ZED as the eye-to-hand camera is developed. Eye-in-hand VS can only provides a narrow field of view, since the sensors are attached in the hand. ZED sensor consists of a pair of progressive scan CCD cameras with fixed alignment in between. It is used to detect the target objects position in workspace. A Least Squares based method is proposed to reduce stochastic errors during the camera calibration process.

To improve robot arm's control performance, an adaptive controller was developed for robot manipulators [68]. It employed a barrier Lyapunov function based synthesis to design a controller for the manipulator to operate in an ellipsoidal constrained region. An adaptive neural network (ANN) control for the robot system in the presence of full-state constraints is designed [69]. The NN enables the system to deal with uncertainties and disturbances effectively. Among these work above, we see that NN technique has been extensively used for robot control system due to its universal approximation ability and its capability to cope with unmodeled dynamics of the robot systems. The highly nonlinear nature of the robot dynamics makes it challenging to obtain an accurate model under practical operational conditions [70]. However, conventional NN control focuses on internal uncertainties. To overcome the uncertainties produced from the unknown payload, a novel NN based intelligent controller is designed in this section with an enhanced performance of VS control.

Furthermore, the learning ability of conventional NN controllers is limited, since even repeating the same task, the parameters of controller need recalculation every time. Therefore, a *deterministic learning* technique has been developed not only to obtain control dynamic knowledge from closed-loop control process, but also to reuse the obtained knowledge for other similar control task without readapting to the uncertainties of the environments [71]. Deterministic learning is proposed by using deterministic calculations that began from adaptive control, rather than utilising syntactical standards. The deterministic learning approach tackles the issue of learning in a dynamic situation and is valuable in numerous applications, for example, dynamic pattern recognition [72, 73], learning and control of robotics [74, 75], and oscillation faults diagnosis [76]. In addition to the designed NN controller, deterministic learning feature is added to reuse the learnt knowledge efficiently. After the initial learning of the environmental uncertainties, the proposed NN controller do not need to re-learn until dynamics changes. It can greatly reduce the computational load.

With the aim of improving the “intelligence” of the robot, a robot-to-robot skill transfer mechanism is proposed. By using surface electromyography signal, human arm stiffness was extracted to transfer human writing skills to the robot [1]. A communication language was developed of transferring grasping skills from a human user to a robot [77]. Unlike these conventional approaches of transferring human skills to a robot, the learned knowledge from NN controller is transferred from arm to arm with a dual-arm robot. With the guaranteed performance, NN controller only needs to learn once of system uncertainties on one side of dual-arm. The other arm can perform the same task without readapting the same uncertainties. It can help to increase the neural learning efficiency and reduce the computational load further.

In this context, this section presents an neural learning enhanced visual servoing control system with knowledge reuse and skill transfer features. This neural learning based compensation mechanism can be used to overcome the effect of the unknown payload as well as uncertainties associated with the telerobot model and environment. The system was successfully implemented on a Baxter humanoid robot, and test results are demonstrated, which show the potential of the novel learning controller.

4.2 Preliminaries

In order to ensure the desired tracking performance during teleoperation, we now consider shaping the transient response (e.g., overshoot, convergence rate, and convergence accuracy) of motion tracking error e_q. Let us define a set of smooth, bounded, strictly positive, and decreasing functions γ_i, $i = 1, 2, \ldots, n$ to impose constraints on the tracking error e_q, with $\lim_{t\to\infty} \gamma_i(t) = \gamma_{i\infty}$. Here, $\gamma_i(t)$ is chosen as [78]

$$\gamma_i(t) = (\gamma_{i0} - \gamma_{i\infty})e_q^{-\kappa_i t} + \gamma_{i\infty} \tag{18}$$

where γ_{i0}, $\gamma_{i\infty}$, and κ_i are positive constants. Then, the mathematical expression of predefined tracking performance can be described by using γ_i as below

$$\begin{aligned} -\delta\gamma_i(t) < e_q(t) < \gamma_i(t) \text{ if } e_q(0) > 0 \\ -\gamma_i(t) < e_q(t) < \delta\gamma_i(t) \text{ if } e_q(0) < 0 \end{aligned} \tag{19}$$

where the positive constant δ can be designed arbitrarily small. $\gamma_{i\infty}(t)$ and $-\delta\gamma_{i\infty}(t)$ represent the upper and lower bound of the tracking error e_q's maximum allowable magnitude at the steady state when $e_q(0) > 0$, respectively. $\delta\gamma_{i\infty}(t)$ and $-\gamma_{i\infty}(t)$ respectively represent the upper and lower bound of the tracking error e_q's maximum allowable magnitude at the steady state, when $e_q(0) < 0$. Therefore, the performance function $\gamma_i(t)$ can predefine the controller performance after determining the above parameters.

Lemma [79]: Consider a parameterised linear time-varying (LTV) multivariable systems in the following form:

$$\begin{bmatrix} \dot{e} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} A(e,\lambda) & B(e,\lambda)^{\mathrm{T}} \\ -C(t,\lambda) & 0 \end{bmatrix} \begin{bmatrix} e \\ \theta \end{bmatrix}, z := \begin{bmatrix} e \\ \theta \end{bmatrix} \tag{20}$$

where $e \in \mathbb{R}^n$, $\theta \in \mathbb{R}^m$, $A(e,\lambda) \in \mathbb{R}^{n\times n}$, $B(e,\lambda) \in \mathbb{R}^{m\times n}$, $C(e,\lambda) \in \mathbb{R}^{m\times n}$, $\lambda \in D \subset R^l$.

There exists a constant $\phi_M > 0$ such that for all $t \geqslant 0$ and for all $\lambda \in D$,

$$\max\left\{\|B(t,\lambda)\|, \left\|\frac{\partial B(t,\lambda)}{\partial t}\right\|\right\} \leqslant \phi_M. \tag{21}$$

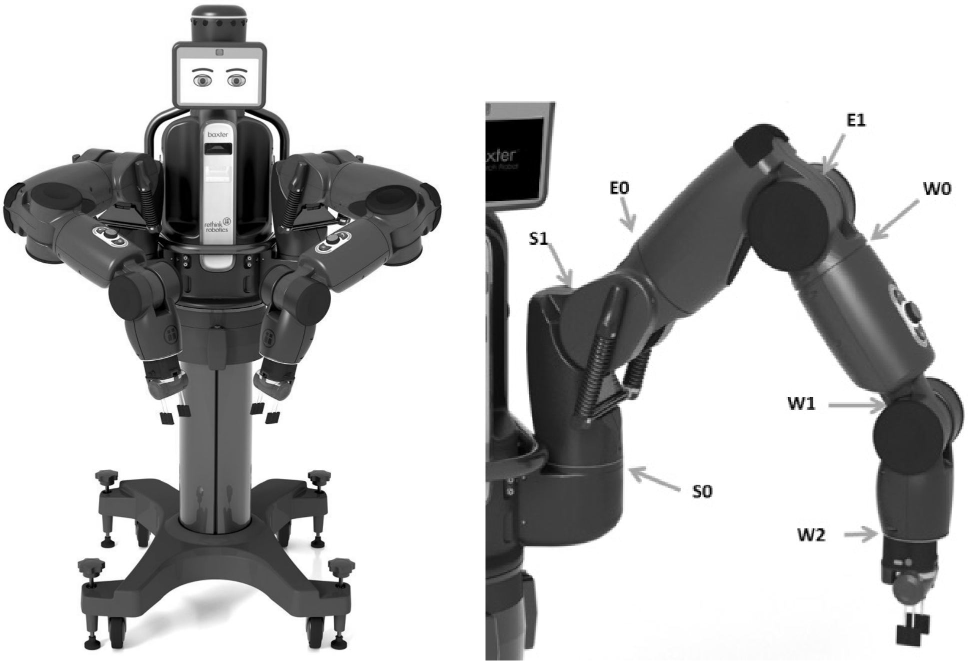
And there exist symmetric matrices $P(t,\lambda)$ and $Q(t,\lambda)$ such that $P(t,\lambda)B(t,\lambda)^{\mathrm{T}} = C(t,\lambda)^{\mathrm{T}}$ and $-Q(t,\lambda) := A(t,\lambda)^{\mathrm{T}}P(t,\lambda) + P(t,\lambda)A(t,\lambda) + \dot{P}(t,\lambda)$. Furthermore, $\exists p_m$, q_m, p_M and $q_M > 0$ such that, for all $(t,\lambda) \in \mathbb{R}_{\geqslant 0} \times D$, $p_m I \leqslant P(t,\lambda) \leqslant p_M I$ and $q_m I \leqslant Q(t,\lambda) \leqslant q_M I$.

Then, the system is λ-uniformly globally exponentially stable (λ-UGES) if and only if $B(\cdot,\cdot)$ is λ-uniformly persistency of excitation (λ-uPE), and the in-bound constants are independent of the initial conditions λ.

4.3 Kinematics Modelling of Humanoid Baxter® Robot Arms

4.3.1 Dual arms workspace identification for humanoid Baxter® robot

Baxter® robot is a humanoid robot with an identical pair of seven degrees of freedom (DOF) manipulators installed. Each manipulator has seven rotational joints and eight links as shown in Fig. 18(a). The joint naming of the arm is displayed in Fig. 18(b).

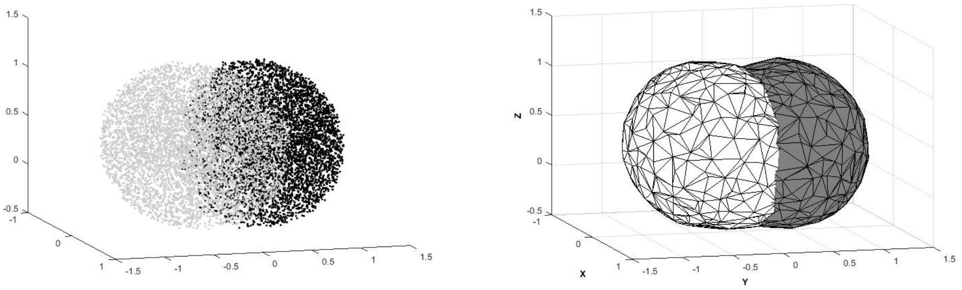


(a) Baxter robot arm

(b) Baxter robot arm joint naming

Figure 18: Baxter humanoid robot and its joint naming. S0 - Shoulder Roll. S1 - Shoulder Pitch. E0 - Elbow Roll. E1 - Elbow Pitch. W0 - Wrist Roll. W1 - Wrist Pitch. W2 - Wrist Roll

Baxter robot's kinematic model together with DH parameters and joint rotation limits were discussed from our previous work [80]. It is essential to estimate the robot manipulator workspace for optimised robotic design and algorithm. The previous method used on a single arm [80] is extended to both arms to calculated the reachable workspace. 6000 randomly chosen points in the joint space for each arm were generated. Then, point clouds of the reachable workspace for both manipulators were generated based on the end-effector positions calculated with forward kinematics, as illustrated in Fig. 19(a). Furthermore, Delaunay triangulation is applied to the point cloud to generate a convex hull of the joint space, as illustrated in Fig. 19(b). These convex hulls are used to identify the individual workspace for left and right arm independently in order to let them co-operate more efficiently during control.



(a) The point cloud of the reachable workspace of Baxter robot arms.

(b) The convex hull of the reachable workspace of Baxter robot arms.

Figure 19: The identification of Baxter's workspace.

4.4 Setup of Stereo Vision Sensor

4.4.1 System Structure Overview

The robot control communication network is shown in the Fig.20. The ZED stereo camera is a passive depth camera consists of two RGB-cameras with fixed alignment. It is used as visual sensors in the robotic control system. It captures videos in 30 fps under 1280×720 resolution to produce dense coloured depth maps for estimating the positions of objects. ZED keeps capturing videos of objects by its two cameras and sends them to a client computer via a USB 3.0 cable. Based on the difference between the two videos, client computer constructs disparity maps where the 3-D position information of objects can be read. Then, the target object's position information is sent to the Server Computer via UDP packets. The server computer will receive and decode them and then command Baxter to follow the target object along a reference trajectory.

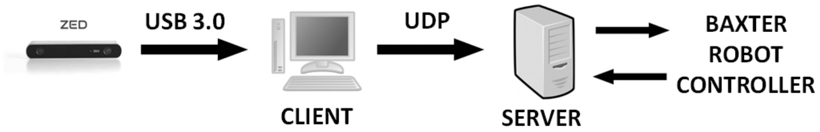


Figure 20: Communication Network

4.4.2 Stereo Camera Calibration

Raw pictures captured by ZED are distorted because lenses in ZED introduce nonlinear lens distortion deviating from the simple pin-hole model. To solve this problem, camera parameters calibration is necessary. To this end, we need to find out the camera parameters such as the intrinsic, extrinsic and distortion. Usually, researchers used a 2D checker-board pattern to evaluate them, instead of the complexity of 3D reference models, given the high cost of precise calibration objects. Among all the above parameters, intrinsic and distortion parameters are provided by the manufacturer, which can be employed directly.

After we complete the camera parameters calibration, undistorted pictures can be captured from ZED. Then, we can get an object's coordinates in ZED's coordinate system. However, in practice, the position of objects is presented in Baxter coordinate system rather than ZED. Therefore, we need to transform the ZED coordinates into the Baxter coordinates, i.e., the position calibration is necessary. The transform equation is shown as equation(22).

$$T\begin{bmatrix} X_1 & X_2 & \dots & X_i \\ Y_1 & Y_2 & \dots & Y_i \\ Z_1 & Z_2 & \dots & Z_i \\ 1 & 1 & \dots & 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \dots & x_i \\ y_1 & y_2 & \dots & y_i \\ z_1 & z_2 & \dots & z_i \\ 1 & 1 & \dots & 1 \end{bmatrix} \tag{22}$$

where T is the transform matrix. (X_i, Y_i, Z_i) denotes coordinates in ZED and (x_i, y_i, z_i) means coordinates in Baxter.

The aim of position calibration is to form the coordinate transform matrix T.

T can be achieved by the formula (23).

$$T = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} X_1 & X_2 & X_3 & X_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \\ Z_1 & Z_2 & Z_3 & Z_4 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \in R^{4\times 4} \tag{23}$$

where (x_i, y_i, z_i) and (X_i, Y_i, Z_i), $i = 1, 2, 3, 4$, are four non-coplanar point coordinates in the robot coordinate system and the ZED coordinate system, respectively.

To measure coordinates in Baxter coordinate system, the most straightforward way is to use direct measurement by a measuring tape. However, it is very coarse because the origin of the Baxter coordinate system is inside Baxter's body which is unavailable. Furthermore, it is also hard to ensure the horizontality and verticality of the ruler. Another way to measure coordinates is to use the kinematics of Baxter. At first, some established reference coordinates are given, and then we command Baxter's end-effector to move to these positions by using kinematics. In this way, we can get the end-effector's coordinates without direct measurement. Then, we use ZED to measure the end-effector's coordinates in ZED's coordinate system, which will be introduced in the next section. Subsequently, the points' coordinates in both Baxter coordinate system and ZED in formula (23) are easily achieved.

However, when using kinematics, stochastic errors always exist. In order to reduce these errors, Least Squares Method is employed. The aim of this algorithm is to calculate an overall solution which minimises the sum of the square errors in the given data. In order to employ this method in the calibration, we must transform equation (22) in the form of equation (25). The transformation can be done as below:

$$\begin{bmatrix} X_1I_4 & Y_1I_4 & Z_1I_4 & I_4 \\ X_2I_4 & Y_2I_4 & Z_2I_4 & I_4 \\ \vdots & \vdots & \vdots & \vdots \\ X_nI_4 & Y_nI_4 & Z_nI_4 & I_4 \end{bmatrix} \begin{bmatrix} T_{c1} \\ T_{c2} \\ T_{c3} \\ T_{c4} \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \\ \vdots \\ x_n \\ y_n \\ z_n \\ 1 \end{bmatrix} \tag{24}$$

where $I_4 \in R^{4\times4}$ means identity matrix. $T_{ci} \in R^{4\times1}$ means the column vector in the transform matrix T.

Let $A = \begin{bmatrix} X_1I_4 & Y_1I_4 & Z_1I_4 & I_4 \\ X_2I_4 & Y_2I_4 & Z_2I_4 & I_4 \\ \vdots & \vdots & \vdots & \vdots \\ X_nI_4 & Y_nI_4 & Z_nI_4 & I_4 \end{bmatrix}$, $X = \begin{bmatrix} T_{c1} \\ T_{c2} \\ T_{c3} \\ T_{c4} \end{bmatrix}$ and $B = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \\ \vdots \\ x_n \\ y_n \\ z_n \\ 1 \end{bmatrix}$, we can rewrite (24) into

$$AX = B \tag{25}$$

while A is a known matrix with a dimension of $4n \times 16$. X represents the transformation matrix T with a dimension of 16×1. B is a column vector with a dimension of $4n \times 1$. In most cases, this equation has no analytical solution. The system is overdetermined because to suppress the undesired effect from measuring noise, the number of sampled data that comprise Eq. (25) is greater than the dimensions of solution. Therefore, only numerical solution to Eq. (25) can be obtained by using the Least Square method is chosen. Initially, Eq. (25) is transformed as below:

$$A^T A X = A^T B \tag{26}$$

If $A^T A$ is nonsingular, the transformation matrix can be calculated as below:

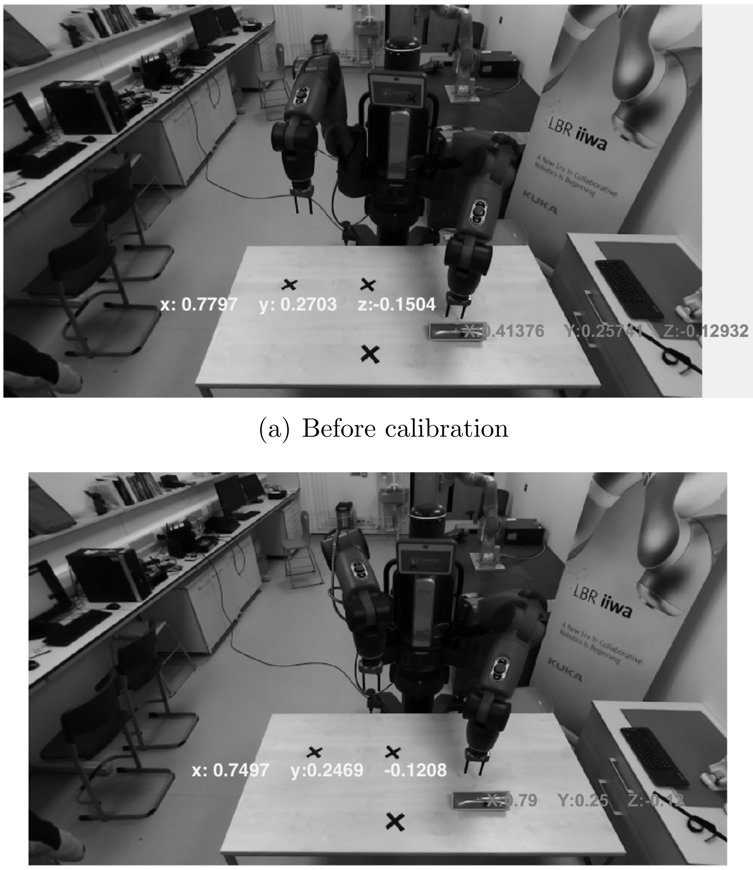
$$X = (A^T A)^{-1} A^T B \tag{27}$$

According to the equation(27), the solution of equation(24) can be achieved, i.e. the transform matrix T can be solved by the method of Least Squares. We can get a more precise solution by completing more coordinates measurement in ZED and Baxter.

Since the robot arms contain red colours and green colours are easily impacted by illumination, a blue object was used for detection. We firstly extracted (X_i, Y_i, Z_i), $i = 1, 2, 3, 4$ of the object's centroid from four different positions, out of ZED camera, as the black XYZ shown in Fig. 21(a). The end-effector's position (x_i, y_i, z_i), $i = 1, 2, 3, 4$ were recorded simultaneously. The end-effector were posed $10cm$ behind the object's centroid, in order to follow the object while not block the object from camera view, as the white xyz shown in Fig. 21(a). Then we substituted (x_i, y_i, z_i) and (X_i, Y_i, Z_i), $i = 1, 2, 3, 4$ into equation (24) to get the transformation matrix T. T was applied to the object's centroid position, and the data was sent to the robot as reference coordinates for following the object. The results are shown in Fig. 21(b) and Fig. 22, black XYZ stands for object's reference coordinates and white xyz stands for the coordinates that robot end-effector actually followed.

4.4.3 Theory of Depth Measurement in ZED

Both pictures captured under active ambient lighting by the ZED stereo camera, are aligned utilising the camera intrinsics and are amended for distortion.


x: 0.7797 y: 0.2703 z:-0.1504
X:0.41376 Y:0.25741 Z:-0.12932
x: 0.7497 y:0.2469 -0.1208
X:0.79 Y:0.25 Z:-0.12

(a) Before calibration

(b) After calibration

Figure 21: Positions of the object and the end-effector, left image used for displaying and monitoring. Black XYZ: object's coordinates under camera's frame of reference. White xyz: end-effector's coordinates under robot's frame of reference.

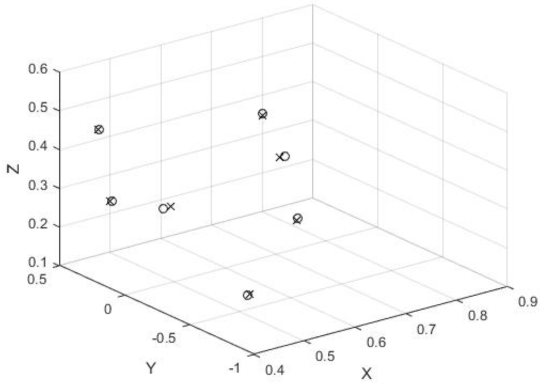


Figure 22: Precision of calibration. Cross mark: Object's position. Circle mark: End-effector's position.

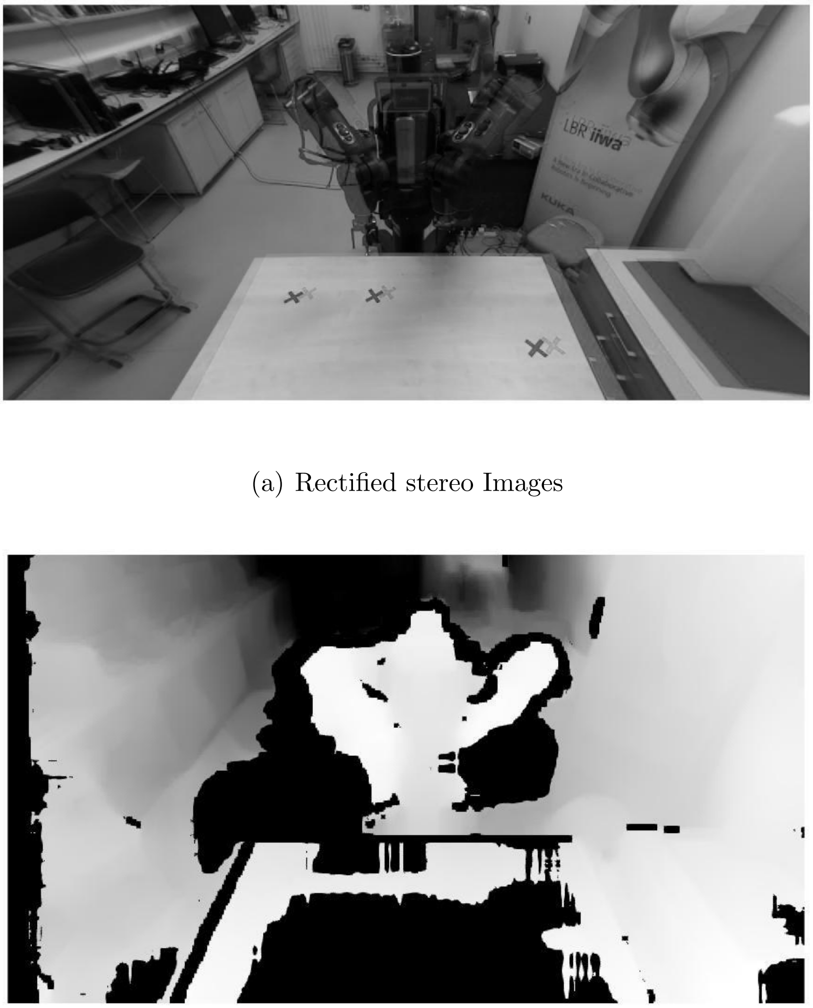
In this way, the undistorted images will be stereo rectified to adjust both the projection planes' epipolar lines and guarantee comparable pixels' presence in a predetermined row of the image. The images acquired are then frontal paralleled and are estimated correspondingly. The Fundamental and the Essential frameworks are figured by utilising Epipolar geometry. There are seven parameters in the Fundamental matrix representing two images' pixel relations, three for two image planes' homography and two for each epipole. The Essential matrix has five parameters in a 3×3 matrix, three of them are the rotation values between the camera projection planes and two for translation. Then the epipolar lines were adjusted, and the epipoles was moved to infinity. Fig. 23(a) delineates the results of stereo correction with row adjusted pixels.

Table 1: Definition of variables

1	x_l	column value of the left image pixel
2	x_r	column value of the right image pixel
3	D	Depth (mm)
4	B	Baseline (mm)
5	f	focal length (mm)
6	d	disparity
7	P	Projection matrix
8	X/ω, Y/ω, Z/ω	3D world coordinates

The definition of variables utilised underneath is given in Table 1. Stereo correspondence is a technique for coordinating pixels with comparative surface texture over two co-planar picture planes. The separation between the columns of these splendidly coordinated pixels is characterised as $d = x_l - x_r$.

Block matching is actualised for assessing the image correspondence. With the use of the sum of absolute differences (SAD), a 15-pixel window block is used to discover the matching results. Considering computational load, the disparity range is selected low as [0 40] to match the low texture difference of the experiment environment. In order to get a more complete outcome, Semi-Global method is used to drive the disparity values to the neighbouring pixels [81]. The output of the disparity map is illustrated in Fig. 23(b).



(a) Rectified stereo Images

(b) Disparity Map

Figure 23: Stereo images and 3D reconstruction.

The disparity can be calculated by the Triangulation equation $D = B\frac{f}{d}$, where D, B, f, d is defined in Table 1. It is inversely proportional to the depth of the pixel. Bouguet's algorithm is used to obtain the Cartesian coordinates from the reconstruction of the image, and the equation is shown below (28).

$$P\left[x, y, d, 1\right]^T = \left[X, Y, Z, \omega\right]^T \tag{28}$$

where $\omega \neq 1$ is the homogeneous component.

4.5 Detection and Localisation of Target Object

4.5.1 Colour object detection

Colour-based segmentation is utilised in order to isolate a single colour object from the captured image. One approach is to convert the entire RGB frame into corresponding Hue-Saturation-Value (HSV) plane and concentrate the pixel values of the colour you want to detect. To do so, one may be able to detect almost every single distinguishable colours in a single frame. However, it implementation in a live video is challenging because of ambient light. An alternative approach was proposed in our team's previous work [82], which converted the captured image into L*a*b* colour space where the value of 'a' and 'b' is related to the colour information of a point.

During the experiments, all images are converted into L*a*b* colour space and the variance between every point's colour and the standard colour marks will be calculated. The estimations are selected based on the minimum variance value of each image. Furthermore, the intersection of the diagonals was used to calculate the centroid and Harris corner detector was used to calculate the corners of the object. According to the centroid point in the image, the object's coordinates in ZED is then extracted from the images. By applying the transformation matrix in section 4.4.2, the object's coordinates in Baxter's coordinate system can be calculated. Fig. 21(b) demonstrates the calculated centroid of the object after coordinate transformation in robot coordinates.

4.5.2 Object Detection Regulation

After several tests, we find that because of the nonuniform distribution of light in space, the object's colour in images keeps changing as the object moves. Sometimes the colour values of 'a' and 'b' change a lot that it affects the stability of object detection. To solve this problem, we employed a regulation algorithm in object detection. The algorithm is described below. (i) Calculate the variance between the image points' colour and the colour marks. (ii) If the value of the variance of the object is not so large, go back to (i) and continue next detection. Conversely, go to (iii). (iii) Calculate the average value of ‘a’ and ‘b’ around the centroid points, and update the older colour marks with new values. Then start next detection based on these new colour marks.

By employing the algorithm above, object detection becomes more stable and more adapted to the environment.

4.6 Switching control between two arms

The depth of the object to be manipulated is mapped to the convex hull to expedite the process of decision making, because a point cloud matching would limit the processing speed. This is done by checking the presence of the point’s respective co-ordinates in the convex-hull projection on the three Cartesian planes. For example, if (X_1, Y_1, Z_1) is the point representing the object, its presence in the 3D hull is detected by following five steps:

Algorithm 1 Object detection in the 3D workspace hull

```
1: Check if the $XY$ plane projection of the hull contains the point $(X_1, Y_1)$,
   $YZ$ plane contains point $(Y_1, Z_1)$ and $XZ$ plane contains point $(X_1, Z_1)$
   using Ray Casting Algorithm;
2: Obtain the presence decision in the workspace of both arms;
3: If the point locates in both of the manipulator workspace, give the control
   priority to the arm with the smallest Euclidean distance from manipulator;
4: Divide the control based on the detection of the point in left or right
   workspaces;
5: If the point lies outside both the workspaces, stop arm movement to avoid
   singularity.
```

The above procedure ensures an efficient motion to reach the object in the robot workspace. It mimics the human intuition of using the nearest possible arm for grabbing in order to avoid the use of excessive body movements and minimize the use of energy.

Once the target object’s co-ordinates has been detected, end effector desired trajectory for the visual tracking task can be defined as the moving trajectory of the target. As for visual guided manipulation task, the end effector desired trajectory can be generated according to the location of the target object. Denote the joint velocities as $\dot{\boldsymbol{\theta}}$ and end-effector velocity as $\dot{\boldsymbol{x}}$, then they must satisfy $\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{\theta}}$ where $\boldsymbol{J}$ is the Jacobian matrix of each arm.

4.7 Neural Network Controller Design

4.7.1 Adaptive Neural Controller

An adaptive NN based controller is designed to achieve the following control of the joint space trajectory. The dynamic equation of the manipulator is shown in (69).

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{G}(\boldsymbol{\theta}) + \boldsymbol{\tau_{ext}} = \boldsymbol{\tau}, \tag{29}$$

where $\boldsymbol{M}(\boldsymbol{\theta})$ is the manipulator inertia matrix, $\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is the Coriolis matrix for the manipulator, $\boldsymbol{G}(\boldsymbol{\theta})$ is the gravity terms and $\boldsymbol{\tau_{ext}}$ denotes the external torque including the payload gravity applied at the end-effector.
Define $\boldsymbol{s} = \dot{\boldsymbol{e}}_{\boldsymbol{\theta}} + \boldsymbol{\Lambda}\boldsymbol{e}_{\boldsymbol{\theta}}$, $\boldsymbol{v} = \dot{\boldsymbol{\theta}}_{\boldsymbol{d}} - \boldsymbol{\Lambda}\boldsymbol{e}_{\boldsymbol{\theta}}$, where $\boldsymbol{e}_{\boldsymbol{\theta}} = \boldsymbol{\theta} - \boldsymbol{\theta_d}$, $\boldsymbol{\Lambda} = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)$. Then, the dynamic equation (69) can be rewritten as (70).

$$\boldsymbol{M}(\boldsymbol{\theta})\dot{\boldsymbol{s}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\boldsymbol{s} + \boldsymbol{F} = \boldsymbol{\tau} \tag{30}$$

where $F \in R^n$, is defined as

$$\boldsymbol{F} = \boldsymbol{M}(\boldsymbol{\theta})\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\boldsymbol{v} + \boldsymbol{G}(\boldsymbol{\theta}) + \boldsymbol{\tau_{ext}} \tag{31}$$

Design the adaptive controller as (71).

$$\boldsymbol{\tau} = \hat{\boldsymbol{F}} - \boldsymbol{K}\boldsymbol{s} \tag{32}$$

where $\hat{\boldsymbol{F}}$ is the estimate of $\boldsymbol{F}$, and $\boldsymbol{K} = \text{diag}\left\{k_i\right\}, i = 1, 2, \cdots, n$ is a diagonal matrix and $\min\left\{k_i\right\} > 0.5$.

Then, by substituting (71) into (70), the closed-loop dynamics of the robot system can be written as (72).

$$\boldsymbol{M}(\boldsymbol{\theta})\dot{\boldsymbol{s}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\boldsymbol{s} = \tilde{\boldsymbol{W}}^{\mathbf{T}}\boldsymbol{S}(\boldsymbol{z}) - \boldsymbol{\epsilon}(\boldsymbol{z}) - \boldsymbol{K}\boldsymbol{s} \tag{33}$$

The following function approximation method is used.

$$\begin{aligned} \boldsymbol{F} &= \boldsymbol{W}^{*\mathbf{T}}\boldsymbol{S}(\boldsymbol{z}) + \boldsymbol{\epsilon}(\boldsymbol{z}) \\ \hat{\boldsymbol{F}} &= \hat{\boldsymbol{W}}^{\mathbf{T}}\boldsymbol{S}(\boldsymbol{z}) \\ \tilde{\boldsymbol{F}} &= \hat{\boldsymbol{F}} - \boldsymbol{F} = \tilde{\boldsymbol{W}}^{\mathbf{T}}\boldsymbol{S}(\boldsymbol{z}) - \boldsymbol{\epsilon}(\boldsymbol{z}) \\ \tilde{\boldsymbol{W}} &= \hat{\boldsymbol{W}} - \boldsymbol{W}^{*} \end{aligned} \tag{34}$$

where $\boldsymbol{W}^{*} = [W_1^*, W_2^*, \cdots, W_n^*] \in R^{N \times n}$ is the weight matrix, $\boldsymbol{S}(\boldsymbol{z})$ is the basis function vector, $\boldsymbol{z} \in \Omega_z \subset R^q$ is the input vector with $\Omega_z \subset R^q$ being a compact set, N is the number of NN node, and $\boldsymbol{\epsilon}(\boldsymbol{z})$ is the approximation error. $\boldsymbol{s}(\boldsymbol{z}) = [s_1(\|z - \mu_1\|), \cdots, s_N(\|z - \mu_N\|)]^T$, is the regressor vector, with $s_i(\cdot)$ being a radial basis function, and μ_i $(i = 1, \cdots, N)$ being the centre. The Gaussian functions choose as

$$s_i(\|z - \mu_i\|) = \exp\left[\frac{-(z - \mu_i)^T(z - \mu_i)}{\varsigma^2}\right] \tag{35}$$

where $\mu_i = [\mu_{i1}, \mu_{i2}, \cdots, \mu_{iq}]^T \in R^q$ represents the centre of each receptive field and ς is the variance.

Choose the following Lyapunov function.

$$\boldsymbol{V} = \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{s}^{\mathbf{T}}\boldsymbol{M}(\boldsymbol{\theta})\boldsymbol{s} + \frac{\mathbf{1}}{\mathbf{2}}\mathbf{tr}(\tilde{\boldsymbol{W}}^{\mathbf{T}}\boldsymbol{Q}\tilde{\boldsymbol{W}}) \tag{36}$$

where $\boldsymbol{Q}$ is a positive definite weight matrix. And using the skew-symmetry [83] of the matrix $\dot{M} - 2C$, the first derivative of V can be calculated as

$$\dot{V} = -\boldsymbol{s}^{\mathbf{T}}\boldsymbol{K}\boldsymbol{s} - \boldsymbol{s}^{\mathbf{T}}\boldsymbol{\epsilon}(\boldsymbol{z}) + \boldsymbol{tr}\left[\tilde{\boldsymbol{W}}^{\mathbf{T}}\left(\boldsymbol{S}(\boldsymbol{z})\boldsymbol{s}^{\mathbf{T}} + \boldsymbol{Q}\dot{\tilde{\boldsymbol{W}}}\right)\right] \tag{37}$$

The update law is designed as follow.

$$\dot{\hat{\boldsymbol{W}}} = -\boldsymbol{Q}^{-1}\big(\boldsymbol{S}(\boldsymbol{z})\boldsymbol{s}^{\mathrm{T}} + \boldsymbol{\sigma}\hat{\boldsymbol{W}}\big) \tag{38}$$

where $\boldsymbol{\sigma}$ is a pre-designed positive constant.

Substituting (80) into (79), we have

$$\dot{\boldsymbol{V}} = -\boldsymbol{s}^{\mathbf{T}}\boldsymbol{K}\boldsymbol{s} - \boldsymbol{s}^{\mathbf{T}}\boldsymbol{\epsilon}(\boldsymbol{z}) - \boldsymbol{\sigma}\boldsymbol{tr}(\tilde{\boldsymbol{W}}^{\mathbf{T}}\hat{\boldsymbol{W}}) \tag{39}$$

Based on Young's inequality, from (81) we can have

$$\dot{\boldsymbol{V}} \leq -(\boldsymbol{\lambda_{min}}(\boldsymbol{K}) - \frac{\mathbf{1}}{\mathbf{2}})\|\boldsymbol{s}\|^{\mathbf{2}} - \frac{\boldsymbol{\sigma}}{\mathbf{2}}\|\tilde{\boldsymbol{W}}\|^{\mathbf{2}} + \boldsymbol{\rho} \tag{40}$$

where $\boldsymbol{\rho} = \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{\varepsilon}^{\mathbf{2}} + \frac{\boldsymbol{\sigma}}{\mathbf{2}}\|\boldsymbol{W}^{*}\|^{\mathbf{2}}$, with $\boldsymbol{\varepsilon}$ is the upper limit of $\|\boldsymbol{\epsilon}\|$ over Ω. If $\tilde{\boldsymbol{W}}$ and $\boldsymbol{s}$ satisfy the following inequality

$$(\boldsymbol{\lambda_{min}}(\boldsymbol{K}) - \frac{\mathbf{1}}{\mathbf{2}})\|\boldsymbol{s}\|^{\mathbf{2}} + \frac{\boldsymbol{\sigma}}{\mathbf{2}}\|\tilde{\boldsymbol{W}}\|^{\mathbf{2}} \geq \boldsymbol{\rho} \tag{41}$$

where I is the unit matrix, then we can have $\dot{\boldsymbol{V}} \leq \mathbf{0}$.

By using LaSalle's theorem, we see that $\|\tilde{\boldsymbol{W}}\|$ and $\|\boldsymbol{s}\|$ will converge to an invariant set $\Omega_s \subseteq \Omega$, on which $\dot{V}(t) = 0$, where Ω is the bounding set that is defined as

$$\Omega = \left\{ \Big(\|\tilde{\boldsymbol{W}}\|, \|\boldsymbol{s}\|\Big) \,\Big|\, \frac{\sigma}{2\rho}\|\tilde{\boldsymbol{W}}\|^2 + \frac{(2K - I)}{2\rho}\|\boldsymbol{s}\|^2 \leq 1 \right\}. \tag{42}$$

4.7.2 Analysis of NN Learning Convergence

By denoting a new subscript ζ, it represents the region which is close to the tracking trajectory, and $\bar{\zeta}$ represents the region which is far away from the tracking trajectory. Let $S_\zeta(z)$ be the element that the neurons located in the region of ζ, and $\hat{W}_\zeta$ is the associated weight matrix of NN. From (80) we can have

$$\dot{\tilde{W}}_\zeta = -Q_\zeta^{-1}\big(S_\zeta(z)s^{\mathrm{T}} + \sigma_\zeta \hat{W}_\zeta\big) \tag{43}$$

and from (73) we have that the NN approximation error $\epsilon_{i\zeta}(z)$ is close to $\epsilon(z)$. $\bar{S}_\zeta$ and $\bar{\tilde{W}}_\zeta$ are defined as below:

$$\bar{S}_\zeta = \begin{bmatrix} S_\zeta & 0_{[N_\zeta \times 1]} & \cdots & 0_{[N_\zeta \times 1]} \\ 0_{[N_\zeta \times 1]} & S_\zeta & \cdots & 0_{[N_\zeta \times 1]} \\ \vdots & \vdots & \vdots & \vdots \\ 0_{[N_\zeta \times 1]} & \cdots & 0_{[N_\zeta \times 1]} & S_\zeta \end{bmatrix} \in R^{nN_\zeta \times n} \tag{44}$$

and

$$\bar{W}_\zeta = [W_{1\zeta}^T, W_{2\zeta}^T, \ldots, W_{n\zeta}^T,]^T \in R^{nN_\zeta} \tag{45}$$

Subsequently, we define an augmented matrix of the diagonal matrix σ_ζ as $\bar{\sigma}_\zeta = [\sigma_\zeta, \sigma_\zeta, \cdots, \sigma_\zeta] \in R^{N_\zeta \times N_\zeta}$. From this, we could rewrite (43) into:

$$\dot{\bar{\tilde{W}}}_\zeta = -\bar{S}_\zeta(z)Q_\zeta^{-1}s^{\mathrm{T}} - Q_\zeta^{-1}\bar{\sigma}_\zeta \hat{W}_\zeta \tag{46}$$

Using the spatially localised approximation ability of RBF NN, the closed-loop system from (72) can be expressed as below:

$$\dot{s} = M^{-1}(\theta)[-Ks + \bar{S}_\zeta(z)\bar{\tilde{W}}_\zeta^{\mathrm{T}} - \epsilon_\zeta(z) - C(\theta, \dot{\theta})s] \tag{47}$$

Then, an LTV system can be created from the system of (47) and (46) as follow

$$\begin{bmatrix} \dot{s}_i \\ \dot{\tilde{W}}_{\zeta i} \end{bmatrix} = \begin{bmatrix} -M^{-1}(\theta)N(t) & M^{-1}(\theta)\bar{S}_{\zeta i}^T(z) \\ -Q_i^{-1}\bar{S}_{\zeta i}(z) & 0_{[N_\zeta \times N_\zeta]} \end{bmatrix} \begin{bmatrix} s_i \\ \bar{\tilde{W}}_i \end{bmatrix} + \begin{bmatrix} -M^{-1}(\theta)\epsilon_i(z) \\ -Q_i^{-1}\sigma_i \hat{W}_i \end{bmatrix} \tag{48}$$

where $N(t) = k_i + C(\theta, \dot{\theta})$, $i = 1, 2, \cdots, n$. Let $P = Q_i^{-1} M(\theta)$, which is symmetric, and let $\mathcal{A} = -M^{-1}(\theta) N(t)$, $\mathcal{B} = M^{-1}(\theta) \bar{S}_{\zeta i}^T(z)$, and $\mathcal{C} = Q_i^{-1} \bar{S}_{\zeta i}(z)$, then we have

$$\mathcal{A}^T P + P\mathcal{A} + \dot{P} = Q_i^{-1} \left(\dot{M}(\theta) - 2C(\theta, \dot{\theta}) - 2K \right) := U \tag{49}$$

Since $min\{k_i\} > 0.5$, Q_i is positive, and using the skew-symmetry [83] of the matrix $\dot{M} - 2C$, such that we can have $U < 0$. This guarantees the exponential stability of the nominal part of the system in Eq. (48). Then on the premise of small enough σ, the parameter error $\tilde{W}_\zeta$ will converge exponentially to a small neighbourhood (determined by $|\epsilon_\zeta(z)|$ and $\| - \sigma_\zeta \hat{W}_\zeta \|$) of zero for all $t > T_1$. Thus, $\hat{W}_\zeta$ can converge exponentially to a small neighbourhood of the desired weight value W_ζ^* for all $t > T_1$.

4.7.3 Knowledge Reusing and Skill Transfer

Now, we can accurately approximate the dynamical system $F(z)$ by using the localisation feature of RBFNN, with the convergence of $\hat{W}_\zeta$ such as

$$F(z) = \bar{W}_\zeta^T S_\zeta(z) + \bar{\epsilon}_\zeta(z) \tag{50}$$

where $\bar{\epsilon}(z)$ is close to $\epsilon(z)$ in the steady-state process, and

$$\bar{W}_\zeta = \text{mean}_{t \in [t_{ai}, t_{bi}]} \hat{W}_\zeta\,(t) = \frac{1}{t_{bi} - t_{ai}} \int_{t_{ai}}^{t_{bi}} \hat{W}_\zeta(s) ds \tag{51}$$

with $[t_{ai}, t_{bi}]$, $t_{bi} > t_{ai} > T_1$ representing a time segment after the transient process.

Let us define

$$\bar{W} = \text{mean}_{t \in [t_{ai}, t_{bi}]} \hat{W}\,(t) = \frac{1}{t_{bi} - t_{ai}} \int_{t_{ai}}^{t_{bi}} \hat{W}(s) ds \tag{52}$$

we will have

$$\hat{W}^T S_{\bar{\zeta}}(z) \approx \bar{W}_{\zeta}^T S_{\zeta}(z) \tag{53}$$

Therefore, we could use $\bar{W}_{\zeta}^T S_{\zeta}(z)$ to replace $\bar{W}_i^T S_i(z)$ for approximating the uncertainties of system dynamics $F(z)$.
Since the learnt knowledge will not keep in the memory, the control parameters have to be recalculated even when reproducing the similar control tasks. However, since the estimate $\hat{W}$ can converge into a small neighbourhood of the optimal W^*, $F(z)$ which is the accurate approximation of the system dynamics can be still achieved. The above learning method can be considered as the system dynamics approximation using constant NN weights.
Based on our previous work [78], the following control law is proposed to reuse the learnt knowledge instead of using the original NN based controller (71) and the updated law of RBFNN's weight (80)

$$\tau = -Ks + \bar{F}(z) \tag{54}$$

where $K = \operatorname{diag}\{k_i\}, i = 1, 2, \cdots, n, \min\{k_i\} > 0.5$ and $\bar{F}(z) = \bar{W}^T S(z)$.
With the property of dual-arm for Baxter robot, once one side of arm learnt the uncertainties of the environment, i.e. payload, the learnt knowledge can also be transferred and reused on the other arm without readapting the uncertainties. This feature can also be extended to a robot-to-robot skill transfer. While performing the same tasks, this mechanism can significantly help to reduce computational load with guaranteed performance.

4.8 Experiment Studies

A visual tracking task was performed to test the proposed VS method, with neural learning and without neural learning for comparison. The experiment setup is shown in Fig. 24. In each set of tests, the blue object was moved by the operator from the starting point ($\boldsymbol{P_1}$: $[0.7, -0.2, -0.2]$) to the end point ($\boldsymbol{P_2}$: $[0.7, 0.2, -0.2]$) in a rectangle trajectory. The object was lifted up after leaving the starting point and generally put down on the operating table level at the end.

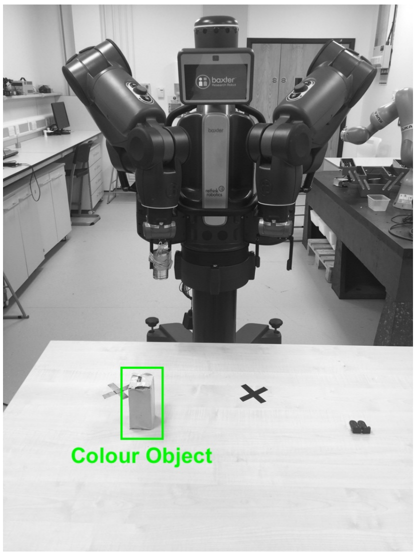


Figure 24: The experiment setup. Left cross: the start point. Right cross: the end point. Two different payloads were held in both grippers on the manipulator. The right and left one each weigh 1.3 kg and 0.7kg respectively.

Due to the 7-DOF robot dynamics, $N = 3^7 \times 7$ nodes are employed for the NN to complete a high precision of approximation. While the NN's weight matrix is initialised as $\hat{W}(0) = 0 \in R^{15309\times7}$. The design parameters K of the controller are specified as $K = \text{diag}\{9, 9, 8, 4.5, 1.8, 1.2, 0.8\}$.

The object reference trajectories which has been recorded using MATLAB and the end-effector trajectories of this set of comparative experiments are demonstrated in Fig. 25. The NN learning weights of individual joints are demonstrated in Fig. 26. The compensation torques obtained by NN of each joint are shown in Fig. 27.

4.8.1 Control without NN Learning

During this initial set of experiments, the performance of the control method without NN learning is tested to establish baseline performance. The colour object was held by the operator and was moved along a predefined trajectory as introduced earlier. From 25(a), we can see the actual position trajectory is below the reference trajectory because of the heavy payload.

4.8.2 Control with NN Learning

During this set of experiments, the same task as the first experiment was performed. In this set, the NN learning was added to the controller, and the performance of the telerobot manipulator was recorded. Compared with the first test, NN learns the payload's weight during teleoperation and subsequently affects the control inputs. As can be seen from 25(b), the robot was able to restore to normal tracking position. The control torque inputs of right and left arm are shown in Fig. 27(a) and 27(b).

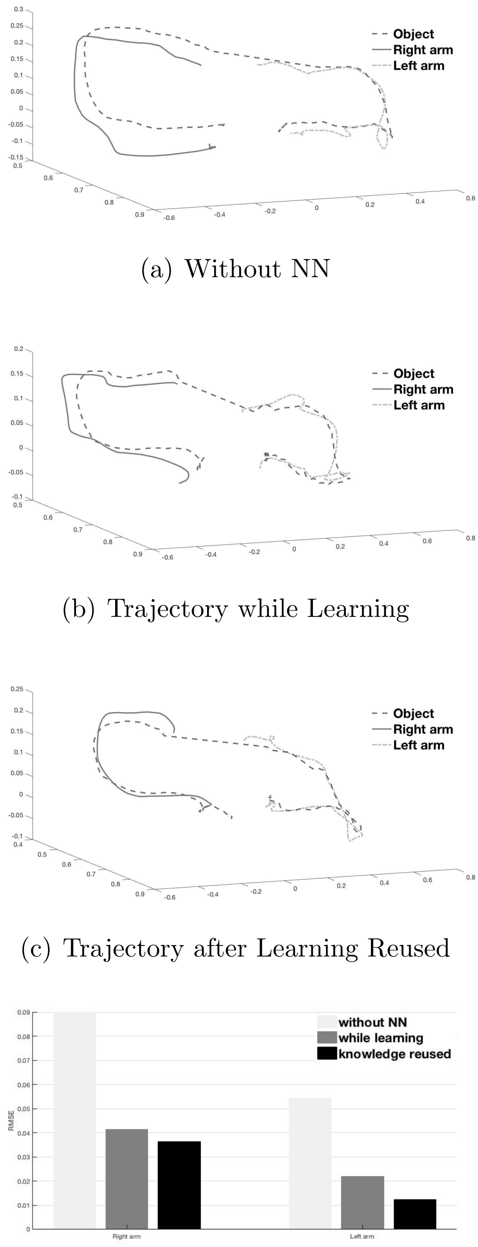
4.8.3 Control after NN Learning

During the last set of experiments, the NN will first learn the dynamics while both manipulators were tracking the object along a repeated trajectory, same as previous two. After four cycles, the NN adapted to the external dynamics (attached payload). So that the trained NN will be reused for the further teleoperation. The control torque inputs of right and left arm are shown in Fig. 27(c) and 27(d). The performance of tracking is illustrated in 25(c).

From Fig. 25(d), it can be seen that the designed adaptive controller can help system compensate tracking error from both internal and external dynamics. The trained NN has a steady performance with reusing the trained knowledge to increase tracking performance.

4.9 Conclusion

This Chapter has been published in [84]. An NN learning enhanced VS control method was developed in this chapter and implemented on a humanoid dual-arm Baxter robot. The colour object was detected by a stereo camera and a regulation algorithm was applied to ensure the effectiveness of detection. The calibration between the camera and robot's coordinates was done with the proposed least squared based method to reduce stochastic errors. The dynamic parameters of the manipulator are estimated by the radial basis function NN and an improved adaptive control method is designed for compensating the effect of uncertain payload and other uncertainties during the dynamic control of the robot. Specifically, a knowledge reuse method with skill transfer feature has been created to increase the neural learning efficiency. So that the learnt NN knowledge can be easily reused for finishing repetitive tasks and also can be transferred to another arm for performing the same task. The proposed NN controller was validated with tests on a Baxter humanoid robot. Experimental results demonstrate that it can realise the optimal performance under the designed VS control.



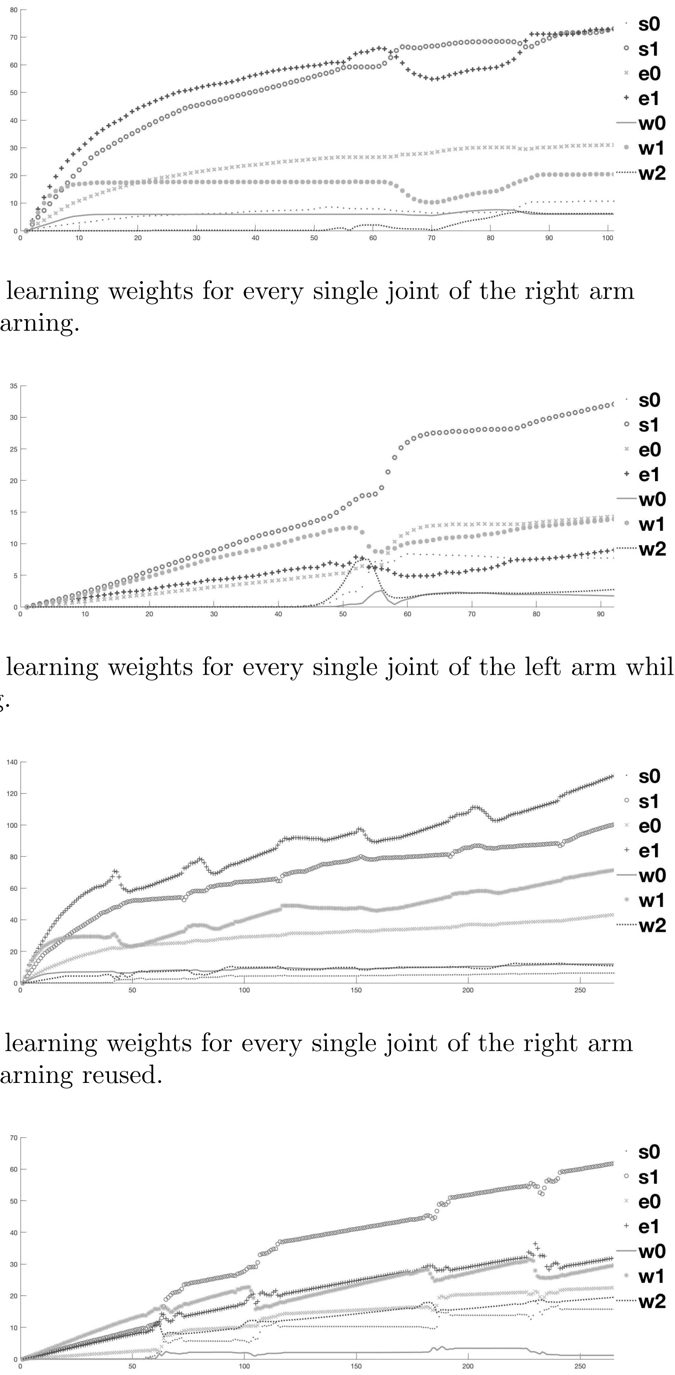
(a) Without NN

(b) Trajectory while Learning

(c) Trajectory after Learning Reused

(d) RMSE

Figure 25: Tracking trajectory and root-mean-square error (RMSE). (a-c) Dashed line: reference trajectories generated by object tracking. Solid and Dash-dot lines: actual position trajectories of both robot right and left manipulators respectively. (d) Left: RMSE of the right arm under three different conditions. Right: RMSE of the left arm under three different conditions.



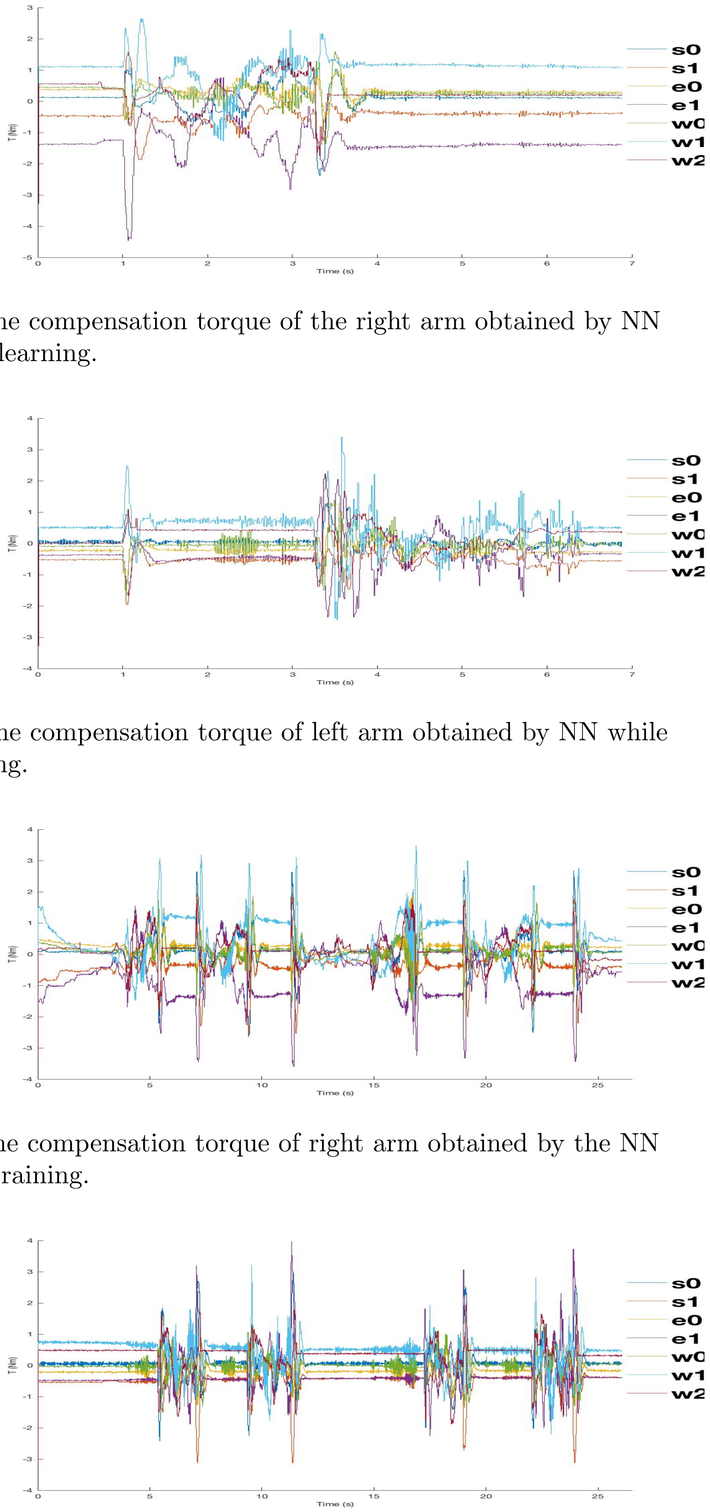
(a) NN learning weights for every single joint of the right arm while learning.

(b) NN learning weights for every single joint of the left arm while learning.

(c) NN learning weights for every single joint of the right arm while learning reused.

(d) NN learning weights for every single joint of the left arm while learning reused.

Figure 26: NN learning weights for every single joint.



(a) The compensation torque of the right arm obtained by NN while learning.

(b) The compensation torque of left arm obtained by NN while learning.

(c) The compensation torque of right arm obtained by the NN after training.

(d) The compensation torque of left arm obtained by the NN after training.

Figure 27: NN control torques for every single joint of both arms. (a) and (b) show the joint torques while NN was learning the uncertainties. (c) and (d) show the joint torques after NN were trained. s0, s1, e0, e1, w0, w1 and w2 represent the 7 joints of the robot arm.

5 Neural Learning Enhanced Teleoperation Control Using IMU Based Motion Capture

This Chapter presents a neural network (NN) control enhanced teleoperation strategy which has been implemented on the Baxter robot. The upper limb motion of the human operator is captured by the inertial measurement unit (IMU) embedded in a pair of MYO armbands worn on the operator's forearm and upper arm, respectively. They are used to detect and to reconstruct the physical motion of shoulder and elbow joints of the operator. Let the human operator's motion be reference trajectories. Subsequently, the robot is controlled using NN technique to compensate for its unknown dynamics. Adaptive law has been synthesised based on Lyapunov theory to enable effective NN learning. Previous experiments have been carried out to test the proposed method, which have shown satisfactory performance on the Baxter robot teleoperation.

5.1 Motion Capture

5.1.1 Human Arm Motion Capture by the Method of X-Y-Z Fixed Angles

As shown in Fig. 28, the global coordinate frame (x_s, y_G, z_G) is defined as: x-axis points to the side in lateral direction; y-axis points forwards, z-axis pointing upwards. For the easy calculation of the shoulder joint angles, the local frame of the humerus (x_H, y_H, z_H) and forearm frame (x_F, y_F, z_F) coincides with the global frame. The local frame of humerus (x_H, y_H, z_H) is defined as: x-axis pointing to the left, y-axis pointing to the forward, $z-axis$ pointing to the upward along the upper arm.

The local frame of the forearm (x_F, y_F, z_F) is defined as: x-axis pointing the left, y-axis pointing the forward along the forearm, z-axis pointing the upward. The rotation matrices of the orientations of the humerus frame and forearm

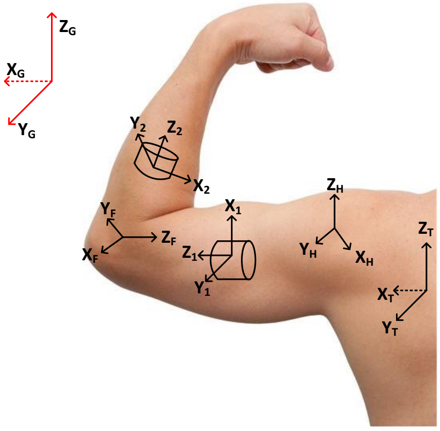


Figure 28: The global frame, local frame and MYO armbands frames (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2).

frame under the global frame can be obtained as below:

$$\mathcal{R}^i_{GH} = \begin{bmatrix} X^i_{GH} & Y^i_{GH} & Z^i_{GH} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{55}$$

$$\mathcal{R}^i_{GF} = \begin{bmatrix} X^i_{GF} & Y^i_{GF} & Z^i_{GF} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{56}$$

where $\mathcal{R} \in R^{3\times3}$ is the rotation matrix, i is the initial position, "G", "H" and "F" denote the global, humerus and forearm frame respectively. The rotation matrix $\mathcal{R}_{YX}$ represent the orientation of the frame "X" with respect to frame "Y". The column vector X_{MN}, Y_{MN}, and Z_{MN} denote unit vectors describing the principal directions of the frame "N" in terms of frame "M", and X_{MN}, Y_{MN}, $Z_{MN} \in R^{3\times1}$.

The local frame of humerus and forearm are not stationary and the orientation of the humerus frame with respect to the first MYO armband frame and the forearm frame with respect to the second MYO armband frame are constant matrixs, given as

$$\mathcal{R}^i_{UH} = (\mathcal{R}^i_{GU})^{\mathrm{T}}\mathcal{R}^i_{GH} \tag{57}$$

$$\mathcal{R}^i_{LF} = (\mathcal{R}^i_{GL})^{\mathrm{T}}\mathcal{R}^i_{GF} \tag{58}$$

where "U" represents the frame of the first MYO armband worn on the upper arm, and "L" represents the frame of the second MYO armband worn on the lower arm.

The orientations of the humerus and forearm under the global frame while the operator moving his/her arm can be described as:

$$\mathcal{R}^f_{GH} = \mathcal{R}^f_{GU}\mathcal{R}^i_{UH} \tag{59}$$

$$\mathcal{R}^f_{GF} = \mathcal{R}^f_{GF}\mathcal{R}^i_{LF} \tag{60}$$

where f denotes the current arm position.

A quaternion $q_1 = [x, y, z, w]^T$ can be obtained from the first MYO's gyroscope, where (x, y, z) is a vector and w is a scalar quantity.

$$q_1 = xi + yj + zk + w$$

According to [85], we can obtain the orientations of the humerus under the global frame from the quaternion, as follows:

$$\mathcal{R}^f_{GH} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{61}$$

$$= \begin{bmatrix} 1-2(y^2+z^2) & 2(xy-wz) & 2(wy+xz) \\ 2(xy+wz) & 1-2(x^2+z^2) & 2(yz-wx) \\ 2(xz-wy) & 2(wx+yz) & 1-2(x^2+y^2) \end{bmatrix}$$

The joint angles present in human arm frame are displayed in Fig. 29. Acknowledgement of describing frame $\{B\}$ in frame $\{A\}$ is as follows: $\{B\}$ is coincident with frame $\{A\}$. First, rotate frame $\{B\}$ about X_A by an angle γ, then about Y_A by an angle β, finally, about Z_A by an angle α [86]. The angles γ, β, α are the roll, pitch and yaw angles respectively.

$$\mathcal{R}_{GH}^{f} = \mathcal{R}_Z(\alpha)\mathcal{R}_Y(\beta)\mathcal{R}_X(\gamma) \tag{62}$$

$$= \begin{bmatrix} cos\alpha cos\beta & cos\alpha sin\beta sin\gamma - sin\alpha cos\gamma & cos\alpha sin\beta cos\gamma + sin\alpha sin\gamma \\ sin\alpha cos\beta & sin\alpha sin\beta sin\gamma + cos\alpha cos\gamma & sin\alpha sin\beta cos\gamma - cos\alpha sin\gamma \\ -sin\beta & cos\beta sin\gamma & cos\beta cos\gamma \end{bmatrix}$$

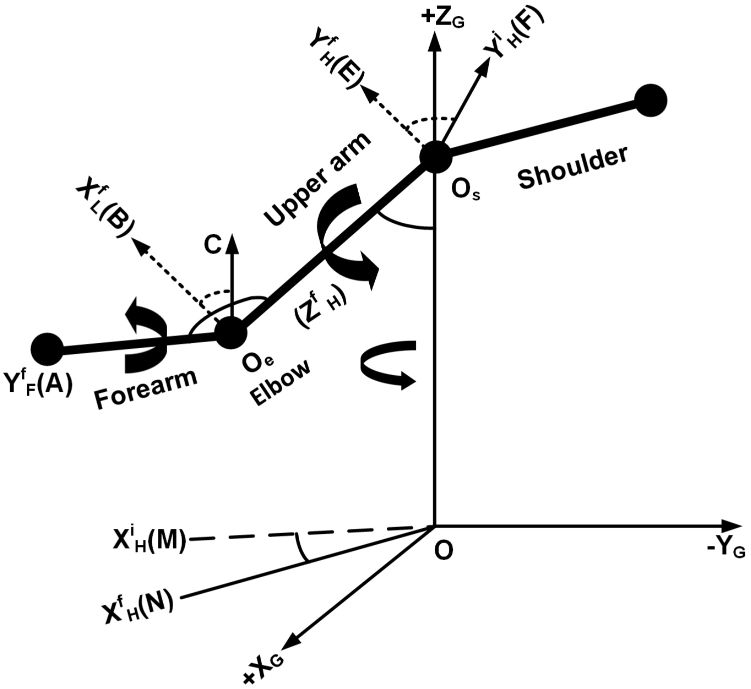


Figure 29: The joint angles present in human arm frame.

With two MYO armbands, we can measure the five joint angles of the operator's arm. From Eq. (61) and (62), the three shoulder joint angles (shoulder flexion/extension, abduction/adduction and internal/external rotation) are calculated. Through the data collected from two MYO armbands, we can calculate these two elbow joint angles (elbow flexion/extension and pronation/-supination).

$$q_{d1} = \arctan(\sqrt{r_{31}^2 + r_{32}^2}/r_{33}) \tag{63}$$

$$q_{d2} = \arctan((r_{23}/\sin q_{d1})/(r_{13}/\sin q_{d1})) \tag{64}$$

$$q_{d3} = \arctan((r_{32}/\sin q_{d1})/(-r_{31}/\sin q_{d1})) \tag{65}$$

where the q_{d3}, q_{d2} and q_{d1} represent the joint angles of shoulder roll, shoulder yaw and shoulder pitch, respectively.

$$\begin{aligned} q_{d4} &= \arccos(\overrightarrow{O_eA} \cdot \overrightarrow{O_eO_s}) && (66) \\ q_{d5} &= \arccos[\overrightarrow{O_eB} \cdot (\overrightarrow{O_eA} \times \overrightarrow{O_eO_s})] && (67) \end{aligned}$$

where the q_{d4} and q_{d5} represent the joint angles of elbow flex and elbow roll respectively. The $\overrightarrow{O_eA}$ is the unit vector along the y-axis of the forearm frame. The $\overrightarrow{O_eB}$ is the unit vector along the x-axis of the forearm frame. The $\overrightarrow{O_eO_s}$ is the unit vector along the z-axis of the humerus frame.

5.2 Neural Networks Controller Design

Let us define the desired joint space trajectory q_d as

$$q_d = [q_{d1}, q_{d2}, q_{d3}, q_{d4}, q_{d5}]^T \in R^5 \tag{68}$$

Now, let us consider apply NN based control technique to achieve the following control of the joint space trajectory. The output signal of the system is required to follow the expected input signal. So responses of the closed-loop system be quick, accurate, and stable. The angle matrix $q_d \in R^5$ exported by the MYO is regarded as the reference signal. and the angle matrix $q \in R^5$ returned by the robot are the actual angles. The dynamic equation of the manipulator is shown in Eq. (69).

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + N(q) = \tau, \tag{69}$$

where $M(q) \in R^{5\times5}$ is the manipulator inertia matrix, $C(q,\dot{q}) \in R^{5\times5}$ is the Coriolis matrix for the manipulator, $G(q) \in R^{5\times1}$ is the gravity terms and $N(q) \in R^{5\times1}$ is unmodeled dynamics caused by exchangeable robot gripper and system uncertainties.

Define $z = \dot{e}_q + \Lambda e_q$, $q_r = \dot{q}_d - \Lambda e_q$, where $e_q = q - q_d$, $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)$. Then the dynamic equation (69) can be rewritten as (70).

$$M(q)\dot{z} + C(q,\dot{q})z + G(q) + M(q)\dot{q}_r + C(q,\dot{q})q_r = \tau - N(q), \tag{70}$$

Design the adaptive controller as (71).

$$\tau = \hat{H}(q) + \hat{M}(q)\dot{q}_r + \hat{C}(q,\dot{q})q_r - Kz, \tag{71}$$

where $\hat{H}(q)$, $\hat{M}(q)$ and $\hat{C}(q,\dot{q})$ are the estimates of $G(q) + N(q)$, $M(q)$ and $C(q,\dot{q})$, respectively.

Then the closed-loop system dynamic can be written as (72).

$$M\dot{z} + Cz + Kz = (\hat{M} - M)\dot{q}_r + (\hat{C} - C)q_r + (\hat{H} - H) \tag{72}$$

The following function approximation method is used.

$$\begin{aligned} M(q) &= W_M^{*\mathrm{T}} S_M(q) + \epsilon_M \\ C(q,\dot{q}) &= W_C^{*\mathrm{T}} S_C(q,\dot{q}) + \epsilon_C \\ H(q) &= W_H^{*\mathrm{T}} S_H(q) + \epsilon_H(z), \end{aligned} \tag{73}$$

where W_M^*, W_C^* and W_H^* are the weight matrices; $S_M(z)$, $S_C(z)$ and $S_H(z)$ are the basis function matrices, and $\epsilon_M(z)$, $\epsilon_C(z)$ and $\epsilon_H(z)$ are the approximation errors.

The basis function matrices are designed as follow.

$$\begin{aligned} S_M(q) &= \operatorname{diag}\left(S_q, \ldots, S_q\right) \\ S_C(q,\dot{q}) &= \operatorname{diag}\left(\begin{bmatrix} S_q \\ S_{\dot{q}} \end{bmatrix}, \ldots, \begin{bmatrix} S_q \\ S_{\dot{q}} \end{bmatrix}\right) \\ S_H(q) &= \begin{bmatrix} S_q^{\mathrm{T}} & \ldots & S_q^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \end{aligned} \tag{74}$$

where

$$\begin{aligned} S_q &= \left[\phi\left(\|q-q_1\|\right)\ \phi\left(\|q-q_2\|\right)\ \ldots\ \phi\left(\|q-q_n\|\right)\right]^{\mathrm{T}} \\ S_{\dot{q}} &= \left[\phi\left(\|\dot{q}-\dot{q}_1\|\right)\ \phi\left(\|\dot{q}-\dot{q}_2\|\right)\ \ldots\ \phi\left(\|\dot{q}-\dot{q}_n\|\right)\right]^{\mathrm{T}} \\ \phi(r) &= e^{-(\epsilon r)^2} \end{aligned} \tag{75}$$

The estimates of $M(q)$, $C(q,\dot{q})$ and $H(q)$ can be written as (76).

$$\begin{aligned} \hat{M}(q) &= \hat{W}_M^{\mathrm{T}} S_M(q) \\ \hat{C}(q,\dot{q}) &= \hat{W}_C^{\mathrm{T}} S_C(q,\dot{q}) \\ \hat{H}(q) &= \hat{W}_H^{\mathrm{T}} S_H(q), \end{aligned} \tag{76}$$

By substituting (76) into (72), we have

$$M\dot{z} + Cz + Kz = \tilde{W}_M^{\mathrm{T}} S_M(q)\dot{q}_r + \tilde{W}_C^{\mathrm{T}} S_C(q,\dot{q})q_r + \tilde{W}_H^{\mathrm{T}} S_H(q) \tag{77}$$

where $\tilde{W}_M^{\mathrm{T}} = \hat{W}_M^{\mathrm{T}} - W_M^{*\mathrm{T}}$, $\tilde{W}_C^{\mathrm{T}} = \hat{W}_C^{\mathrm{T}} - W_C^{*\mathrm{T}}$ and $\tilde{W}_H^{\mathrm{T}} = \hat{W}_H^{\mathrm{T}} - W_H^{*\mathrm{T}}$.

Choose the following Lyapunov function.

$$\begin{aligned} V =& \frac{1}{2}z^{\mathrm{T}}Mz \\ &+ \frac{1}{2}tr\left(\tilde{W}_M^{\mathrm{T}}Q_M\tilde{W}_M + \tilde{W}_C^{\mathrm{T}}Q_C\tilde{W}_C + \tilde{W}_H^{\mathrm{T}}Q_H\tilde{W}_H\right), \end{aligned} \tag{78}$$

where Q_M, Q_C and Q_H are positive definite weight matrices. And the derivative of V is

$$\begin{aligned} \dot{V} =& - z^{\mathrm{T}}Kz - \\ & tr\left[\tilde{W}_M^{\mathrm{T}}\left(S_M(q)\dot{q}_r z^{\mathrm{T}} + Q_M\dot{\hat{W}}_M\right)\right] + \\ & tr\left[\tilde{W}_C^{\mathrm{T}}\left(S_C(q,\dot{q})q_r z^{\mathrm{T}} + Q_C\dot{\hat{W}}_C\right)\right] + \\ & tr\left[\tilde{W}_H^{\mathrm{T}}\left(S_H(q)z^{\mathrm{T}} + Q_H\dot{\hat{W}}_H\right)\right] \end{aligned} \tag{79}$$

The update law is designed as follow.

$$\begin{aligned} \dot{\hat{W}}_M &= -Q_M^{-1}\big(S_M(q)\dot{q}_r z^{\mathrm{T}} + \sigma_M\hat{W}_M\big) \\ \dot{\hat{W}}_C &= -Q_C^{-1}\big(S_C(q,\dot{q})q_r z^{\mathrm{T}} + \sigma_C\hat{W}_C\big) \\ \dot{\hat{W}}_H &= -Q_H^{-1}\big(S_H(q)z^{\mathrm{T}} + \sigma_H\hat{W}_H\big) \end{aligned} \tag{80}$$

where $\sigma_M, \sigma_C, \sigma_H$ are predesigned positive constants. Substituting (80) into (79), we have

$$\begin{aligned} \dot{V} =& - z^{\mathrm{T}}Kz - \sigma_M tr\left(\tilde{W}_M^{\mathrm{T}}\hat{W}_M\right) - \sigma_C tr\left(\tilde{W}_C^{\mathrm{T}}\hat{W}_C\right) \\ & - \sigma_H tr\left(\tilde{W}_H^{\mathrm{T}}\hat{W}_H\right) \end{aligned} \tag{81}$$

Using Young's inequality, (81) can be further relaxed as

$$\begin{aligned}
\dot{V} = & -z^{\mathrm{T}}Kz + \frac{\sigma_M tr(W_M^{*\mathrm{T}} W_M^*)}{2} - \frac{\sigma_M tr(\tilde{W}_M^{\mathrm{T}} \tilde{W}_M)}{2} \\
& - \frac{\sigma_M tr(\hat{W}_M^{\mathrm{T}} \hat{W}_M)}{2} + \frac{\sigma_C tr(W_C^{*\mathrm{T}} W_C^*)}{2} \\
& - \frac{\sigma_C tr(\tilde{W}_C^{\mathrm{T}} \tilde{W}_C)}{2} - \frac{\sigma_C tr(\hat{W}_C^{\mathrm{T}} \hat{W}_C)}{2} \\
& + \frac{\sigma_H tr(W_H^{*\mathrm{T}} W_H^*)}{2} - \frac{\sigma_G tr(\tilde{W}_H^{\mathrm{T}} \tilde{W}_H)}{2} \\
& - \frac{\sigma_H tr(\hat{W}_H^{\mathrm{T}} \hat{W}_H)}{2} \\
= & -z^{\mathrm{T}}Kz + \frac{\sigma_M tr(W_M^{*\mathrm{T}} W_M^*)}{2} - \frac{\sigma_M tr(\tilde{W}_M^{\mathrm{T}} \tilde{W}_M)}{2} \\
& + \frac{\sigma_C tr(W_C^{*\mathrm{T}} W_C^*)}{2} - \frac{\sigma_C tr(\tilde{W}_C^{\mathrm{T}} \tilde{W}_C)}{2} \\
& + \frac{\sigma_H tr(W_H^{*\mathrm{T}} W_H^*)}{2} - \frac{\sigma_H tr(\tilde{W}_H^{\mathrm{T}} \tilde{W}_H)}{2}
\end{aligned} \tag{82}$$

So that we have

$$\dot{V} \leq -\eta V + \kappa \tag{83}$$

where $\eta = \min[2K, \sigma_M/\big(\lambda_{\max}(Q_M)\big), \sigma_C\big(\lambda_{\max}(Q_C)\big), \sigma_H/\big(\lambda_{\max}(Q_H)\big)]$, $\kappa = \frac{1}{2}\mathrm{tr}(\sigma_M W_M^{*\mathrm{T}} W_M^* + \sigma_C W_C^{*\mathrm{T}} W_C^* + \sigma_G W_G^{*\mathrm{T}} W_G^*)$. Since $V > 0$ and κ is the product of the predesigned constants and weight matrixes that we given, as long as $\kappa \leq \eta$, we can have $\dot{V} \leq 0$. According to Lyapunov theory, the system is stable.

5.3 Experimental Results

5.3.1 Experimental Set-up

An operator stands upright with two MYO armbands on the same arm in the lab. The first MYO armband is worn near the centre of the upper arm, and it can measure the orientation of the upper arm. The second MYO armband is worn near the centre of the forearm, and it is used to estimate the orientation of the forearm by the build-in gyroscope and calculate the wrist joint angles by eight bioelectrical sensors. The experiment setup with the initial pose of the robot is shown in Fig. 30.

Before the operator controls the virtual robot arm, the MYO armband must be calibrated and the EMG sensors must be warm up so that the MYO armband can discern different hand poses. The operator should stand still before the Baxter completes mimicking the human arm's configuration, and his/her arm should not move too fast through the experiment.

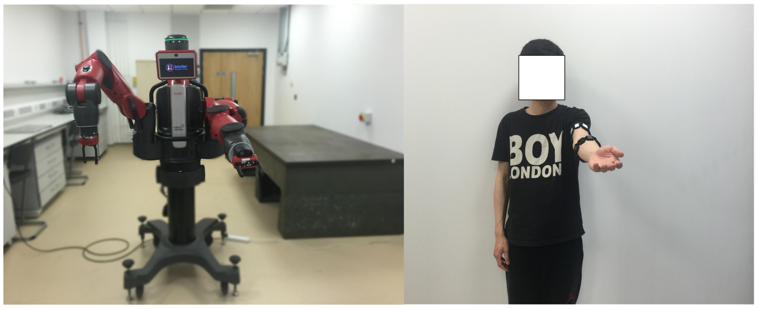


Figure 30: The experiment setup with the initial pose of the robot.

5.3.2 Results of Experiment

The norms of weights relating to each joint are given as shown in Fig. 31. It is seen that, with compensation, the parameter estimates are smooth. Fig. 32 represents the position difference of the five joints between the robot trajectory and the reference one given by Myo armband. The left row of Fig. 32 was generated without NN learning, while the NN enhanced controller was applied in the right row of Fig. 32. It can be seen that the tracking performance of the manipulator's joints has been improved after adopting the NN learning control. The model of using MYO armband to control Baxter robot arm was designed and tested. The proposed NN learning controller was also tested, and the results show its effectiveness of compensating the unknown dynamics during teleoperation.

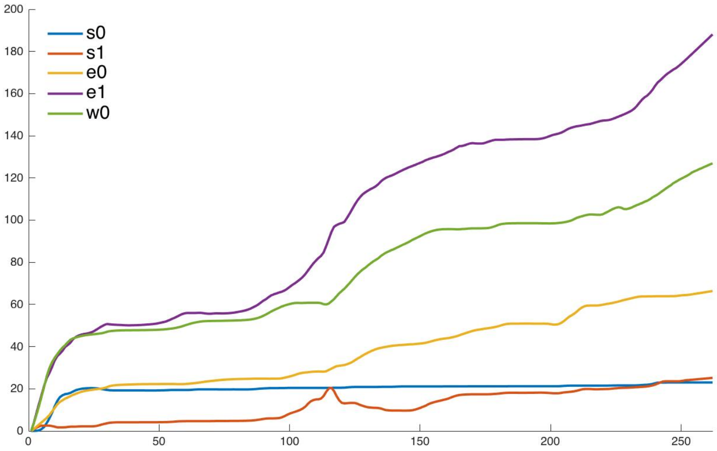


Figure 31: NN learning weights for every single joint. s0, s1, e0, e1 and w0 represent the five robot joints used in this experiment, respectively.

5.4 Conclusion

In this published chapter [87], an NN learning enhanced teleoperation control of the Baxter robot is developed. The motion of the human operator's arm can be detected and reconstructed by using physical IMU signals, provided by a pair of MYO armbands worn on the operator's arm. Then the data was applied to Baxter robot's joints respectively for teleoperation. The advantages of using the MYO armband are that MYO armband is portable and the calculated angles of shoulder joint and elbow joint are accuracy. NN learning based compensation mechanism for the controller helps the user to overcome the effect of the uncertainties associated with the telerobot model and environment while teleoperating. The proposed controller guaranteed the system output tracking errors satisfied the prescribed transient and steady-state control behaviour bounds. Experimental tests have demonstrated the effectiveness of the proposed design techniques.

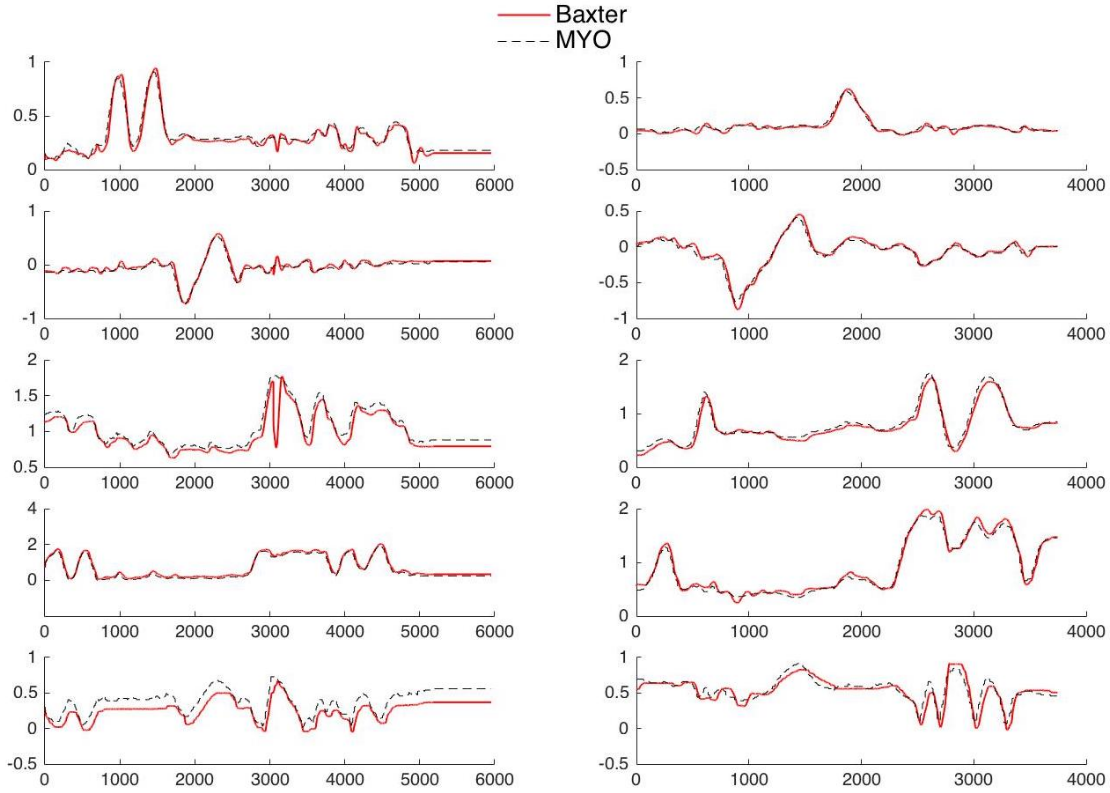


Figure 32: Robot trajectory compares with MYO reference point. Left: Without NN, Right: With NN

6 Development of an Immersive Interface for Robot Teleoperation

In this section, various techniques have been investigated in order to enhance user experience (UX) for robot teleoperation. In this teleoperation system design, the human operators are provided with both immersive visual feedback and intuitive skill transfer interface such that when controlling a telerobot arm, a user is able to "feel" in a first-person perspective in terms of both visual and haptic sense. Some high-tech devices including Omni haptic joystick, MYO armband, Oculus Rift DK2 headset, and Kinect v2 camera are integrated. The surface electromyography (sEMG) signal allows the operator to naturally and efficiently transfer his/her motion skills to the robot, based on the properly designed elastic force feedback control. For visual feedback, operators can control the pose of a camera on the head of the robot via the wearable visual headset, such that the operator can perceive from the robot's perspective. Tests involving human subjects have been performed to evaluate the design, and the experimental results have shown that superior performance and better UX have been achieved by the proposed method in comparison with the traditional methods.

6.1 Introduction

With the advance of automation technologies, robots have become increasingly prominent in our everyday lives [88]. Autonomous robots have become in common use in industrial, as these robots can complete high precision tasks with minimal completion time, especially within hazard environments. However, specialised tasks that require extensive interactions with uncertain environment, e.g., medical operation which is not suitable for fully autonomous robots to be applied to. A tele-controlled robot remotely controlled by human operators may be desired for these tasks [89], i.e., teleoperation. Works demonstrated in [90] shows how stereo vision contributes to improving the performance of applications in mobile robot teleguide. [91] presents a non-contacting vision-based method of robot teleoperation which improves the accuracy of object gripping tasks. Teleoperation has progressed to the point, where human operators can easily control a multi-jointed robotic arm [1, 85, 92, 93], via a haptic feedback device. As introduced in [94], with haptic force feedback, a pilot can have a natural representation when approaching obstacles. Force feedback was also added in [95] to aid camera system avoiding the obstacle in mobile robot teleoperation. It is natural to combine haptic information with visual information for coordinated information feedback. However, conventional means of visual feedback using a monitor do not provide an operator with an immersive user experience.

Immersive visual feedback technique has been utilised in many research projects to provide an operator with a sense of telepresence [96] [97], i.e., the feeling that the operator exists at the robot's remote location. An Oculus Rift headset is used in this work to give an immersive visual feedback of the workspace of a robotic arm from a fixed position stereo camera in [98]. However, this system does not allow an operator to manipulate the orientation of the stereo camera. An underwater simulation environment is created in [99], and it allows an operator to control the orientation and operators view of a simulated ROV (Remotely Operated Vehicle) via an Oculus Rift headset. In comparison to this design, our design is based on a physical platform. By combining haptic feedback, and visual feedback, an operator would naturally experience telexistence [96]. However, none of the research mentioned above investigates whether there is a link between operator task performance and operator immersion whilst teleoperating. Therefore, extensive tests have been performed in this work to compare operation-friendliness with or without the proposed immersive interfaces.

To enhance the telepresence, in this work, an immersive visual feedback device Oculus Rift DK2 is employed, together with a Kinect v2 RGB-D imaging device, and haptic device Geomagic Touch, to teleoperate a Baxter robot arm.

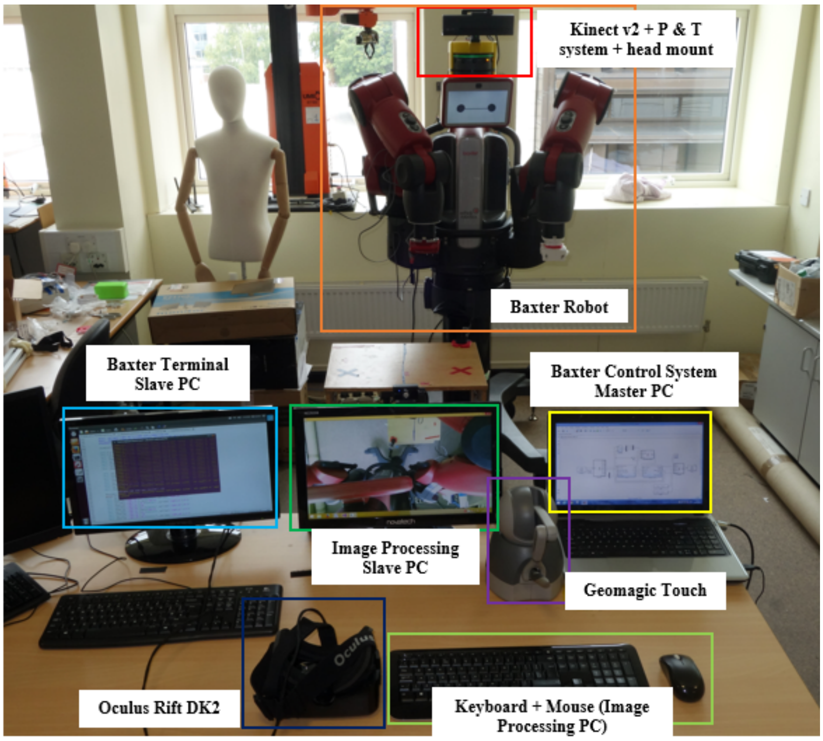

Kinect v2 + P & T system + head mount
Baxter Robot
Baxter Terminal Slave PC
Baxter Control System Master PC
Image Processing Slave PC
Geomagic Touch
Oculus Rift DK2
Keyboard + Mouse (Image Processing PC)

Figure 33: An illustration of the teleoperation setup, with each key component labeled, including a Kinect head mount system, a Baxter robot, a Geomagic Touch and an Oculus Rift DK2 [photo taken at Plymouth University].

Haptic feedback devices which give an operator force feedback from a telerobot have been utilised extensively among researchers [100, 101]. In this work, using IR sensor at the cuff of Baxter's end-effector(Fig. 34), the haptic feedback information is provided as the distance between an object/surface and Baxter's gripper. This haptic force feedback is also bonded with two other customised force feedbacks, tactile force, and restoration force. Tactile force is derived from a force sensing resistor (FSR) mounted onto one of the pincers of Baxter's end-effector gripper (referred to as the FSR pincer, Fig. 34), which allows an operator to sense the firmness of objects picked up by the gripper. Restoration force tends to drag the haptic joystick back to the pre-set initial pose, after an operator has used the haptic joystick to change the position of the robot's gripper's pincers. Force feedback is presented to an operator via a Geomagic Touch haptic feedback joystick (Fig. 33), which is also used by the operator to control the position of Baxter's end-effector.

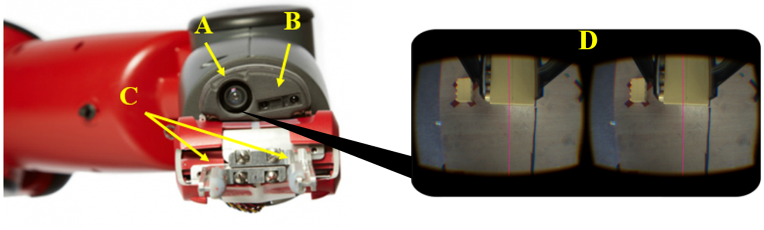


Figure 34: An illustration of one of the Baxter robot's end-effectors. A: the cuff camera; B: the IR range sensor; C: active, gripper attachment [2]; D: VR headset capture of the operator's view of the robot workspace, captured by the cuff camera

Immersive visual feedback technique which provides an operator a sense of telepresence has been implemented in many research projects [96, 102]. In [103], a panoramic display system was used for improving the teleoperation performance of a mining robot. A virtual reality (VR) headset, Oculus Rift, was used in [104] to control a simulated robot in a combat scenario. An intuitive 3D interface was designed to allow operator immerse in a virtual environment while teleoperating [105]. Rather than using the conventional interface composed of a monitor and a keyboard [106], immersive visual feedback technique can give operators a feeling that they are controlling the robot as sitting inside a robot. In this work, the output image stream after image processing is displayed on the Oculus Rift DK2 headset. The operator is able to manipulate the gaze of the Kinect camera's field of view (via a servo pan and tilt kit), as they move their head in a manner which is naturally similar, towards a certain direction for a better view. The IMU unit built into the Oculus Rift headset records the operator's head motion, and the pan and tilt servo platform, upon which the Kinect camera is fixed, tracks the recorded motion. The Kinect pan and tilt servo platform, which is then mounted onto the head of Baxter, enables an operator to feel as though they are looking at the robot's workspace from a static position, in place of the robot.

In order for the operator to control the gaze of visual feedback camera, i.e., Kinect v2, a pan and tilt servo mount has been designed and manufactured, and fixed on top of Baxter's head. This also could be beneficial to other research carried out on the Baxter robot platform in the future. Baxter is a robust, reliable research robot platform, and researches that have already conducted on this platform has shown promising and interesting results. To the best of our knowledge, it is the first attempt among the teleoperation community to implement a system which combines haptic and immersive visual feedback on this platform. Extensive tests have been carried out, based on our previous work [107], to compare the user experiences of non-immersive and immersive teleoperation interfaces and their performances.

6.2 System Communications

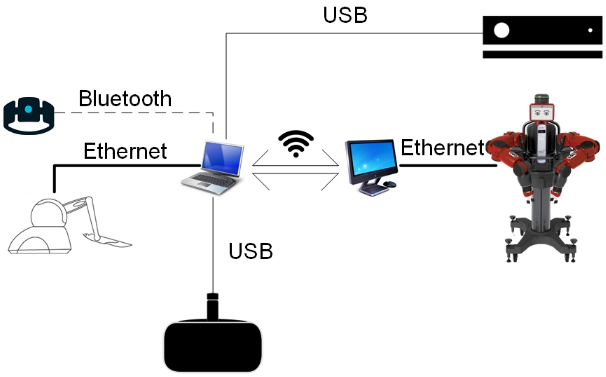


Figure 35: The illustration of the teleoperation system.

To enhance the telepresence, in this work, Oculus Rift DK2 that could provide vision from robot's perspective is employed (Fig. 37(a)), together with haptic device Omni (Fig. 33), to teleoperate a Baxter robot arm (Fig. 35). This haptic device is of 6 degree-of-freedom (DOF), which allows the user to perceive the obstacle and environment-related forces. The forward kinematics, inverse kinematics and Jacobian of the Omni haptic device has been well studied by [55]. In addition, infrared (IR) sensor built into the cuff of Baxter's end-effector are used to provide haptic feedback information to an operator about the distance between the robot's gripper and an object/surface (referred to as *proximity force*). The whole haptic feedback is combined with two force feedbacks: referred to as *proximity force* (Fig. 38), and *elastic force* (Fig. 40), respectively, to provide a compound haptic feedback to the operator. And then, by wearing MYO armband (Fig. 37(b)), the sEMG signal was collected from the operator and used to transfer skills from the human operator to the robot. A brief user manual of the teleoperation system can be found in Appendix A.

6.2.1 Kinect

Kinect (Fig. 36) is a line of motion sensing input devices by Microsoft for Xbox 360 and Xbox One video game consoles and Windows PCs. Based around a webcam-style add-on peripheral, it enables users to control and interact with their console/computer without the need for a game controller, through a natural user interface using gestures and spoken commands. It is basically a depth camera. Normal webcams collect light that is reflected by the objects in its field of view and turn them into an image. Kinect can also measure the distance between the objects in front of it and itself. Then it displays the depth data in form of image using grey-scale values. By fusing the 2-D image and depth informations, Kinect can output a less memory-consuming depth image that contains 3D information of objects in front of it. The depth image, compared with the 3D cloud points, are easier to process and more suitable for real-time application.

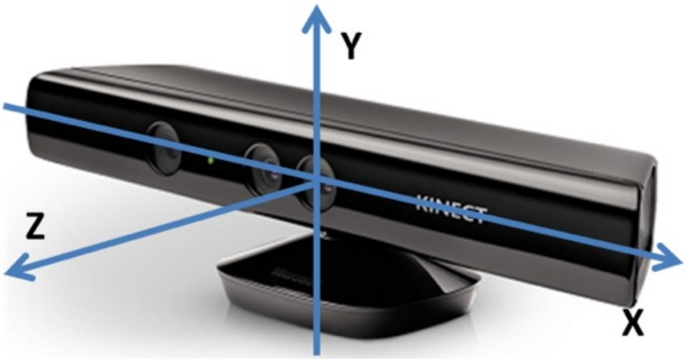


Figure 36: Kinect sensor with its coordinate system.

6.2.2 Oculus Rift

Oculus Rift is a head-mounted display, which uses the position tracking sensor and optical lens to consist a system. With it, the wearers can feel an illusion that they are actually in a virtual environment. In particular, Oculus Rift Development Kit 2 (DK2) is used in this work, which has developed four key improvements compare with DK1. The DK2 has higher resolution (960×1080 per eye), low-persistence OLED display, higher refresh rate and positional tracking function.

(a) Oculus Rift DK2 (b) Myo Armband

Figure 37: Oculus Rift DK2 and Myo armband.

6.2.3 MYO Armband

MYO armband, developed by the Thalmic Labs company, is a wearable device, which can recognise the hand gesture and arm movement by wearing it on the arm. Based on the muscles' movement, the eight built-in EMG sensors, along with a 9-axis IMU sensor can identify the hand gesture and send the signals to the system via Bluetooth. The sensors generate data by electrical impulses from arm muscles. Given that each user has a different type of skin, muscle size and etc, it is necessary to calibrate the device before each different user starts using. It can help to increase the accuracy of the gesture recognising performance.

6.3 Immersive Visual Feedback

In this section, the proximity force and the 3D stereo vision are introduced. As by combining these two tactile and vision feedback, the system can provide the user a more immersive teleoperation experience.

6.3.1 Proximity Force

In order for the operator to sense that an object is close to the gripper, it is required that the pitch and yaw of the end-effector are known. By adding an extra point (P_W in Cartesian space) to the wrist of Baxter's right arm, it is possible to determine the direction that the end-effector is pointing to in Cartesian space, which is represented as a resultant vector in Eq. 84:

$$P_s \overset{\text{def}}{=\!=} \begin{bmatrix} x & y & z \end{bmatrix}^T; P_{offset} \overset{\text{def}}{=\!=} P_W - P_E \tag{84}$$

where P_s is any Cartesian point in Baxter's workspace, including x, y, z coordinates, P_W is the added wrist point, P_E is the end-effector point, and $P_{offset} \in R^3$ is the resultant vector. The magnitude of the resulting vector is determined by the output value of the IR sensor on the end-effector of the Baxter robot. The range of distances detectable by the end-effector IR sensor is 40mm (closest) to 400mm (furthest). Therefore, before the range value (scalar) being multiplied by the end-effector direction vector, the detected distance must be subtracted from the maximum detectable distance, since the closer the object to the gripper, the greater the magnitude of the force feedback the operator should experience, and conversely the further away the object/surface, the smaller the magnitude of the force feedback the operator experiences. The resulting Cartesian force vector is analogous to a spring, between the end-effector IR sensor and the detected object (as shown in figure 38). The resulting Cartesian force vector is determined by Eq. 85:

$$F_p = P_{offset} \cdot k\left(d_{IRmax} - d_{object}\right) \tag{85}$$

where $F_p = \begin{bmatrix} F_{px} & F_{py} & F_{zp} \end{bmatrix}^T$, k is a distance gain, $d_{IRmax} = 400mm$, and d_{object} is the IR-sensed distance to the object.

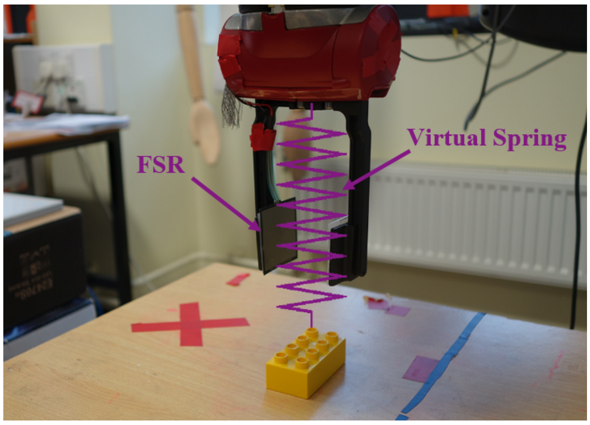


Figure 38: A description of the spring-like behaviour of the object proximity force feedback (derived from end-effector IR sensor measurements, equation 85); an Interlink Electronics 406 force sensing resistor is attached to the left gripper pincer.

6.3.2 Visual feedback from the robot's perspective

A Kinect v2 device is mounted onto the head of Baxter to give the operator a complete view of the robot's workspace, and a 3D printed mount was designed to fit directly onto the head of Baxter, as shown in Fig. 39(a). The Kinect v2 is mounted upon a 3D printed support bar, which fixes to an aluminium bracket. The aluminium bracket is fixed to a Dynamixel MX28R servo (tilt servo), which mounts on the top of a Dynamixel MX28R servo (pan servo). *Controlling Kinect's View:* The positions of both pan and tilt MX-28R servos are controlled by an Arduino Micro-controller board. The operator can view the entirety of Baxter's workspace through the Oculus Rift (via the Kinect v2) by changing the orientation of their head (Fig. 39). This gives the operator the ability to control the direction of the Kinect device from their head position.

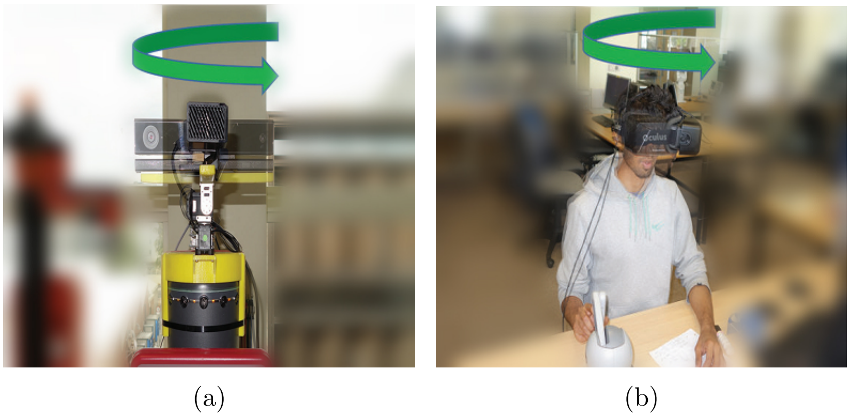


Figure 39: (a) The movement of the Kinect camera, determined by the head movements of the operator. (b) The natural head movements of the operator, tracked by the headset's built-in IMU.

Rendering on Oculus Rift: To display images on the screen of the Oculus Rift headset: a program is used to implement D3D11 to create a 2D texture from the supplied colour image (Kinect or end-effector camera), and then present this to the Oculus. This simple method, however, will apply distortion, and scaling to the supplied texture, and will lead to reducing motion blur. Applying a distortion to original image cancels distortion contributed by the Oculus Rift's lenses. The final image is presented to each of the operator's eyes via the headset's screen. The 1920 x 1080 pixel screen is split into two halves (each 960 x 1080 pixels), both halves are presented with the same image in this system.

6.4 Intuitive Skill Transfer

In this section, the elastic force feedback and the extraction of muscle activation are introduced. By applying elastic force feedback on the haptic device, the operator will naturally generate force during the remote control in order to counter the feedback force so that the user can transfer his/her skills more intuitively to the robot.

6.4.1 Elastic Force Feedback

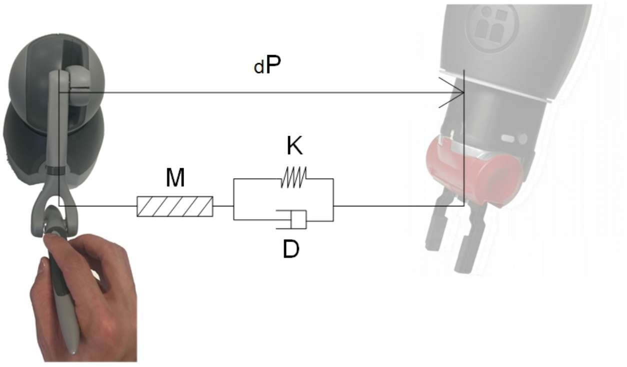


Figure 40: Force feedback generation.

Rather than the measuring interaction forces in traditional ways, such as mounting a force sensor onto the robot manipulator, we estimate the interaction forces by a haptic rendering algorithm. The algorithm is obtained by the tracking error during motion. In Fig. 40, we assume that the interaction force is governed by an admittance model given in Eq. (86):

$$F_e = M(d\ddot{P}) + D(d\dot{P}) + K(dP) \tag{86}$$

where K is a virtual spring's stiffness, D is a virtual damper's damping ratio, M is a virtual mass and dP is the position difference between the Omni joystick's commanded reference position and the Baxter end-effector's actual position. In this work, we only consider the three dimensions translational motion for simplicity, and $dP = P_s - P_m$, where $P_s \in R^3$ is the actual translational position of Baxter's end-effector and $P_m \in R^3$ is the Omni device's reference position. Then, we can formulate the torque of the Omni as below:

$$\tau_o = J_o^{\mathrm{T}}(F_e + F_a) \tag{87}$$

where τ_o is the torque of the Omni device, and J_o is the Jacobian matrix of the Omni device.

$$F_a = J^T \cdot \begin{bmatrix} F_{py} & F_{pz} & F_{px} \end{bmatrix}^T \tag{88}$$

where the force vector is rearranged to match the Baxter and Omni's coordinate frames, $T = \begin{bmatrix} T_{J1} & T_{J2} & T_{J3} \end{bmatrix}^T$, and $J \in R^{3\times3}$ is the Jacobian matrix. The elastic force generated by the haptic rendering algorithm (86) can help human operator to improve awareness of the tracking performance, i.e., combining visual sensing into the tactile sensing. Above all, the human operators will subconsciously apply an increase force on their arm so as to counter the increased feedback force on the haptic device, since the feedback force will tend to pull/push the user's hand according to the difference between actual moving direction and the lagged robot end-effector position or overshoot position. Thus, the control gain will increase or decrease respectively according to the sEMG signals, and accordingly, the end-effector will accelerate or slow down to match the user's action.

6.4.2 Extraction of Muscle Activation

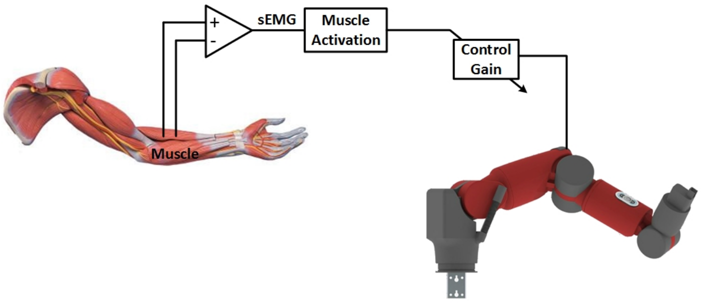


Figure 41: Human Arm stiffness Mapping to Robot Joint.

Fig. 41 shows how human arm stiffness is mapped to robot joint. The estimation of muscle activation level from raw sEMG signals will impact the performance of the human-robot skill transfer. Therefore, extracting muscle activation levels from the collected sEMG signals is an essential step for the experiment. At first, all the eight channels of raw sEMG signals obtained from the MYO armband will be squared and low-pass filtered (see Fig. 42), according to our previous work [1].

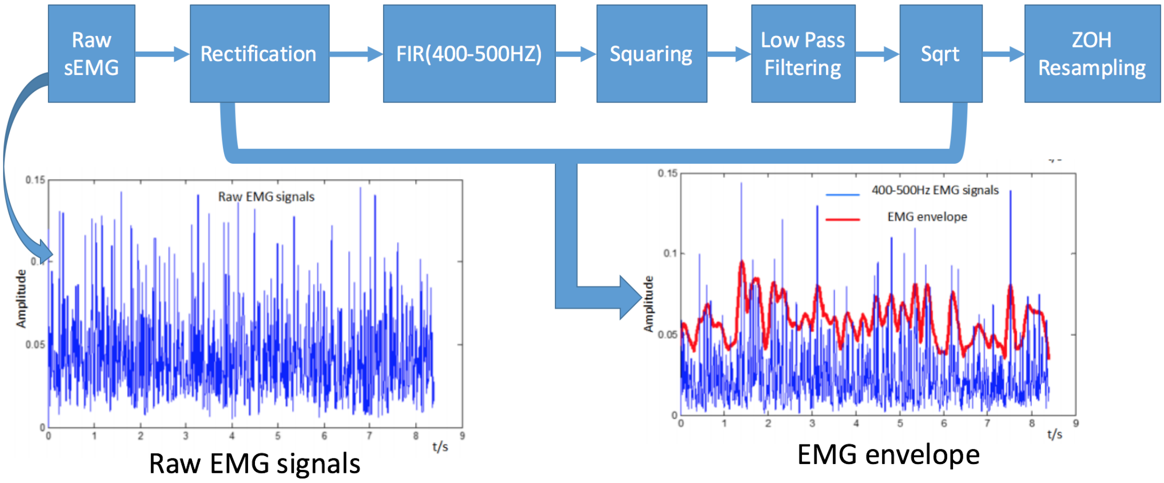


Figure 42: sEMG signals processing procedure by using Squaring and Low-pass Filter.

Then recursive filter is applied to $e(t)$, the rectified, normalised and filtered sEMG signals, and determine the neural activation $u(t)$. sEMG was represented below showing its time-varying feature [108]:

$$u(t) = \alpha * e(t-d) - \beta_1 * u(t-1) - \beta_2 * u(t-2) \tag{89}$$

where d is the electromechanical delay, and α, β_1, β_2 are coefficients of a second-order dynamics system. In order to make sure (89) always have a positive stable solution, we must have:

$$\beta_1 = \gamma_1 + \gamma_2 \quad \beta_2 = \gamma_1 * \gamma_2 \quad \alpha - \beta_1 - \beta_2 = 1 \tag{90}$$

where $|\gamma_1| < 1$, and $|\gamma_2| < 1$.

Then, as we explained in our previous paper [61], there is a nonlinear relationship between frequency and force for single motor units. And here is the nonlinear mapping method from $u(t)$ to muscle activation $a(t)$ [108]:

$$a(t) = \frac{e^{Au(t)} - 1}{e^{A} - 1} \tag{91}$$

where $A \in [-3, 0]$ is the nonlinear shape factor. It can affect muscle activation level from $A = -3$ being highly exponential to $A = 0$ being linear.

In addition, if the control gain is set proportionally to the muscle activation level, control rate, and stiffness variations could result in instability and incompatibility of the system. Therefore, we normalise the control gain κ at the tth sampling instant within the specified range as:

$$\kappa(t) = (\kappa^{max} - \kappa^{min}) \frac{(a(t) - a^{min})}{(a^{max} - a^{min})} + \kappa^{min} \tag{92}$$

where κ^{max} and κ^{min} are the stable robot motion's gain range, a^{max} and a^{min} are the maximum and minimum muscle activation. These parameters can all be obtained experimentally beforehand.

6.5 Evaluation of System Performance

The system was evaluated in two different scenarios, visual immersion and intuitive skill transfer. The basic system communication interfaces of two scenarios were setup as shown in Fig. 35. The Baxter robot was connected to the Baxter terminal slave PC. The Kinect, Omni, Oculus, and MYO were all connected to the master PC. The master PC and slave PC were connected wirelessly within the same local network. The VR headset, Oculus, was used in the first scenario to test how the visual immersion levels would affect the teleoperation experiences. The sEMG device, MYO armband, was wore in the second scenario to verify how variable control gain help operators to gain a better control during teleoperation.

6.5.1 Test of Visual Immersion

Twelve subjects participated in testing, where they were asked to complete a simple pick up and place task, which involved stacking 3 Lego blocks on top of one another and picking up one block at a time, as shown in Fig. 43. This task was repeated for three different types of teleoperation systems: In the first system, the operator who controlled the Kinect's position with the headset, was given force feedback when controlling the position of Baxter's right end-effector, and could switch between the Kinect and cuff camera's view. In the second system, the operator was not presented with any force feedback when controlling the position of Baxter's right end-effector. In the third system, the operator did not wear the Oculus visual headset. Instead, the Kinect images were presented to the operator via an LCD monitor. The testing environment is shown in Fig. 43.

Operators were asked to complete a questionnaire after completing the task for each of the systems mentioned above, which asked the operator to rate various aspects of each system. When the operator was completing each task, the number of times that any part of the end-effector collided with either the testing table or Lego blocks was recorded as shown in Fig.44.

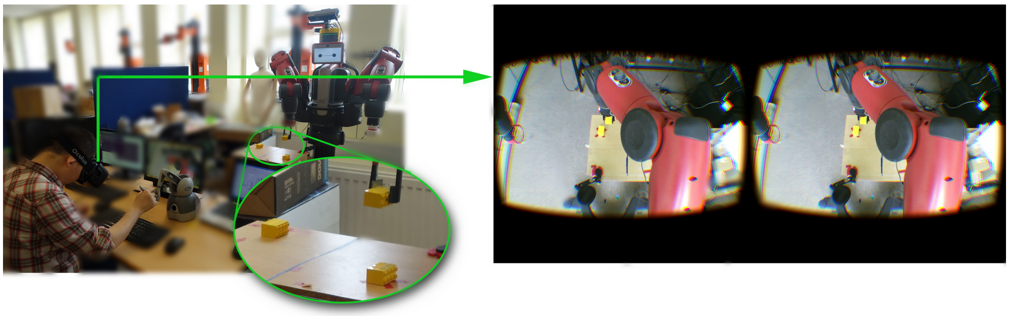


Figure 43: An operator attempts the pick-and-place task using the Oculus Rift, and haptic controller (force feedback enabled)

According to the test results, 67% of the participant operators preferred the use of the Oculus Rift to supply visual feedback, instead of an LCD monitor. The remaining 33% preferred the LCD monitor due to discomfort when viewing images through the Oculus Rift headset.

Several operators expressed that they experienced difficulties when determining distance of objects from Baxter's right end-effector when viewing Baxter's workspace via the Kinect. Coincidentally, 92% of operators said they prefer utilising the combination of Kinect and cuff camera views to pick up and place objects when completing the given tasks, rather than solely viewing images from the Kinect.

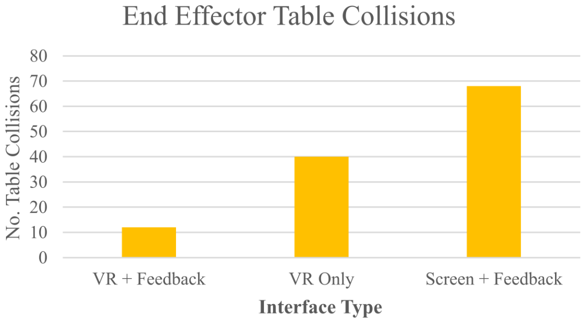


Figure 44: Correlation between interface type and total number of end effector table collisions; based on all 12 participants' experimental data.

Apparently, seen from Figs. 44 and Fig. 45, the number of end-effector collisions and the level of immersion experienced by the operators is relevant. As the operator's sense of immersion decreases, the number of end-effector table

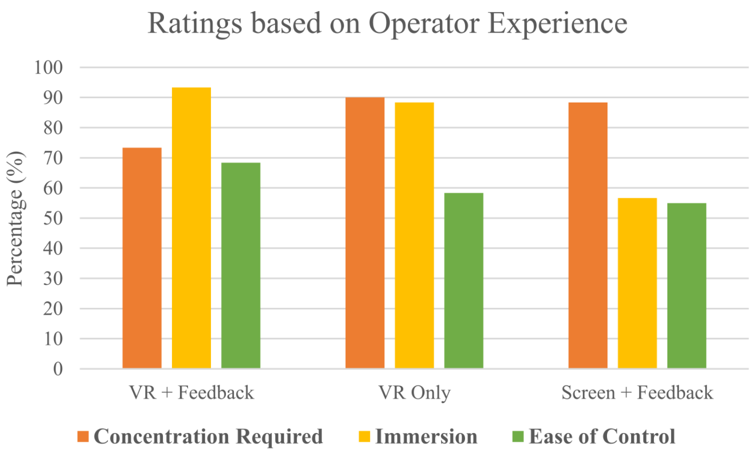


Figure 45: Average experience ratings of each system variant, reported by all participating operators.

collisions increases. The lack of deep sense when viewing the robot workspace via the LCD screen or third-person visual headset causes the operator to collide with the test table. Additionally, the lack of sense of depth the operator experiences via visual feedback, can also be improved by end-effector object proximity haptic feedback.

After comparing third-person visualisation and haptic feedback with screen & feedback (Fig. 45), it can be seen that the immersiveness of the operator's experience can affect the concentration required for the operator to complete the given task. The average of operators' third-person visual experience is at 93% when using the Oculus Rift headset and 57% when presenting Baxter workspace images via the LCD screen relate to average operator, required concentration ratings of 73%, and 88% respectively. The increase in the concentration required for the third-person visualisation only system compared to the third-person visualisation and haptic feedback system (Fig. 45), can be attributed to the lack of force feedback in the third-person visualisation only system. 92% of tested operators agreed that the haptic controller force feedback was helpful when moving the controller joints back to their respective initial positions, after the operator has switched from the cuff camera to the Kinect camera view, or the operator has finished changing the position of the right end-effector gripper pincers. However, 58% of those who agreed, commented that the force feedback was too stiff and uncomfortable at times, i.e. the force feedback would sometimes spike, giving the operator a large force for a small period of time (approximately 0.1 seconds).

6.5.2 Intuitive Skill Transfer

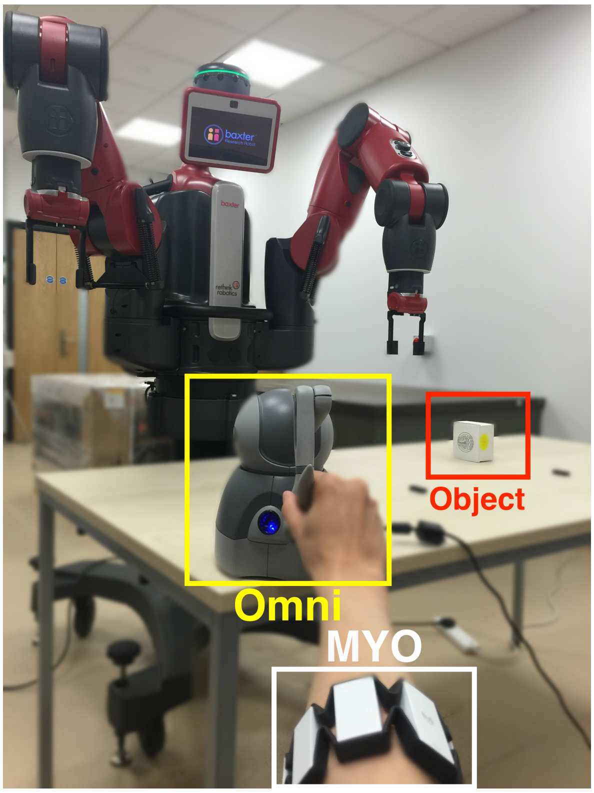


Figure 46: Designed Pick-up and Place task setup.

In this set of the experiments, the manipulator's tracking speed can be adjusted online. A pick-up and place task is designed to verify whether the sEMG based teleoperation can help the user to have a more intuitive teleoperation experience and improve the tracking performance in terms of efficiency and accuracy. In the experiments, the global task objective is to pick up the target object from a start position, and then place it to a target position, as shown in Fig. 46. Different types of control gains will be applied to each task.

The control gain was set to be high, low and variable as extracted from user's sEMG in a different iteration of the experiment for comparative studies. In

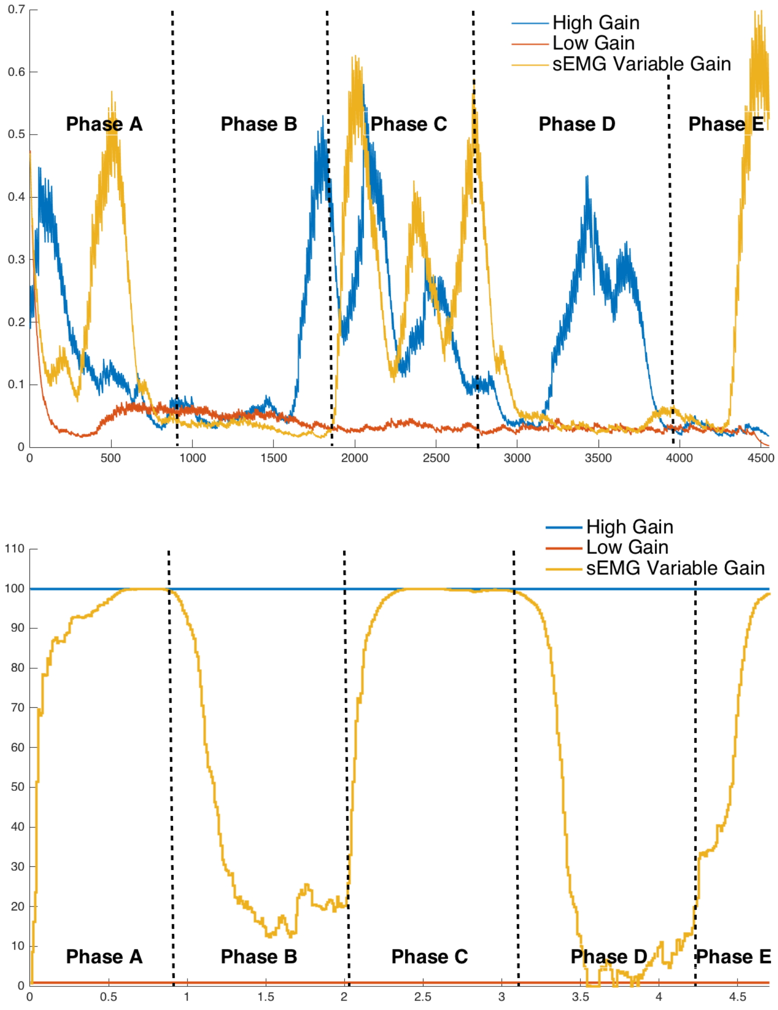
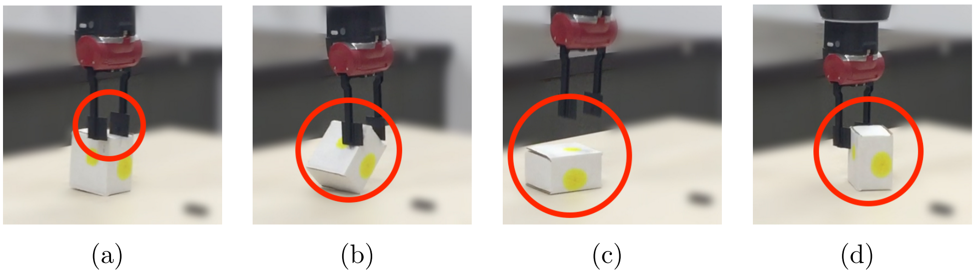


Figure 47: a. End-effector Speed of Baxter robot. b. Control gain value of the Baxter robot. Phase A: Approaching the object. Phase B: Picking up the object. Phase C: Moving to the placing area. Phase D: Placing the object. Phase E: Moving back to the initial position.



(a) (b) (c) (d)

Figure 48: Set of screenshots of the recorded test tries under high gain control mode.

Fig. 47, the blue line represents the tracking speed corresponding to a high control gain. The tracking performance was good due to its high following speed. However, it is difficult to perform high accuracy required movement such as pick-up the target object. As we can see from a set of screenshots of the recorded test trials (Fig. 48), the manipulator will either miss the object or knock over/move the object from its original position.

The yellow line in Fig. 47 represents the tracking speed when the control gain is sEMG variable based. In Phase A, when we need to approach the object quickly, the control gain immediately increased and almost reach the same level as in the high gain mode. Next, it dropped to a controllable level when picking up the object, in Phase B. It raised back to a high level after the object was picked up and remain the level until the operator controlled the robot arm move to the placing area, as shown in Phase C. Then, the control gain immediately dropped again when the operator did the place manoeuvre, as in Phase D. Finally, the control gain raised back when the operator moved the manipulator to its initial position, as in Phase E.

The red line in Fig. 47 shows the tracking speed under low control gain. The manipulator was able to finish the task, and the performance was accurate due to its relatively slow speed. However, it is time-consuming for finishing the same task compare with the previous two control gain set. The time consumption of finishing the entire task in the sEMG based variable gain mode almost matches that in the high gain mode, while it is two times longer than that in the low gain mode.

With these comparative experiments, the sEMG variable gain control mode demonstrates a more accurate, smooth and efficient teleoperation experience than the other two modes. Furthermore, it is more intuitive for the untrained, unskilful operators to transfer their skills to the robot.

6.6 Conclusion

In conclusion, visual feedback from the robot's perspective is provided combined with compound haptic feedback for teleoperating a robot arm. sEMG signals are collected to vary the control gain of the manipulator. Using Oculus Rift headset, the visual feedback enables an operator to view robot workspace in the robot's perspective, which greatly assists the operator in concentrating on completing the task while reducing the cognitive load on the user, since the operator is able to move their head in a natural way to manipulate the position of the camera on robot's head. Wearing Myo armband, the sEMG signal enables the operator to teleoperate the robot more intuitively, since the control gain will be changed while the user is naturally countering the force feedback from the haptic device. Experimental results show that most tested operators preferred using the Oculus Rift headset for visual feedback of the robot workspace, and the tracking performance becomes better when sEMG is applied. The visual and haptic feedback techniques investigated in this work help to improve user experience in both immersive and intuitive ways, and can be applied in similar teleoperation setup. This chapter's work has been published in [109].

7 Conclusion

7.1 Summary of Thesis and Contributions

This thesis is a research of developing control methods and designing interface for HRI. The works have been carried out aiming to enhance the UX during teleoperation in both immersive and intuitive ways. The contributions of the thesis are reviewed, and the research work results are summarised.

In Chapter 3, the proposed physiological signals enhanced teleoperation strategy is implemented in both position control mode and torque control mode. The former mode enables the human operator to adjust motion tracking speed through muscle contraction, while in the latter one, the stiffness of the robot arm could be set through muscle activation directly. The proposed method greatly improve the user's ability to manipulate a telerobot for a delicate task, especially in the presence of motion amplification of the teleoperation system. The effectiveness of the proposed approach has been tested and confirmed by comparative experiments where a 7-DOF (degrees of freedom) robotic arm of Baxter® robot is employed.

Chapter 4 presents a novel combination of visual servoing (VS) control and neural network (NN) learning on a humanoid dual-arm robot. A VS control system is built with stereo vision obtaining the 3D point cloud of a target object. A least square based method is proposed to reduce the stochastic error in workspace calibration. An NN controller is designed to compensate for the effect of uncertain payload and other internal or external uncertainties during the tracking control. In contrast to the conventional NN controller, a deterministic learning technique is utilised in this work, to reuse the neural learnt knowledge before current dynamics changes. A skill transfer mechanism is also developed to apply the neural learnt knowledge from one arm to the other, to increase the neural learning efficiency. The moving trajectory of the object tracked by the proposed VS algorithm provides target positions to the coordinated dual arms of a Baxter robot in the experimental study. Robotic implementations have demonstrated the efficiency of the developed VS control system and have verified the effectiveness of the proposed NN controller with knowledge-reuse and skill transfer features.

In Chapter 5, the neural learning enhanced control techniques has been successfully applied to teleoperate the Baxter robot manipulator. A pair of MYO armbands worn on the operator's forearm and upper arm are used to teleoperate the robot manipulator. Preliminary experiments are carried out to test the desire performance and the results are given.

In Chapter 6, we have investigated various techniques that can be used to enhance the user experience for robot teleoperation. In our teleoperation system design, the human operator is provided with both immersive visual feedback and intuitive skill transfer interface such that when controlling a telerobot arm, a user is able to "feel" in a first-person perspective with both visual and haptic sense. Some high-tech devices including Omni haptic joystick, MYO armband, Oculus Rift DK2 headset, and Kinect v2 camera are integrated. The surface electromyography (sEMG) signal allows the operator to naturally and efficiently transfer his/her motion skill to the robot, based on the properly designed elastic force feedback. For visual feedback, operators can control the pose of a camera on the head of the robot via the wearable visual headset, such that the operator is able to perceive from the robot's perspective. Tests have been performed with human subjects to evaluate the design, and the experimental results have shown that superior performance and better user experience have been achieved by the proposed method in comparison with the traditional methods.

7.2 Proposals for Future Works

Due to the restriction of the university's network, long distance remote control experiments were not conducted. Consequently, how much the communication lag will affect the performance of remotely controlling a robot is not explored in the research work of this thesis. Therefore, discussions about the topic of communication lags are expected in future work. For the other, in the future more mature technologies and various types of sensors are also expected to be emerged into the HRI teleoperation framework. An electroencephalogram (EEG) sensor would be a good example to let operators control robot as a part of their own body. Moreover, deep learning used in designing controller in this paper can be also used on image processing part of future systems. We believe that these would allow telebots to increase autonomous level and operators to use more user-friendly commands to control a robot.

Another interesting research topic would be developing a new interface for accelerating industrial robot programming on VR or mixed reality (MR) based platform. Traditional robotic programming methods are considered inefficient and outdated in current industrial and market demands, because they require professional knowledge for robot programming to complete a task. Developing a more efficient programming platform for industrial robots by integrating cutting edge technologies, i.e., MR and advanced programming by demonstration (PbD), could be a solution. Our previous work [110] has proved that VR/MR and our developed PbD can be combined seamlessly in the platform, thus enabling it to inherit the advantages of the both techniques. VR/MR serves as an interface of transferring human impedance adaptive skills to robots. The platform could provide its user an immersive experience and personalised assistance with a fast and high-efficiency method while training the industrial robot by demonstration.

Here has listed several objectives for future references: Development of online training an industrial robot to perform a set of motor primitives by grounding concrete action using PbD techniques; Increasing the combinations of actions that can be taught to the robot for the generation of more complex movements; Development of online learning techniques for the improvement of the learned manipulation skills; Grounding EMG signal from human masters to enable a robot to pose more human-like behaviours; Development of force/haptic feedback mechanism into VR/MR interface for enhanced demonstration experience of human user; Combing visual feedback of robot workspace into VR/MR environment to provide an immersive operation environment for human operators; Integration of human master skill transfer, PbD system and virtual vision into one single system; Investigation of how different behavioural and cognitive capabilities can enhance the motor skills learning through Human-robot physical Interaction (pHRI); Development of a set of prototypes that depict an interactive PbD process between a human user and a robot.

References

[1] P. Liang, C. Yang, Z. Li, and R. Li, "Writing skills transfer from human to robot using stiffness extracted from semg," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015 IEEE International Conference on*, pp. 19–24, 2015.

[2] R. Robotics, "Rethink robotics," May 2014. Available: http://sdk.rethinkrobotics. com/mediawiki-1.22.2/images/thumb/4/48/ Hand_sensors.png/425px-Hand_sensors.png. Accessed: 2015-07-01.

[3] "Baxter robot profile." `http://www.rethinkrobotics.com/products/baxter/`. Accessed: 2014-05-01.

[4] Z. Ju, C. Yang, and H. Ma, "Kinematics modeling and experiment verification for baxter robot," in *Proceedings of the 33rd Chinese Control Conference (CCC2014)*, (Nanjing), CCC, 2014.

[5] L. Cheng, Y. Lin, Z. G. Hou, M. Tan, J. Huang, and W. J. Zhang, "Integrated design of machine body and control algorithm for improving the robustness of a closed-chain five-bar machine," *IEEE/ASME Transactions on Mechatronics*, vol. 17, pp. 587–591, June 2012.

[6] Z. Ju, C. Yang, Z. Li, L. Cheng, and H. Ma, "Teleoperation of humanoid baxter robot using haptic feedback," in *Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on*, pp. 1–6, Sept 2014.

[7] M. Tawfik, D. Lowe, C. Salzmann, D. Gillet, E. Sancristobal, and M. Castro, "Critical factors in the architectural design of modern educational remote laboratories," in *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, pp. 1–8, IEEE, 2014.

[8] C. Yang, Z. Li, R. Cui, and B. Xu, "Neural network-based motion control of an underactuated wheeled inverted pendulum model," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, pp. 2004–2016, Nov 2014.

[9] D. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks, complex systems, vol. 2," 1988.

[10] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1481–1497, 1990.

[11] C. P. Bechlioulis and G. A. Rovithakis, "Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2090–2099, 2008.

[12] X. Wang, C. Yang, H. Ma, and L. Cheng, "Shared control for teleoperation enhanced by autonomous obstacle avoidance of robot manipulator," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 4575–4580, Sept 2015.

[13] C. Yang, S. Chang, P. Liang, Z. Li, and C. Y. Su, "Teleoperated robot writing using emg signals," in *Information and Automation, 2015 IEEE International Conference on*, pp. 2264–2269, Aug 2015.

[14] A. Ajoudani, N. Tsagarakis, and A. Bicchi, "Tele-impedance: Towards transferring human impedance regulation skills to robots," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 382–388, May 2012.

[15] A. Ajoudani, M. Gabiccini, N. Tsagarakis, and A. Bicchi, "Human-like impedance and minimum effort control for natural and efficient manipulation," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 4499–4505, May 2013.

[16] 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*, pp. 109–119, Springer, 2014.

[17] Y. Huang, K. Chen, K. Wang, Y. Chen, and X. Zhang, "Estimation of human arm motion based on semg in human-robot cooperative manipulation," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1771–1776, IEEE, 2018.

[18] C. Yang, J. Chen, Z. Li, W. He, and C. Y. Su, "Development of a physiological signals enhanced teleoperation strategy," in *Information and Automation, 2015 IEEE International Conference on*, pp. 13–19, Aug 2015.

[19] C. Li, H. Ma, C. Yang, and M. Fu, "Teleoperation of a virtual icub robot under framework of parallel system via hand gesture recognition," in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1469–1474, July 2014.

[20] H. Reddivari, C. Yang, Z. Ju, P. Liang, Z. Li, and B. Xu, "Teleoperation control of baxter robot using body motion tracking," in *Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on*, pp. 1–6, 2014.

[21] Y. Liu, Y. Zhang, B. Fu, and R. Yang, "Predictive control for robot arm teleoperation," in *Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE*, pp. 3693–3698, Nov 2013.

[22] I. Jo, Y. Park, and J. Bae, "A teleoperation system with an exoskeleton interface," in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1649–1654, July 2013.

[23] J. Chen, M. Glover, C. Li, and C. Yang, "Development of a user experience enhanced teleoperation approach," in *2016 International Confer-*

ence on Advanced Robotics and Mechatronics (ICARM), pp. 171–177, Aug 2016.

[24] J. C. Garc B. Patr L. Almeida, J. Pz, P. Menezes, J. Dias, and P. J. Sanz, "A natural interface for remote operation of underwater robots," *IEEE Computer Graphics and Applications*, vol. 37, pp. 34–43, Jan 2017.

[25] A. Gaschler, M. Springer, M. Rickert, and A. Knoll, "Intuitive robot tasks with augmented reality and virtual obstacles," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6026–6031, May 2014.

[26] S. Livatino, F. Banno, and G. Muscato, "3-d integration of robot vision and laser data with semiautomatic calibration in augmented reality stereoscopic visual interface," *IEEE Transactions on Industrial Informatics*, vol. 8, pp. 69–77, Feb 2012.

[27] A. Billard and R. Siegwart, "Robot learning from demonstration," 2004.

[28] C. Yang, P. Liang, Z. Li, A. Ajoudani, C. Y. Su, and A. Bicchi, "Teaching by demonstration on dual-arm robot using variable stiffness transferring," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1202–1208, Dec 2015.

[29] Y. Sung and K. Cho, "Collaborative programming by demonstration in a virtual environment," *IEEE Intelligent Systems*, vol. 27, pp. 14–17, March 2012.

[30] L. Hamon, "Virtual reality and programming by demonstration: Teaching a robot to grasp a dynamic object by the generalization of human demonstrations," *Presence*, vol. 20, pp. 241–253, June 2011.

[31] H. I. Krebs, N. Hogan, M. L. Aisen, and B. T. Volpe, "Robot-aided neurorehabilitation," *Rehabilitation Engineering, IEEE Transactions on*, vol. 6, no. 1, pp. 75–87, 1998.

[32] K. Inoue, T. Tanikawa, and T. Arai, "Micro hand with two rotational fingers and manipulation of small objects by teleoperation," in *Micro-NanoMechatronics and Human Science, 2008. MHS 2008. International Symposium on*, pp. 97–102, IEEE, 2008.

[33] W.-K. Yoon, T. Goshozono, H. Kawabe, M. Kinami, Y. Tsumaki, M. Uchiyama, M. Oda, and T. Doi, "Model-based space robot teleoperation of ets-vii manipulator," *Robotics and Automation, IEEE Transactions on*, vol. 20, pp. 602–612, June 2004.

[34] X. Yang, Q. Chen, D. Petriu, and E. Petriu, "Internet-based teleoperation of a robot manipulator for education," in *Haptic, Audio and Visual Environments and Their Applications, 2004. HAVE 2004. Proceedings. The 3rd IEEE International Workshop on*, pp. 7–11, Oct 2004.

[35] N. Wang, C. Yang, M. Lyu, and Z. Li, "An emg enhanced impedance and force control framework for telerobot operation in space," in *Aerospace Conference, 2014 IEEE*, pp. 1–10, March 2014.

[36] D. W. Franklin and D. M. Wolpert, "Computational mechanisms of sensorimotor control," *Neuron*, vol. 72, no. 3, pp. 425–442, 2011.

[37] D. W. Franklin, E. Burdet, K. P. Tee, R. Osu, C.-M. Chew, T. E. Milner, and M. Kawato, "Cns learns stable, accurate, and efficient movements using a simple algorithm," *The journal of neuroscience*, vol. 28, no. 44, pp. 11165–11173, 2008.

[38] A. Bierbaum, M. Rambow, T. Asfour, and R. Dillmann, "A potential field approach to dexterous tactile exploration of unknown objects," in

Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on, pp. 360–366, Dec 2008.

[39] A. Morales, T. Asfour, P. Azad, S. Knoop, and R. Dillmann, "Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 5663–5668, Oct 2006.

[40] E. Burdet, R. Osu, D. W. Franklin, T. E. Milner, and M. Kawato, "The central nervous system stabilizes unstable dynamics by learning optimal impedance," *Nature*, vol. 414, no. 6862, pp. 446–449, 2001.

[41] D. W. Franklin, R. Osu, E. Burdet, M. Kawato, T. E. Milner, D. W, R. Osu, E. Burdet, and M. Kawato, "Adaptation to stable and unstable dynamics achieved by combined impedance control and inverse dynamics model," *J. Neurophysiol*, pp. 3270–3282, 2003.

[42] L. P. Selen, P. J. Beek, and J. H. van Dieën, "Can co-activation reduce kinematic variability? a simulation study," *Biological cybernetics*, vol. 93, no. 5, pp. 373–381, 2005.

[43] A. Ajoudani, N. Tsagarakis, and A. Bicchi, "Tele-impedance: Teleoperation with impedance regulation using a body–machine interface," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1642–1656, 2012.

[44] W. Song, X. Guo, F. Jiang, S. Yang, G. Jiang, and Y. Shi, "Teleoperation humanoid robot control system based on kinect sensor," in *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on*, vol. 2, pp. 264–267, Aug 2012.

[45] C. Stanton, A. Bogdanovych, and E. Ratanasena, "Teleoperation of a humanoid robot using full-body motion capture, example movements,

and machine learning," in *Proc. Australasian Conference on Robotics and Automation*, 2012.

[46] R. Osu, D. W. Franklin, H. Kato, H. Gomi, K. Domen, T. Yoshioka, and M. Kawato, "Short- and long-term changes in joint co-contraction associated with motor learning as revealed from surface emg," *Journal of Neurophysiology*, vol. 88, no. 2, pp. 991–1004, 2002.

[47] G. Ray and S. Guha, "Relationship between the surface e.m.g. and muscular force," *Medical and Biological Engineering and Computing*, vol. 21, no. 5, pp. 579–586, 1983.

[48] J. Vogel, C. Castellini, and P. Van Der Smagt, "Emg-based teleoperation and manipulation with the dlr lwr-iii," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 672–678, Sept 2011.

[49] K.-J. You, K.-W. Rhee, and H.-C. Shin, "Finger motion decoding using emg signals corresponding various arm postures," *Experimental neurobiology*, vol. 19, no. 1, pp. 54–61, 2010.

[50] A. Ajoudani, N. Tsagarakis, and A. Bicchi, "Tele-impedance: Towards transferring human impedance regulation skills to robots," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 382–388, May 2012.

[51] R. Osu and H. Gomi, "Multijoint muscle regulation mechanisms examined by measured human arm stiffness and emg signals," *Journal of Neurophysiology*, vol. 81, no. 4, pp. 1458–1468, 1999.

[52] E. Burdet, R. Osu, D. W. Franklin, T. E. Milner, and M. Kawato, "The central nervous system stabilizes unstable dynamics by learning optimal impedance," *Nature*, vol. 414, no. 6862, pp. 446–449, 2001.

[53] G. C. Burdea, C. Burdea, and C. Burdea, *Force and touch feedback for virtual reality.* Wiley New York, 1996.

[54] R. J. Stone, "Haptic feedback: A brief history from telepresence to virtual reality," in *Haptic Human-Computer Interaction*, pp. 1–16, Springer, 2001.

[55] T. Sansanayuth, I. Nilkhamhang, and K. Tungpimolrat, "Teleoperation with inverse dynamics control for phantom omni haptic device," in *SICE Annual Conference (SICE), 2012 Proceedings of*, pp. 2121–2126, Aug 2012.

[56] A. J. Silva, O. A. D. Ramirez, V. P. Vega, and J. P. O. Oliver, "Phantom omni haptic device: Kinematic and manipulability," in *Electronics, Robotics and Automotive Mechanics Conference, 2009. CERMA'09.*, pp. 193–198, IEEE, 2009.

[57] J. Rastegar and D. Perel, "Generation of manipulator workspace boundary geometry using the monte carlo method and interactive computer graphics," *Journal of mechanical design*, vol. 112, no. 3, pp. 452–454, 1990.

[58] M. L. Soria, J. P. Martinez, and P. Laguna, "A multilead wavelet-based ecg delineator based on the rms signal," in *2006 Computers in Cardiology*, pp. 153–156, IEEE, 2006.

[59] L. S. Gettes and J. W. Mason, "Understanding proarrhythmic potential in therapeutic drug development: alternate strategies for measuring and tracking repolarization," *Journal of electrocardiology*, vol. 39, no. 4S, p. S161, 2006.

[60] D. Lloyd and T. Buchanan, "A model of load sharing between muscles and soft tissues at the human knee during static tasks," *Journal of biomechanical engineering*, vol. 118, no. 3, pp. 367–376, 1996.

[61] C. Yang, J. Chen, Z. Li, W. He, and C.-Y. Su, "Development of a physiological signals enhanced teleoperation strategy," in *2015 IEEE International Conference on Information and Automation*, pp. 13–19, IEEE, 2015.

[62] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics Automation Magazine*, vol. 13, pp. 82–90, Dec 2006.

[63] F. Chaumette and S. Hutchinson, *Visual Servoing and Visual Tracking*, pp. 563–583. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[64] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 651–670, Oct 1996.

[65] F. Chaumette and S. Hutchinson, "Visual servo control. ii. advanced approaches [tutorial]," *IEEE Robotics Automation Magazine*, vol. 14, pp. 109–118, March 2007.

[66] J. Stavnitzky and D. Capson, "Multiple camera model-based 3-d visual servo," *IEEE Transactions on Robotics and Automation*, vol. 16, pp. 732–739, Dec 2000.

[67] C. Dune, E. Marchand, and C. Leroux, "One click focus with eye-in-hand/eye-to-hand cooperation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 2471–2476, April 2007.

[68] K. P. Tee, S. S. Ge, R. Yan, and H. Li, "Adaptive control for robot manipulators under ellipsoidal task space constraints," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1167–1172, IEEE, 2012.

[69] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Transactions on Cybernetics*, vol. 46, pp. 620–629, March 2016.

[70] B. Xu, C. Yang, and Y. Pan, "Global neural dynamic surface tracking control of strict-feedback systems with application to hypersonic flight vehicle," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, pp. 2563–2575, Oct 2015.

[71] C. Wang and D. J. Hill, *Deterministic Learning Theory for Identification, Recognition, and Control.* Boca Raton: CRC Press, 2009.

[72] C. Wang and D. J. Hill, "Deterministic learning and rapid dynamical pattern recognition," *Neural Networks, IEEE Transactions on*, vol. 18, no. 3, pp. 617–630, 2007.

[73] F. Liu, W. Zeng, C. Yuan, Q. Wang, Y. Wang, and B. Lu, "Trajectory-based hand gesture recognition using kinect via deterministic learning," in *2018 37th Chinese Control Conference (CCC)*, pp. 9273–9278, IEEE, 2018.

[74] Z. Xue, C. Wang, and T. Liu, "Deterministic learning and robot manipulator control," in *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, pp. 1989–1994, IEEE, 2007.

[75] Y. Chengzhi, A. Marwan, D. Xiaonan, Z. Wei, S. Paolo, and D. Chang, "Cooperative deterministic learning control of multi-robot manipulators," in *2018 37th Chinese Control Conference (CCC)*, pp. 2824–2829, IEEE, 2018.

[76] C. Wang and T. Chen, "Rapid detection of small oscillation faults via deterministic learning," *Neural Networks, IEEE Transactions on*, vol. 22, no. 8, pp. 1284–1296, 2011.

[77] M. Ralph and M. A. Moussa, "Toward a natural language interface for transferring grasping skills to robots," *IEEE Transactions on Robotics*, vol. 24, pp. 468–475, April 2008.

[78] S. L. Dai, M. Wang, and C. Wang, "Neural learning control of marine surface vessels with guaranteed transient tracking performance," *IEEE Transactions on Industrial Electronics*, vol. 63, pp. 1717–1727, March 2016.

[79] A. Loria and E. Panteley, "Uniform exponential stability of linear time-varying systems: revisited," *Systems Control Letters*, vol. 47, no. 1, pp. 13 – 24, 2002.

[80] Z. Ju, C. Yang, and H. Ma, "Kinematics modeling and experimental verification of baxter robot," in *Control Conference (CCC), 2014 33rd Chinese*, pp. 8518–8523, July 2014.

[81] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 807–814 vol. 2, June 2005.

[82] C. Yang, S. Amarjyoti, X. Wang, Z. Li, H. Ma, and C. Y. Su, "Visual servoing control of baxter robot arms with obstacle avoidance using kinematic redundancy," in *Intelligent Robotics and Applications* (H. Liu, N. Kubota, X. Zhu, R. Dillmann, and D. Zhou, eds.), (Cham), pp. 568–580, Springer International Publishing, 2015.

[83] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control.* Springer Science & Business Media, 2010.

[84] C. Yang, J. Chen, Z. Ju, and A. S. Annamalai, "Visual servoing of humanoid dual-arm robot with neural learning enhanced skill transferring

control," *International Journal of Humanoid Robotics*, vol. 15, no. 02, p. 1750023, 2018.

[85] H. Reddivari, C. Yang, Z. Ju, P. Liang, Z. Li, and B. Xu, "Teleoperation control of baxter robot using body motion tracking," in *Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on*, pp. 1–6, Sept 2014.

[86] J. J. Craig, *Introduction to robotics: mechanics and control*, vol. 3. Pearson Prentice Hall Upper Saddle River, 2005.

[87] C. Yang, J. Chen, and F. Chen, "Neural learning enhanced teleoperation control of baxter robot using imu based motion capture," in *2016 22nd International Conference on Automation and Computing (ICAC)*, pp. 389–394, IEEE, 2016.

[88] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer handbook of robotics*, pp. 1371–1394, Springer, 2008.

[89] A. Sarmah and U. Gulhane, "Surgical robot teleoperated laparoscopic grasper with haptics feedback system," in *Emerging Trends in Robotics and Communication Technologies (INTERACT), 2010 International Conference on*, pp. 288–291, Dec 2010.

[90] S. Livatino, G. Muscato, and F. Privitera, "Stereo viewing and virtual reality technologies in mobile robot teleguide," *IEEE Transactions on Robotics*, vol. 25, pp. 1343–1355, Dec 2009.

[91] J. Kofman, X. Wu, T. J. Luu, and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *IEEE Transactions on Industrial Electronics*, vol. 52, pp. 1206–1219, Oct 2005.

[92] Z. Ju, C. Yang, Z. Li, L. Cheng, and H. Ma, "Teleoperation of humanoid baxter robot using haptic feedback," in *Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on*, pp. 1–6, Sept 2014.

[93] X. Wang, C. Yang*, and H. Ma, "Automatic obstacle avoidance using redundancy for shared controlled telerobot manipulator," in *The 5th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, (Shenyang, China), pp. 1338–1343, June 8-12 2015.

[94] X. Hou and R. Mahony, "Dynamic kinesthetic boundary for haptic teleoperation of vtol aerial robots in complex environments," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, pp. 694–705, May 2016.

[95] S. K. Cho, H. Z. Jin, J. M. Lee, and B. Yao, "Teleoperation of a mobile robot using a force-reflection joystick with sensing mechanism of rotating magnetic field," *IEEE/ASME Transactions on Mechatronics*, vol. 15, pp. 17–26, Feb 2010.

[96] S. Tachi, K. Komoriya, K. Sawada, T. Nishiyama, T. Itoko, M. Kobayashi, and K. Inoue, "Telexistence cockpit for humanoid robot control," *Advanced Robotics*, vol. 17, no. 3, pp. 199–217, 2003.

[97] V. Do, I. Longshore, D. Mally, and S. Raper, "Lakitu: Long-range aerial kinetic immersive telepresence unit," 2015. Available: http://robohub.org/wp-content/uploads/2013/04/BaxterResearch.jpg. Accessed: 2015-08-20.

[98] D. Pamungkas and K. Ward, "Immersive teleoperation of a robot arm using electro-tactile feedback," in *Automation, Robotics and Applications (ICARA), 2015 6th International Conference on*, pp. 300–305, Feb 2015.

[99] C. Ha, S. Park, J. Her, I. Jang, Y. Lee, G. R. Cho, H. I. Son, and D. Lee, "Whole-body multi-modal semi-autonomous teleoperation of mobile manipulator systems," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 164–170, May 2015.

[100] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker, "Human-inspired robotic grasp control with tactile sensing," *IEEE Transactions on Robotics*, vol. 27, pp. 1067–1079, Dec 2011.

[101] S. Lee, G. Sukhatme, G. J. Kim, and C. M. Park, "Haptic teleoperation of a mobile robot: A user study," *Presence*, vol. 14, pp. 345–365, June 2005.

[102] D. Y. Kim, M. S. Lee, S. H. Choi, K.-J. Koo, I. Hwang, and Y. J. Kim, "An immersive telepresence platform based on distributed architecture," in *2013 International Conference on ICT Convergence*, pp. 465–467, Oct 2013.

[103] C. A. James, T. P. Bednarz, K. Haustein, L. Alem, C. Caris, and A. Castleden, "Tele-operation of a mobile mining robot using a panoramic display: an exploration of operators sense of presence," in *2011 IEEE International Conference on Automation Science and Engineering*, pp. 279–284, Aug 2011.

[104] M. A. Conn and S. Sharma, "Immersive telerobotics using the oculus rift and the 5dt ultra data glove," in *2016 International Conference on Collaboration Technologies and Systems*, pp. 387–391, Oct 2016.

[105] J. Regenbrecht, A. Tavakkoli, and D. Loffredo, "A robust and intuitive 3d interface for teleoperation of autonomous robotic agents through immersive virtual reality environments," in *2017 IEEE Symposium on 3D User Interfaces*, pp. 199–200, March 2017.

[106] A. Dünser, M. Lochner, U. Engelke, and D. R. Fernández, "Visual and manual control for human-robot teleoperation," *IEEE Computer Graphics and Applications*, vol. 35, pp. 22–32, May 2015.

[107] J. Chen, M. Glover, C. Li, and C. Yang, "Development of a user experience enhanced teleoperation approach," in *2016 International Conference on Advanced Robotics and Mechatronics*, pp. 171–177, Aug 2016.

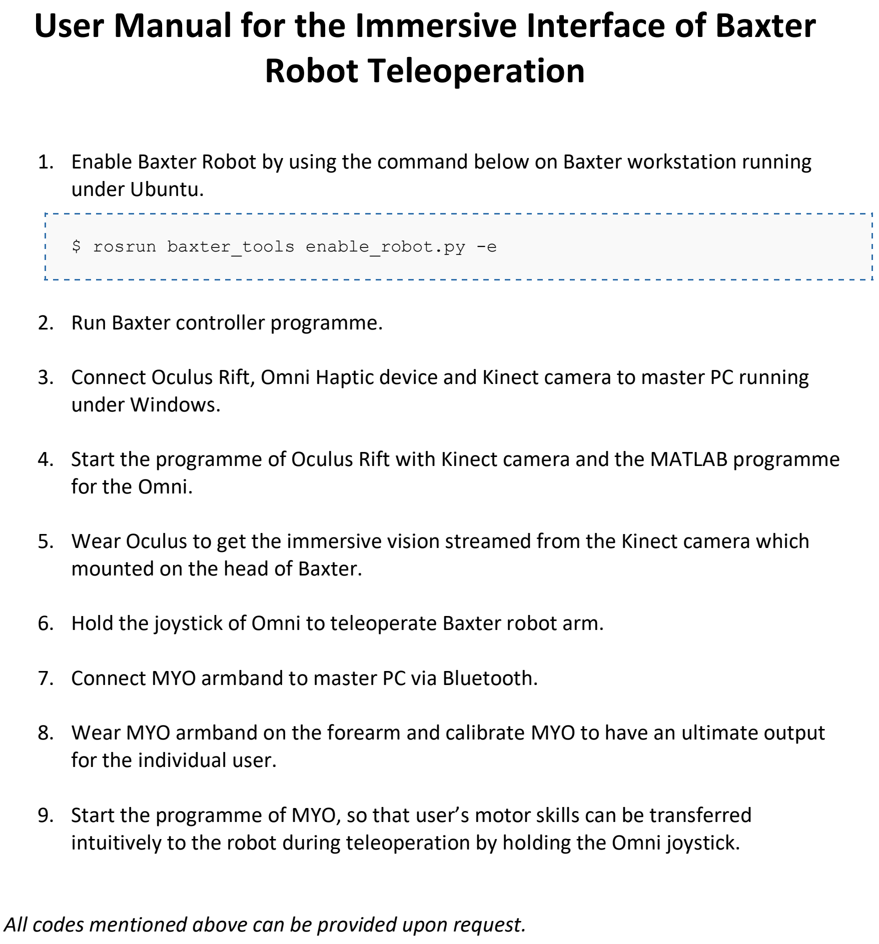
[108] Z. Xiao, Z. Li, and M. Chen, "Position/force estimation using hill muscle model incorporating adaboost with svm-based component classifiers," in *Intelligent Control and Automation (WCICA), 2014 11th World Congress on*, pp. 1923–1928, June 2014.

[109] J. Chen, M. Glover, C. Yang, C. Li, Z. Li, and A. Cangelosi, "Development of an immersive interface for robot teleoperation," in *Annual Conference Towards Autonomous Robotic Systems*, pp. 1–15, Springer, 2017.

[110] M. Cousins, C. Yang, J. Chen, W. He, and Z. Ju, "Development of a mixed reality based interface for human robot interaciotn," in *2017 International Conference on Machine Learning and Cybernetics (ICMLC)*, vol. 1, pp. 27–34, IEEE, 2017.

Appendices

A Teleoperation System User Manual



User Manual for the Immersive Interface of Baxter Robot Teleoperation

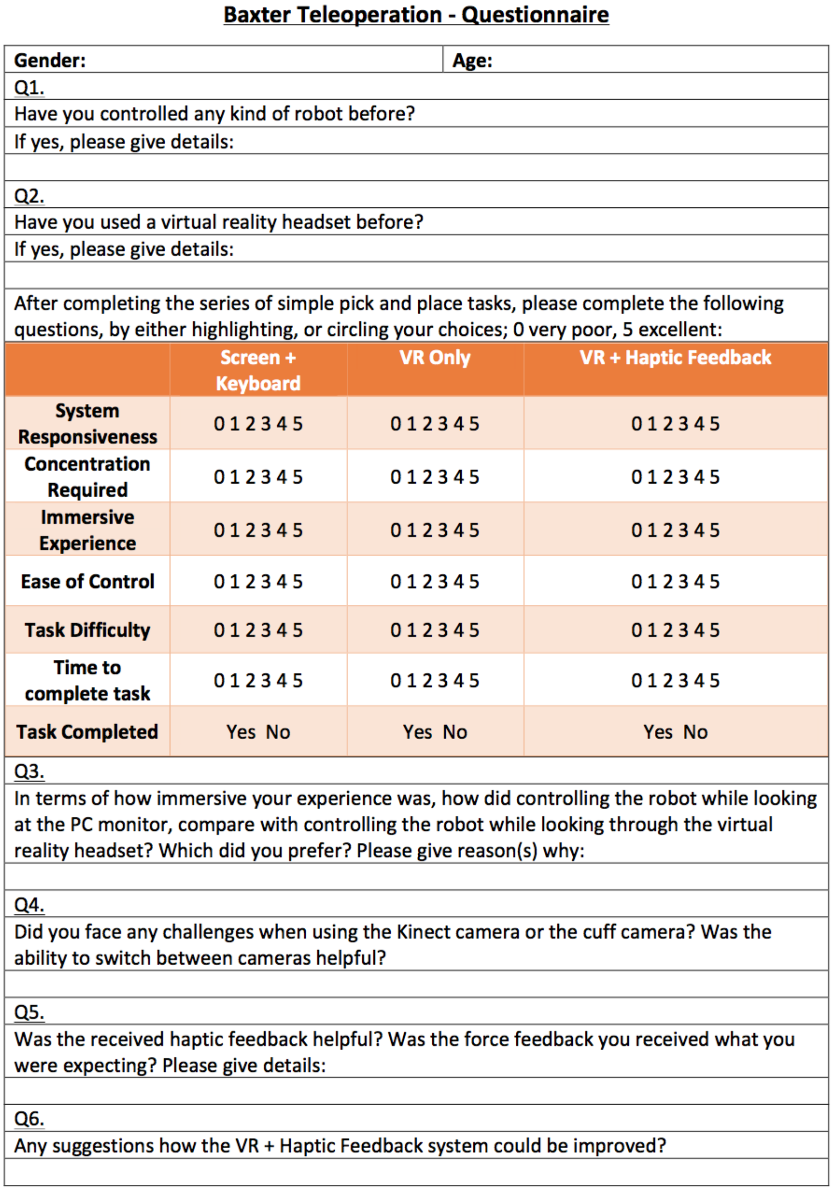
1. Enable Baxter Robot by using the command below on Baxter workstation running under Ubuntu.

```
$ rosrun baxter_tools enable_robot.py -e
```

2. Run Baxter controller programme.
3. Connect Oculus Rift, Omni Haptic device and Kinect camera to master PC running under Windows.
4. Start the programme of Oculus Rift with Kinect camera and the MATLAB programme for the Omni.
5. Wear Oculus to get the immersive vision streamed from the Kinect camera which mounted on the head of Baxter.
6. Hold the joystick of Omni to teleoperate Baxter robot arm.
7. Connect MYO armband to master PC via Bluetooth.
8. Wear MYO armband on the forearm and calibrate MYO to have an ultimate output for the individual user.
9. Start the programme of MYO, so that user’s motor skills can be transferred intuitively to the robot during teleoperation by holding the Omni joystick.

All codes mentioned above can be provided upon request.

B Participant Questionnaire



Baxter Teleoperation - Questionnaire

Gender: **Age:**

Q1.

Have you controlled any kind of robot before?

If yes, please give details:

Q2.

Have you used a virtual reality headset before?

If yes, please give details:

After completing the series of simple pick and place tasks, please complete the following questions, by either highlighting, or circling your choices; 0 very poor, 5 excellent:

	Screen + Keyboard	VR Only	VR + Haptic Feedback
System Responsiveness	0 1 2 3 4 5	0 1 2 3 4 5	0 1 2 3 4 5
Concentration Required	0 1 2 3 4 5	0 1 2 3 4 5	0 1 2 3 4 5
Immersive Experience	0 1 2 3 4 5	0 1 2 3 4 5	0 1 2 3 4 5
Ease of Control	0 1 2 3 4 5	0 1 2 3 4 5	0 1 2 3 4 5
Task Difficulty	0 1 2 3 4 5	0 1 2 3 4 5	0 1 2 3 4 5
Time to complete task	0 1 2 3 4 5	0 1 2 3 4 5	0 1 2 3 4 5
Task Completed	Yes No	Yes No	Yes No

Q3.

In terms of how immersive your experience was, how did controlling the robot while looking at the PC monitor, compare with controlling the robot while looking through the virtual reality headset? Which did you prefer? Please give reason(s) why:

Q4.

Did you face any challenges when using the Kinect camera or the cuff camera? Was the ability to switch between cameras helpful?

Q5.

Was the received haptic feedback helpful? Was the force feedback you received what you were expecting? Please give details:

Q6.

Any suggestions how the VR + Haptic Feedback system could be improved?