Swansea University

School of Engineering

Zienkiewicz Centre for Computational Engineering

Learning Algorithm Design for Human-Robot Skill Transfer

Author:
Chunxu Li

Supervisor:
Prof. Chenguang Yang

Submitted to Swansea University in fulfilment of the requirements for the Degree of Doctor of Philosophy

May 30, 2019

Abstract

In this research, we develop an intelligent learning scheme for performing human-robot skills transfer. Techniques adopted in the scheme include the Dynamic Movement Primitive (DMP) method with Dynamic Time Warping (DTW), Gaussian Mixture Model (GMM) with Gaussian Mixture Regression (GMR) and the Radical Basis Function Neural Networks (RBFNNs). A series of experiments are conducted on a Baxter robot, a NAO robot and a KUKA iiwa robot to verify the effectiveness of the proposed design.

During the design of the intelligent learning scheme, an online tracking system is developed to control the arm and head movement of the NAO robot using a Kinect sensor. The NAO robot is a humanoid robot with 5 degrees of freedom (DOF) for each arm. The joint motions of the operator's head and arm are captured by a Kinect V2 sensor, and this information is then transferred into the workspace via the forward and inverse kinematics.

In addition, to improve the tracking performance, a Kalman filter is further employed to fuse motion signals from the operator sensed by the Kinect V2 sensor and a pair of MYO armbands, so as to teleoperate the Baxter robot. In this regard, a new strategy is developed using the vector approach to accomplish a specific motion capture task. For instance, the arm motion of the operator is captured by a Kinect sensor and programmed through a processing software. Two MYO armbands with embedded inertial measurement units are worn by the operator to aid the robots in detecting and replicating the operator's arm movements. For this purpose, the armbands help to recognize and calculate the precise velocity of motion of the operator's arm. Additionally, a neural network based adaptive controller is designed and implemented on the Baxter robot to illustrate the validation for the teleoperation of the Baxter robot.

Subsequently, an enhanced teaching interface has been developed for the robot using DMP and GMR. Motion signals are collected from a human demonstrator via the Kinect v2 sensor, and the data is sent to a remote PC for teleoperating the Baxter robot. At this stage, the DMP is utilized to model and generalize the movements. In order to learn from multiple demonstrations, DTW is used for the preprocessing of the data recorded on the robot platform, and GMM is employed for the evaluation of DMP to generate multiple patterns after the completion of the teaching process. Next, we apply the GMR algorithm to generate a synthesized trajectory to minimize position errors in the three dimensional (3D) space. This approach has been tested by performing tasks on a KUKA iiwa and a Baxter robot, respectively.

Finally, an optimized DMP is added to the teaching interface. A character recombination technology based on DMP segmentation that uses verbal command has also been developed and incorporated in a Baxter robot platform. To imitate the recorded motion signals produced by the demonstrator, the operator trains the Baxter robot by physically guiding it to complete the given task. This is repeated five times, and the generated training data set is utilized via the playback system. Subsequently, the DTW is employed to pre-process the experimental data. For modelling and overall movement control, DMP is chosen. The GMM is used to generate multiple patterns after implementing the teaching process. Next, we employ the GMR algorithm to reduce position errors in the 3D space after a synthesized trajectory has been generated. The Baxter robot, remotely controlled by the user datagram protocol (UDP) in a PC, records and reproduces every trajectory. Additionally, Dragon Natural Speaking software is adopted to transcribe the voice data. This proposed approach has been verified by enabling the Baxter robot to perform a writing task of drawing different Chinese characters after the robot has been taught to write only one character.

List of Abbreviations/Nomenclatures

Table 1: Abbreviations

Abbreviation	Description
DOF	Degrees of freedom
DTW	Dynamic Time Warping
GMM	Gaussian Mixture Model
GMR	Gaussian Mixture Regression
DMP	Dynamic Movement Primitive
NN	Neural Networks
UDP	User Datagram Protocol
DH	Denavit & Hartenberg's method
KF	Kalman Filter method
TbD	Teaching by Demonstration
ROS	Robot Operation System
UAV	Unmanned Aerial Vehicle
RBF	Radial Basis Function
EM	Expectation - Maximization algorithm
BIC	Bayesian Information Criterion
PID	Proportional Integral Derivative
3D Space	Three Dimensional Space

Table 2: Nomenclatures

Symbol	Description
n	Number of joints
τ	Torque
d, c	Spring and Damping coefficients
x, v	Cartesian position and velocity of the end-effector
s	Canonical system states
k	Number of GMMs
$p_k, \mu_k, \sum_k$	Weight, mean and variance of GMM
z	Input vector
$S(z)$	Gaussian regressor vector
$q, \dot{q}, \ddot{q} \in R^{n\times 1}$	Vector of joint position, angular velocity and acceleration respectively
a, d, α, θ	Variables denoting the Denavit-Hartenberg parameters
${}^{i}T_{j} \in R^{4\times 4}$	Homogenous transform from link i to j
$I(q) \in R^{5\times 5}$	Inertia matrix
$C(q, \dot{q}) \in R^{5\times 5}$	Coriolis matrix
$G(q) \in R^{5\times 1}$	Gravity terms
$U(q) \in R^{5\times 1}$	Unmodeled elements
K	Control gain
$s(t), r(t)$	System noise and measured noise variance intensities
$\omega(t), v(t)$	Noise vectors of system and measurement
$x(t), u(t)$	System state variables
$y(t)$	Measurement vectors

Acknowledgements

First, I would like to express my sincere thanks to my supervisor Prof. Chenguang Yang. During the course of the past three years, there were moments of joy and bitterness. It is his never-ending support and encouragements that provided me with the confidence to pursue directions to my heart's desire. Without his guidance, I could never have hoped to achieve what I have today. I would also like to convey my thanks to all the lovely colleagues at the ZCCE where I worked, lived and laughed in the past three years, including the veterans, the past members and the new comers.

It is a great pleasure and honor to work among such a fun and ingenious bunch of people. It is a place filled with fresh ideas. Finally, thanks to my wife Nana Ji, for her love and support, especially when the writing of this thesis coincided with such an important moment of our lives. Without her help and understanding, I may never have glimpsed the light at the end of this tunnel. Thanks also goes to my mother, who travelled all the way from the other side of the globe to provide help during the last days of my writing when I needed it the most. Also, it is the long-term love, support and understanding of my family during all these years behind the scenes that have given me the chance to devote myself to my passion. Thanks to you all.

Statement of Authorship

This thesis is submitted to the Department of Engineering, Swansea University, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

List of Figures

List of Tables

Contents

1 Chapter One: Introduction

1.1 Research Motivations

Rapid development in robot technologies have necessitated the development of the robot's locomotive capacities. In the light of the myriad of uses, robots can serve in modernized hospitals, schools, business and other fields, the research regarding human-robot skill transfer has flourished during the last decades. Currently, robots are already being utilized with greater efficiency in the industrial field and it has been made possible to transfer human motor skills to robots after the motion signals of the operator have been captured by a computer.

Generally, the objective of human-robot skill transfer is to enable robots to manipulate their movements with better dexterity and versatility as human beings [11]. To this end, two approaches have been adopted. One is by modelling the human motor control which can be adapted and implemented on a robot. The other is to facilitate robot learning through demonstration in which the human operator plays the role of a tutor, and the robot, a tutee [12]. Development in this field started with the fixed trajectory based skills transfer from humans to robots. Thereafter, a hybrid mode was developed combining both position and force, which was further upgraded into an interactive mode by involving online feedback [9]. A recent progress is the enhanced skill transfer via the use of biosignals and physiological signals of the human body. These signals can provide us with a richer and deeper account of the mechanism for human muscle activities compared with physical signals [13].

Nowadays, with the advance of technology for human-robot skill transfer, demand for human operators has increased [14]. Meanwhile, robot's training processes have also significantly reduced. In [15], an approach was developed for transferring skills between tutor and tutee by capturing the movements of the tutor on a sensory-based computational model. This information was then used to produce online master commands for developing the learning process of the less-skilled tutees. In another research [16], a research group employed a neural network to record human skills on a computer, and then tried different ways to minimize the time of transferring skills from a human tutor to a tutee.

In recent years, motion capture technology has been developed and utilized in a wide range of areas, including HRI (human-robot interaction), computer animation and 3D film production [17]. It serves to accurately record the 3D motion trajectory of each part of a moving object [18]. Based on this information, accurate modelling of moving objects and concurrent semantic analyses of the movements can be achieved which is of great help in technological fields, such as animation [19]. However, motion capture is an extremely time-consuming process, due to the fact that the captured movement data require a huge amount of both the pre-processing and manual segmentation of the action sequences and also requires the accurate identification of semantics for each segment [20]. To overcome

these limitations, DTW has been proposed and employed to speed up the initial processes, for instance, in speech recognition, it assists in measuring the similarity between the two-time series to identify whether the two words represent the same term [21].

Except for the application in a structured environment, such as fixed-point operation in an industrial environment, research on robots application in the unstructured one has also gained the increasing attention, such as deep sea resources exploration [22], disaster site search and rescue operations [23], operation in radiation environment [24] [25]. The research regarding the autonomous work of robots in the dangerous natural environments has helped in reducing the risks of human lives for doing dangerous tasks in an unreachable environment. To start with, the robots have an obligation to autonomously work in an unstructured environment. However, traditional research methods are insufficient in meeting the demands, hence, inspiration has been drawn from the constantly evolving creatures that thrive in complex environments. By studying their shape, structure, cognitive abilities, motion mechanisms and behavioural adaptations, various bio-inspired ideas have been proposed for the development of robots. According to the speech of experts delivered at the 2004 edition of the IEEE International Conference on Robotics and Bionics, "*Bionics that mimic the function and structure of living organisms with biological characteristics will gradually replace industrial operators and become the focus of research in the field of robotics*" [26].

1.2 Research Innovations

With the advancement in artificial intelligence, research and application in robotics have also seen a great development [14]. At the same time, the research on human-robot skill transfer has attracted widespread public attention in the past decade. Human-robot skill transfer methodologies have been widely used in the industrial field [1]. As this technology helps to directly transfer information on the human motion to the robot, it helps to effectively replace humans with robots in high-risk operations and also benefits from work efficiency [27]. At present, regarding technological innovations and the frequent replacement of humans by robots in the industrial facilities, human operators are required to constantly learn new mechanical skills in order to operate the equipment efficiently. Therefore, the development of scalable intelligent human-robot interaction technology is innovating the traditional production line that involved repetitive movements, and is thus of great practical significance to industrial development.

This research focuses on the design of artificial intelligence algorithms which enable robots to mimic human actions, as well as to facilitate them in adapting and adjusting in complex environments, i.e. replacing or cooperating with human operators in performing various tasks. In the traditional manufacturing industry, the production line was lengthy and repetitive with no chance of skill expansion or skill adjustments [28]. In the latest research & development tasks, it is often necessary to re-analyse the point-to-point trajec-

tory so as to allow the machine equipment to re-read the work cycle [29]. The challenge in this research thinks this process is usually incredibly time-consuming and financially costing.

Inspired by the above-stated demands and challenges, the following topics are investigated in my research work:

a) Through the designing of the robot learning methodologies by combining the DMP, GMM, GMR and DTW algorithms, the robot is able to generalize the motion trajectories spatially/temporally, which greatly reduces the training/teaching time. This will, in turn, improve working efficiency. For example, when the manufacturing line/type has been changed, there is no need to re-program and re-teach the robots for the new production line [30];

b) Designing different teaching methods for different work conditions, such as the production of hazardous/toxic items, the real-time remote teleoperation based human-robot skill transfer can be employed. For productions that required high-precision operations, such as binding, and cutting, the physical teaching by demonstration method can be applied [31];

c) The voice interaction has been combined with human-robot skill transfer to achieve verbal commands control for robots while carrying out specific tasks [32];

d) Applying the KF to reduce the noise of the multiple sensors and NN to overcome the dynamic uncertainties of the robot learning system during the teleoperation process [33].

1.3 Publications

The research during my PhD study has led to a number of publications as detailed below:

1. Chunxu Li, Chenguang Yang, & Cinzia Giannetti. Segmentation and Generalization for Writing Skills Transfer from Humans to Robots. Cognitive Computation and Systems, 1(1):20-25, 2019.

2. Chunxu Li, Chenguang Yang, Zhaojie Ju, & Andy SK Annamalai. An enhanced teaching interface for a robot using DMP and GMR. International journal of intelligent robotics and applications, 2(1):110-121, 2018.

3. Chunxu Li, Chenguang Yang, Jian Wan, Andy SK Annamalai, & Angelo Cangelosi. Teleoperation control of baxter robot using kalman filter-based sensor fusion. Systems Science & Control Engineering, 5(1):156-167, 2017.

4. Chunxu Li, Chenguang Yang, Andy SK Annamalai, Qingsong Xu, & Shaoxiang Li. Development of Writing Task Recombination Technology Based on DMP Segmentation via Verbal Command for Baxter Robot. Systems Science & Control Engineering, 6(1):350-359, 2018.

5. Chunxu Li, Chenguang Yang, Jian Wan, Andy SK Annamalai, & Angelo Cangelosi. Neural learning and Kalman filtering enhanced teaching by demonstration for a Baxter

robot. In Automation and Computing (ICAC), 2017 23rd International Conference on (pp. 1-6). IEEE, 2017.

6. Chunxu Li, Chenguang Yang, Peidong Liang, Angelo Cangelosi, & Jian Wan. Development of Kinect based teleoperation of NAO robot. In Advanced Robotics and Mechatronics (ICARM), International Conference on (pp. 133-138). IEEE, 2016.

7. Junshen Chen, Mark Glover, Chunxu Li, & Chenguang Yang. Development of a user experience enhanced teleoperation approach. In Advanced Robotics and Mechatronics (ICARM), International Conference on (pp. 171-177). IEEE, 2016.

8. Alex Smith, Chenguang Yang, Chunxu Li, Hongbin Ma, & Lijun Zhao. Development of a dynamics model for the Baxter robot. In IEEE International Conference on Mechatronics and Automation (ICMA), International Conference on (pp. 1244-1249). IEEE, 2016.

9. Rijin Raju, Chenguang Yang, Chunxu Li, & Angelo Cangelosi. A video game design based on Emotiv Neuroheadset. In Advanced Robotics and Mechatronics (ICARM), International Conference on (pp. 14-19). IEEE, 2016.

10. Junshen Chen, Mark Glover, Chenguang Yang, Chunxu Li, Zhijun Li, & Angelo Cangelosi. Development of an immersive interface for robot teleoperation. In Conference Towards Autonomous Robotic Systems (pp. 1-15). Springer, Cham, 2017.

1.4 Organization of the Thesis

This thesis is structured in eight chapters. In the introduction (first chapter), the motivation and innovation of the research have been presented, discussing the differences between the traditional human-robot skill transfer methods with ours. Chapter 2 presents a detailed description, review of the academic products of researchers and the general strategies used in human-robot skill transfer technologies worldwide as well as analysing their novelties and challenges. In addition, we especially arrange a chapter (Chapter 3) to introduce the devices and basic algorithms. In Chapter 4, we develop an online tracking system to control the arm and head of the NAO robot using Kinect sensor for the teleoperation (representing the 6^{th} paper in the publication list above), however, experimental results show large errors using Kinect alone to track the human motion. Therefore, in Chapter 5 we add the KF technique to fuse the signals from multiple sensors in order to reduce the noise and the NN learning based control system is also developed to compensate the dynamic uncertainties, where the fused date are collected from Kinect sensor and a pair of MYO armbands (refer to my 5^{th} published paper from the list). In Chapter 6, an enhanced teaching interface for a robot using DMP and GMR has been designed and this proposed approach is tested on a KUKA iiwa and a Baxter robot by performing two tasks, which are passing through the obstacle of different heights and curve drawings with generalization, respectively, which refers to the 2^{nd} publication. Chapter 7 focuses on the development of the verbal commands based writing task recombination technology based on the proposed segmentation

mechanism. The technology is validated through performing a Chinese character writing task with the Baxter robot. During the task, different Chinese characters are written by teaching only one character representing the 4^{th} paper in the publication list above. Chapter 8 summarizes the contribution of our work, and presents some suggestions for future work.

2 Chapter Two: Review of the Related Work

2.1 Research Background

In recent years, there has been a rapid advancement in robotics, with its applications growing extensively in numerous fields: from entertainment to the medical field and many others. Development in robotics has greatly facilitated people's lives and is one of the most rapidly emerging industries in this day and age. Most countries, especially developed countries, are paying much attention to the development of robots, such as the National Robot Program of the United States, the Seventh Framework Program of the European Union and the National Natural Science Foundation of China's 863 Program [34]. Along with its unprecedented growth, research on robotics also faces a number of challenges.

From the beginning of the Industrial 4.0, the role of robots has become increasingly significant with the advance of science and technology. However, this role remains largely limited to traditional industrial applications such as pre-set and repeated tasks. At present, it remains a challenge to create robots that are capable of performing an assortment of tasks in unknown, complex and dynamic environments. Besides, service-oriented, medical as well as industrial robots are required to work in complex external environments. Hence, control of robots in unknown and dynamic environments has been widely studied and a higher demand for advanced intelligence in robots has been recognized. It is noteworthy that in the working environment, robots are inevitably affected by unpredictable and uncertain external disturbances. In this regard, traditional control methods are not sufficient for qualifying them in terms of intellectuality. Recent research has been actively focused on the interaction and control between robots and external forces. Interactive control technology cannot only assure compliance of robot behaviour, but also provide a relatively safe operating environment for operators, which are impossible to achieve through traditional methods of robot control. The interaction between robots and complex environments can lead to potential instability, inaccuracy, mechanical nonlinearity and saturation, thus may have a negative impact on the overall working system. At present, there are two main methods of interactive control between robot and environment external forces: hybrid force/position control and impedance/admittance control [35]. Hybrid force/position control has the potential of achieving good control performance and anti-disturbance ability. However, in case of a strong interaction between the robot and the environment, the hybrid force/position control method often leads to system instability and security problems [36]. As compared to hybrid force/position control, impedance/admittance control aims to establish the correct balance in the relationship between the robot and interactive force received from the external environment. Through impedance/admittance control method, robot's behaviour can be adapted to the environment and the overall stability of the control system can be ensured [37]. Therefore, further study and development of robot interactive control technology for dealing with various uncertainties in a real-world situation is neces-

sary so as to ensure that robots can work in a dynamic and unknown environment. Hence, the development of robot interactive control technology is of great practical significance.

In most nonlinear physical and mechanical systems, there exist wide-scale model dynamic uncertainties. These uncertainties are usually due to changes in the external environment, loss of mechanical devices, and model errors, etc. Currently, there is no general method for the control of nonlinear systems, due to the limitations in existing detection technology. In this regard, many experts and scholars have conducted the extensive and in-depth research to explore different control methods. So far, many control methods have already been applied in practical systems. However, different control methods have their own unique advantages and disadvantages; hence, research on relevant control theories and methods for nonlinear systems is still an open problem. At present, a feasible common approach is employed to combine control methods of various nonlinear systems for utilizing their advantages and compensating for their shortcomings, which achieves the desired control performance.

In addition, due to physical constraints, nonlinearities such as saturation, dead zone and hysteresis are prevalent in physical actuators which greatly limit the actual control effect and may potentially lead to instability of the system. For instance, saturation is one of the inevitable problems in most actuators. Therefore, it is of great importance to consider and solve the saturation nonlinearities in control design. In summary, this research focuses on transferring human motion skills for designing the smart control of industrial robots. For this purpose, the research studies interactive control between robots and external forces, and realizes the adaptability of robots towards external forces with the aim of guaranteeing control accuracy. Thus, this research not only has abundant value as a formative research in robotics, but also has a huge number of prospective applications in a variety of fields.

2.2 Development of Industrial Robots

As the highest integrated application of mechatronics technology, industrial robots are a testament to the great development automation that has achieved in the industrial field. In order to introduce innovative advancement in the industry, the design and manufacture of industrial robots have become increasingly important. In robotics, a robust robot control system is the core of the entire robot system, and plays an important role in improving the dynamic performance of robots, helping in reducing costs and improving work efficiency [27]. A new trend is the development of lightweight robots, such as the ABB industrial robots which have undergone a marked reduction in the load-to-weight ratio by a factor of three since the 1990s [1]. Lightweight design reduces incurred costs and overall energy consumption, while also reducing the mechanical stiffness of the system [38]. However, such a robot design results in complex vibration modes, which pose a challenge for robot control. Therefore, it is imperative to establish a dynamic robot model that efficiently incorporates flexible characteristics in robot design to promote the optimization

and effective control. Generally, industrial robots are highly versatile. Though some robot applications require a high control performance, most industrial robots normally still have some versatility in controlling performance requirements.

(d) : GSK arc welding robot (e) : COMAU arc welding robot (f) : SIASUN spot welding robot

Figure 2.1: The mainstream industrial robots worldwide [1]

Followings are some of the main requirements for control performance of industrial robots [39]:
(1) assuring accuracy in trajectory tracking during continuous motions (such as laser welding, laser cutting or water flow cutting);
(2) maintaining velocity and accuracy during continuous motions (such as spraying and gluing);
(3) managing high velocity and acceleration during manipulations (for instance material handling);
(4) sustaining small overshoot and short settling time during tasks (such as spot welding and palletizing) [29].

To improve the performance of robots, mainstream robot manufacturers (see Fig.2.1) have invested significantly in hiring qualified personnel and in control system research. As a result, numerous control system structures and algorithms have been proposed which effectively meet the practical engineering requirements. One of these is the method proposed

by Fanuc Robotics to reduce vibrations in robots [40]. The method designs a position estimation observer, which is later combined with the internal model control (IMC) structure. In this way, the controller design transforms the original system model into a single output and dual inertia one. However, this method has its limitations – it is complicated and lacks self-adaptivity in terms of generalization. Their invention, the Fanuc controller RJ3iC, features the enhanced vibration suppression, enabling accurate control. Another investigation, known as ABB Robotics' motion control system, uses a model-based control mechanism with continuous and path-tracking accuracy called the TrueMove mode [41]. In this mode, the velocity and accuracy of the manipulation can be closely monitored, however, it does not minimize vibration to a great extent. Additionally, Motoman Robotics designs a high-performance, accurate trajectory-tracking and vibration suppression control method based on an advanced robot motion control concept. Furthermore, the industrial robot control system developed by B&R and Beckhoff adopts a model-based control approach and combines torque feed-forward control method to compensate for the rapid change in nonlinear inertia and joint flexural deformations. Thus, it guarantees stability in controlling the robot as it moves at a high velocity [42].

2.3 Review of Human-Robot Interaction

According to [43], the human-robot interaction (HRI) is a sub-area of the human-computer interaction (HCI), which studies the interaction between humans and robots, and focuses on developing more intelligent and anthropomorphic robots. HRI is widely used in research and implementation of robot systems in hazardous zones where human involvement needs to be minimized and remote operation of robots is required. These inventions can also apply to care for the elderly and the disabled as well as to entertainment purposes [44]. There are various methods of human-robot interaction, amongst which, two are considered the main methods of — physical interaction and teleoperation interaction.

The physical human-robot interaction (pHRI) studies the design, control, and planning involved in the close physical interaction between humans and robots in a shared workspace. Previous research in pHRI leads to the development of a safe and responsive control method to coordinate and control the physical reactions that occur when a robot performs a task. Based on the research of Hogan et. al., impedance control is one of the most widely used methods for commanding a robot tracking a given orbit when there is a human partner in the workspace [45]. In this control method, the novelty is that the robot can act like a spring: it allows to be propelled by people and bounces back to its original position when not in use; the disadvantages are the control precision and non-adaptive speed, which limits its applications.

For specialised tasks, such as medical operations, the safest method for human-robot interaction is teleoperation [46]. In this method, the user interfaces with the robot through a haptic signal for coordinated feedback. The research [47] carries out experiments to

characterize this problem, and derives several methods to provide haptic feedback in order to improve a surgeon's performance. However, they do not take the enough required control room for the surgeon into consideration. As we all know, the surgeon's activities have high requirements for cleaning, and thus the remote control is a better choice. In [48], a research team introduces a visual impedance scheme for vision-based control of the robot so as to realize task-level dynamical control. In the control scheme, the image features are applied to the impedance equation so that integration of a visual servo and a conventional servo system could be accomplished. However, this research is also limited in the aspects of the self-adaptive decision.

In the research and application of robot technology, western countries gained a head-start before the eastern ones. Robot teaching and learning devices play an important part of the industrial robot control system and have thus been heavily investigated throughout the development of robot technology. To pursue development in robot application, numerous robot teaching devices have been developed by a number of industrial robot manufacturers and teams through independent scientific research. So far, industrial robot manufacturers and various scientific research institutions have developed a series of standards that can help promote the wide-scale application of robot technology in industrial production. Teaching devices design varies with different underlying concepts. Nevertheless, interface designs can be categorized into two categories: touch screen operation and keyboard operation [49].

(a) Human-robot interaction interface represented by keyboard operation

A FlexPendant consists of a display screen and the physical control buttons. The display screen is adopted to display information about robot's motion, and some may equip with touch screen functions for simple utilities. The robot motion function informs the work status of the robot. Physical control buttons provided under the display screen allow functional control of robot's motion. It benefits from efficiently maneuvering a robot's movements in a fast-paced industrial work environment wherein a skilled operator can quickly control an industrial robot work's file with the press of a key or a combination of keys [50].

An example of robot FlexPendant utilizing keyboard operation on the Kawasaki robot is shown in Fig.2.2. Buttons for controlling physical motions of the robots are provided alongside an operation manual. The information about task parameters and robot's state is displayed on the screen above the FlexPendant. The FlexPendant operates faster at processing information than the touch-screen, as it can collect messages faster through the use of physical buttons. However, too many physical buttons increase robot FlexPendant size thereby causing inconvenience for users [51].

(b) Human-robot interaction interface represented by touch screen operation

In this type of FlexPendant, the operator does not have a great variety in physical buttons for operation, in fact, sometimes there are not any physical buttons at all. Hence, FlexPendants are relatively small, and are thus convenient for personnel operation and simplified

Figure 2.2: The FlexPendant of the Kawasaki robot [2]

for operator tasks. However, as functions are mainly implemented by using a software program, control performance is somehow affected.

An example of touch-screen operation is the FlexPendant designed by KUKA Robotics as shown in Fig.2.3. In the corresponding human-robot interaction interface, various functions for controlling robot motion are provided, including monitoring various information states, creating robot motion work files, and reproducing the robot teaching function. In addition, the KUKA robot's control system has an automatic operation mode which ensures that the robot continues to operate even without an external higher-level controller [52]. In this thesis, the HRI interface is chosen as the latter in a KUKA iiwa robot platform because of its convenience and simple, easy-to-understand control system of the FlexPendant.

Figure 2.3: The FlexPendant of the KUKA robot

2.4 Review of Machine Learning in Robotics

With the development of science and technology, the demand for higher robotic intelligence and the minimization of dependent on human labour is increasing. Additionally, with the advancement in artificial intelligence, it has become possible to improve robots' intelligence to a level that it can accomplish some tasks independently in a specific scenario. A growing body of research is focused on exploring the interrelationship between the robots and the artificial intelligence [53] [54] [55]. It has come to be known that artificial intelligence consists of a series of machine learning algorithms [56]. Machine learning algorithms are divided into the following categories: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning, each of which contains many more specific algorithms [57] [58]. The RBF neural network algorithm used in this research is one of the representative algorithms of supervised learning. Although machine learning has helped to achieve excellent results in image detection and recognition, speech recognition, games and many other applications, the research is ongoing for combining it with robot control to improve the intelligent behaviour of robots.

At present, the supervised learning algorithm and reinforcement learning algorithm are used in robotics. In 2016, the University of Berkeley trains robots with convolutional neural networks, which enables robots to complete tasks such as screwing bottle caps [59]. Giusti et al. of Zurich University mount a monocular camera on the Unmanned Aerial Vehicle (UAV), and realize the real-time planning and control of UAV motion on real forest roads by using deep neural networks [60]. In the same year, Deep Mind collects a large amount of data on 14 robot arms and trained these to grab objects by means of enhanced learning [61]. After nearly 3,000 hours of training and more than 800,000 grabbing attempts, the level of intelligence of robot arms for grabbing random objects have been evidently improved. In addition, Deep Mind and Google have worked on training robots to open doors by using deep reinforcement learning [53]. In 2016, Mark Pfeiffer et al. propose an end-to-end learning algorithm based on a convolution neural network for ground mobile robots [54]. Using 2D laser sensor to sense information about the surrounding environment and by using data-driven methods, the path planning of indoor robots is realized, where the planned path is compared with the one planned by the ROS on the actual robot.

In [62], an approach to overcome the shortcomings of Wiener filtering is proposed called the KF, and it is used to estimate past, current, and future state signals regardless of the knowledge about the exact nature of the model. Filtering is a signal processing and transformation technique. There are dynamic uncertainties in a robot's functioning which have consequences in its teaching and learning experience. These uncertainties can be controlled with the development of controls using Fuzzy Logic or Radial Basis Function (RBF) Networks.

RBF is a three-layer forward network with a single hidden layer [63]. Its first layer is

an input layer and consists of signal source nodes, and its second layer is a hidden layer. The total number of hidden layer nodes depends on the complexity and requirements of the problem described. The transformation function of the neurons in the hidden layer, for example the radial basis function, is a non-negative linear function that is radially symmetric and attenuated to the center point [63]. It is a local-response function, and the specific local-response is reflected in the transformation of its visible layer to the hidden layer, a phenomenon which differs it from other networks. The third layer is the output layer which responds to the input mode, and therein the input layer only acts as a transmission signal [64]. Previous forward network transformation functions are used for a global response. Functionally, the input layer and the hidden layer can be connected with a weight factor of 1. The output layer and the hidden layer perform different tasks, hence learning strategies used for them will also be different [65]. The output layer adjusts to the linear weight using a linear optimization strategy, so the learning speed becomes faster. However, the hidden layer adjusts parameters of the activation function (Green's function and Gaussian function, more commonly the latter) through nonlinear optimization strategies which make the learning process slower [66].

According to [67], research about the mutual relationship of machine learning and robot control is ongoing. Through the above examples of supervised learning, enhanced learning and enhanced control, it is clear that extensive research is still required to create autonomous and truly intelligent robots. Despite its progress in image and voice recognition, combining machine learning with control technology still remains challenges. This thesis summarizes three reasons for combining machine learning with control technology:

a) Machine learning that solves traditional problems is "independent", implying that all inputs are assumed to be independent and identically distributed by default. Taking image classification as an example, different individual images do not affect the output of each other. However, for robot control, specific time-sequence and correlation between front and back behaviour of the robot are required.

b) Most of the problems that have been solved before machine learning are "no subject". "No subject" means that in the process of the machine learning algorithm, the only input image and output results are needed. However, for effective control of the robot, information about the state of the robot also needs to be considered apart from the input of environmental information.

c) Most of the problems solved in machine learning are "static", while robot control is a dynamic process as robot's behaviour and environment constantly interact with each other. Information comes from the environment may affect the decision-making of the robot. After the robot executes its decision, the observed environment also changes. Hence, a robot's environment and its state are always in a dynamic state of change.

Based on the requirement of human-robot skill transfer in terms of intelligence and autonomy, this thesis studies the key issues involved the process of controlling robot behaviour (such as remote teleoperation, target/position detection, decisive motion planning

and human-action imitation), and the design and implementation of an advanced robot control system based on machine learning techniques. In this process, we have changed "independence" into the "association", "no subject" into "with subject" and "static" into "dynamic" to implement an optimum combination of machine learning with robot control. Therefore, the intellectualization is actualized, and the combination of machine learning with robot control is validated.

2.5 Robot Controller Design

In robotics, control methods are derived to control both the position and the velocity of the joint motor. There are two main categories in the robot control broadly: kinematic control and dynamic control. In the kinematic control, the control input is often the data related to the joint positions or velocities, while in the most cases of dynamic control methods, the input data are relevant to the joint torque [68].

2.5.1 Robot Modelling

Establishing a dynamic robot model is the foundation of robotics and controller design research. Currently, researchers endeavour to improve the dynamic performance of robots while reducing incurred cost alongside achieving high speed and minimizing the heavy load. As robots' load-to-weight ratio decreases, robots become more flexible in design. With the enhanced flexibility, a dynamic model which accurately describes the characteristics of a robot is built, and the feed-forward torque is calculated based on the dynamic model, which improves the response speed of a robot.

The accurate dynamic robot has wide application in a number of fields – manipulator motion simulation, motion control system design, mechanical design analysis, etc. Some control schemes, such as predictive control [69], sliding mode control [70] and computed torque control [71], require an accurate model of robot dynamics. The robot dynamics models are commonly presented as:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_{ext} = \tau \tag{1}$$

where q denotes the vector of joint angles, $M(q) \in R^{n\times n}$ represents the definite inertia matrix, and n denotes the degree of freedom (DoF); $C(q,\dot{q})\dot{q} \in R^n$ denotes the Coriolis and Centrifugal torques. Moreover, $G(q) \in R^n$ represents torques due to the gravitational force, $\tau \in R^n$ is the control input vector and τ_{ext} is the external disturbance. Here, we define the kinetic energy of the robot as: $M(q)\ddot{q} + C(q,\dot{q})\dot{q}$, and the potential energy in terms of gravity as $G(q)$. This calculation can be used to further calculate forward dynamics for the simulation to measure manipulator motion based on the applied control input, or inverse dynamics for control of robots by obtaining torque for a given set of joints.

Based on a robot's specific geometry and inertia parameters, there are two commonly used methods for developing the dynamics as given in (1): the Lagrangian-Eulerian (L-E) formula and the Recursive Newton-Eulerian (RN-E) method [72]. Both types of the research detail methods to describe the dynamic execution of robot motion.

The L-E method is based on a simple, systematic approach to obtaining values of the kinetic and potential energy of a rigid body system. In this regard, Bajeczy's work illustrates dynamic equations of motion for a robot. This robot is highly nonlinear, consisting of terms for inertia and gravity, and depending on physical parameters and configuration of the relationship among position, angular velocity and acceleration [73, 74]. According to [75], the L-E method provides a closed form of robot dynamics, and is therefore suitable for analytical calculations. It can also be used to design joint space (or task space, if the robot Jacobian matrix from the base to the end-effector is available).

Additionally, the L-E equation can be used for calculating forward and inverse dynamics, however this involves calculating a wide variety of coefficients for $M(q)$ and $C(q, \dot{q})$ as outlined in the equation (1), which proves to be time-consuming. Hence, this technique is relatively unsuitable for computing on-line dynamic calculations, since alternative strategies, especially when the RN-E or Lee's Generalised d'Alembert Equations (GAE) calculate with fewer derivations and with higher speed [76]. In the research [77], an algorithmic for L-E technique is developed which greatly reduces the computational burden of the L-E formulation and aligns it with RN-E strategies.

The N-E formula is based on balancing out forces acting on the general link of a robot. This involves a series of equations with recursive solutions [78]. The forward recursion propagates link speed and acceleration, and then it recursively propagates forces and torque along the chain of robots. This method is more efficient than L-E since it utilizes a serial chain as a manipulator; when a force is applied to a link, it can also produce motion in the connected link. Considering the considerable computational duplications, the algorithm is expressed in a recursive form [79]. The reduction in computational load greatly reduces overall computation time, allowing real-time forward and reverse dynamic calculations, which assists in implementing real-time torque control methods.

2.5.2 Kinematic Control

Kinematics control is responsible for solving the inverse kinematics, i.e. generating the joint position or velocity trajectory when the task trajectory in the Cartesian space and the initial posture of the robot are given. One fundamental example of kinematics control is the proportional integral derivative (PID) control; the principle of PID control is illustrated in Fig.2.4. The input $r(t)$ and output $c(t)$ of the system denote the velocity of the system, respectively. The velocity response of the system can be adjusted via K_P. Moreover, the position response can be manipulated by tuning K_I, and the acceleration can be adjusted by tuning K_D.

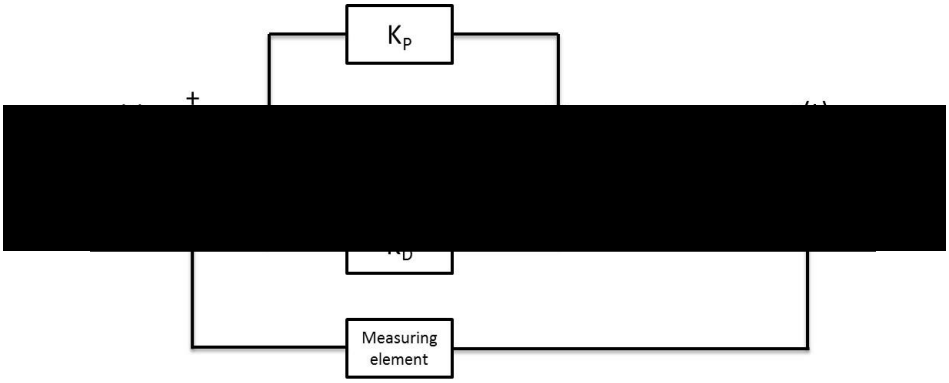


Figure 2.4: Flowchart of the PID control architecture [3]

In [80], the authors propose a fuzzy partition controller mechanism based on error partitioning method. This method combines fuzzy control technology with error partitioning PID controller. In this research, the design of the controller and improvement of system control performance is explained. Finally, the experiment proved that the proposed method can greatly improve the overall performance of the control system. In another study [81], the authors design a nonlinear PD controller for a manipulator with six degrees of freedom and four-bar linkage trajectory tracking system to overcome a number of issues. For online trajectory planning and kinematics control of modular three-legged parallel robots, [82] designs an inverse kinematics method for directly solving the issue of joint angular displacement based on local exponential product formula. Moreover, according to [83], the author designs a PD controller plus gravity compensation control law for a four-bar linkage mechanism and achieved good control performance on the experimental platform. Unfortunately, all of the above-mentioned designs have limitations in terms of variability and dexterity.

The study [84] improves upon the design of the one [81] by developing a nonlinear PID controller. Through comparison, it is inferred that the designed controller performance that they achieved is better than PD. Based on the kinematics model of a six-degree-freedom parallel robot [85], a nonlinear PID controller has been designed to overcome problems of system interference and noise measurement. Experimental results demonstrate that the controller is easy to implement and has good control performance. In the research [3], the trajectory tracking control is studied based on the kinematics model of the robot. The trajectory tracking control of the mobile robot is divided into two categories – trajectory tracking controller (based on back-stepping and time-varying state feedback) and robot velocity PID controller. In another study [86], back-stepping method is used to construct a mobile robot trajectory tracking controller, which during the design disintegrates mobile robot system into low-order subsystems. To this end, intermediate virtual control and partial Lyapunov function are used to simplify controller design such that accurate

characteristics for tracking error convergence are obtained.

A robot trajectory tracking control system based on kinematics mode offers a number of advantages including simple structure, less computational burden, and ease of implementation. However, the ignorance of the dynamic characteristics limits the high performance of the mobile robot, since actual control designs of robotics need to coordinate all joints for optimal efficacy.

2.5.3 Dynamic Control

A dynamic controller solves the inverse dynamics problem and calculates the torque required by each joint of the robot. This is executed after the kinematic model has planned out the desired trajectory for each joint to drive joint motion. In the study [87], the coefficient matrix of the Stewart platform dynamic equation is assumed to be a constant, and a PID inverse dynamics controller is designed. In the process, the error is approximated as interference, and a compensation controller is developed. The experimental results show a better performance of the designed controller than that obtained by using an uncompensated PID inverse dynamics controller, and it is able to obtain the faster reference signals. In another study [88], kinematic controller and dynamic PD controller are designed for redundantly driven parallel robots. Subsequently, a comparison of control performance comparison is conducted on the experimental platform for both controllers, which shows a better performance of the dynamic controller. On the basis of the study [88], a research team design a PD controller, an augmented PD controller and torque controller based on the dynamic learning model. They then compare the performance of the three controllers [89].

In addition, computed torque control, is also a commonly used control scheme for robot manipulator, where the nonlinear terms of the dynamics model are compensated directly. This information is sent to a robot which provides feedback regarding position and velocity which are used for control. Command torque τ is calculated using the above equation (1). Fig.2.5 is an illustration of the diagram of computed torque control, where s, v, a denote the position, velocity and acceleration of the reference trajectory, respectively, and q, $\dot{q}$ represent the position and velocity feedback of the robot. The gains K_p and K_v can be modified to alter the stiffness and damping of the system.

According to the research [4], a computational torque controller based on neural network optimization is designed for achieving high-speed motion on 2-DOFs parallel manipulators. In another study [90], the authors propose a modeling method of differential geometry for a multi-joint robot with a position/force hybrid control algorithm, and the experiments are carried out on a series of multi-joint robot prototypes with different degrees of freedom. Moreover, in [91], a dynamic analysis method based on linear projection mapping is established and there are some controllers that consider performance indicators for robot motion control. This strategy solves the point-to-point motion control problem of

Figure 2.5: Flowchart of the computed torque control architecture [4]

parallel robots, and also proves that the control method can effectively improve system performance. The advantage of dynamic control is its capacity to consider external force interference such as uncertainty of kinetic parameters and friction. Furthermore, the study [92] develops a robust nonlinear controller for Stewart platform motion control, introducing Friedland-Park friction mechanism, estimating frictional force to improve the system's robustness.

3 Chapter Three: Preliminary

3.1 NAO Robot

In this research, we chose the NAO robot, a 5 DOF humanoid robot produced by Aldebaran-Robotics in France (see Fig.3.1) [93]. It supports multiple sensors and controllers, including head and jaw cameras, chest sonar sensors, movement motors on neck, hands and feet, three colour LEDs (red, green and blue) on the eyes, and head and feet tactile sensors [94]. The robot has a body mass index (BMI) of about 13.5 kg/m^2, relatively light as compared to other robots of the same height [94]. According to the research [95], the NAO robot has 25 degrees of freedom (DOF) in total, wherein its 11 joints are found in the legs and pelvis and the rest are located in the trunk region, arms and head. In addition, each arm consists of a 2 DOF shoulder, a 2 DOF elbow, 1 DOF wrist and 1 DOF hand-gripper. The head is also able to rotate on both yaw and pitch axes [95].

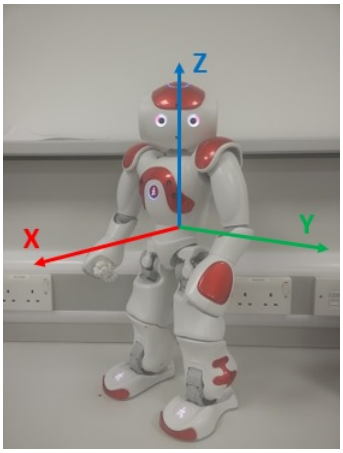


Figure 3.1: Illustration of NAO Robot and coordinate system

3.2 Baxter Robot

The Baxter robot was developed by Rethink Robotics in the United States. It is an innovative, intelligent and collaborative robot (Fig.3.2). It is an ideal alternative to manpower outsourcing and fixed work automation [96]. With its unique features and benefits, the Baxter robot enables manufacturers to create cost-effective solutions for handling small batches and a variety of producing tasks as well as minimizing the requirement of technical staff. There are many leading industrial companies that have gained a significant competitive advantage by incorporating the use of the Baxter robot [97].

The Baxter robot comprises of the following parts: one torso, one 2-DOF head and two 7-DOF arms, which are shoulder joint: s_0, s_1, elbow joint: e_0, e_1 and wrist joint:

w_0, w_1, w_2, respectively [98]. It also consists of coordinated cameras, torque sensors, encoders, and sonar. Researchers can directly program the Baxter robot using open-source such as a standard ROS interface. Seven Serial Elastic Actuators (SEAs) drive the joints in the Baxter robot arm which effectively regularizes the robot's movements and helps it to overcome the effect produced by surrounding obstacles [98]. Commonly, people teleoperate and program the Baxter robot using ROS (Robot Operating System) through the Baxter SDK running on Ubuntu Operation System. The ROS is an open-source system and comprises of libraries, module devices and correspondence [99]. The use of this system improves the task in terms of displaying and programming of the robot on various types of automated platforms [99].

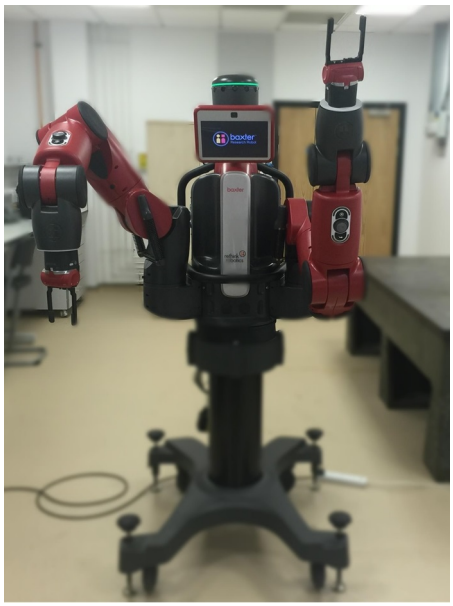


Figure 3.2: Image of the Baxter robot

3.3 KUKA LBR iiwa robot

KUKA LBR iiwa robot (a robot with human-robot collaboration capabilities) is the first commercial robot that is approved for human-robot collaboration (HRC). The KUKA L-BR iiwa robot aids human-robot collaboration for the completion of highly sensitive and precise tasks [100]. To ensure high control accuracy, the robot has an advanced design of 7 degrees of freedom (DOFs) robot arm [100]. The arms are programmed via Workbench, which is a standard KUKA modifying platform employing KUKA robot language (KRL) and Java [100]. The KUKA LBR is controlled by the KUKA SmartPad (Fig. 3.3).

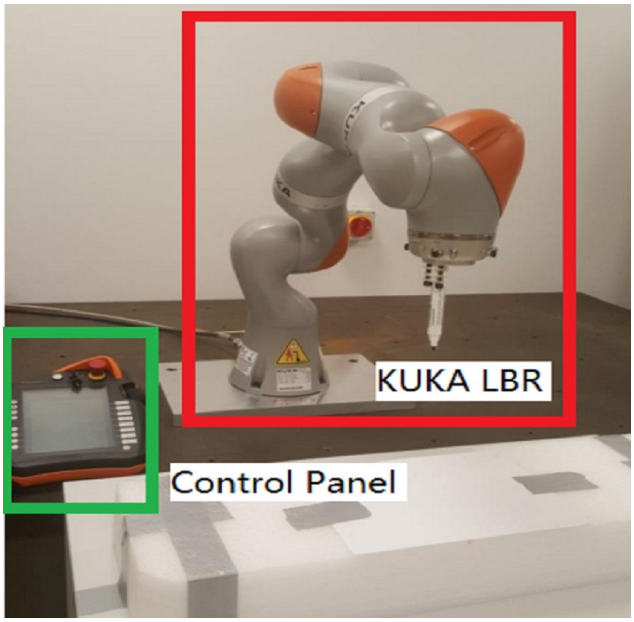


Figure 3.3: Image of KUKA LBR robot

3.4 Kinect v2

The Kinect v2, produced by Microsoft, is an RGB-D device, and it can be used to capture depth, colour, and IR images (as well as sound) using the Kinect 2.0 SDK (Software Development Kit) [101]. The depth, IR and colour image resolutions are significantly better than the ones obtained using the first generation Kinect v1. Using the SDK, the obtained information about colour and depth can be consolidated (transformed) into real-world co-ordinates, which are called camera space. These co-ordinates are directed towards the centre of the depth sensor [102]. Skeletal tracking can also be achieved with the use of the Kinect 2.0 SDK and this function has been used in this project. Consequently, this technique is used to track the position of an operator in front of a Kinect device. When a point is selected by the operator on the Kinect colour image, its depth information is acquired in the Kinect depth image. In the frame that is constructed, both the colour image, and depth image are put into the same frame, and its origin is located at the centre of the depth camera. In this regard, the coordination system of the camera space follows a right-hand convention (see Fig.3.4).

3.5 MYO Armband

The MYO armband (shown in Fig. 3.5) is a wearable device produced by Thalmic Labs. With the MYO armband, an operator is able to communicate with the system via Blue-

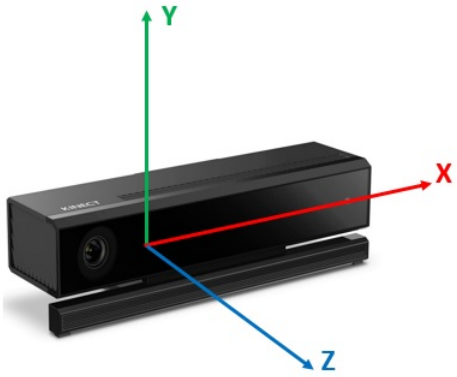


Figure 3.4: Illustration of the origin of the Kinect v2's camera space. It is the same as its depth sensor origin which is a modified form of [5]

tooth. In addition, it has 8 built-in EMG sensors and one IMU sensor with 9 axes. These assure that the hand posture and arm motion can be efficiently detected when people move their arm muscles as the EMG sensors can distinguish the difference in hand gestures. Since every user has distinctive muscle size, skin type and other key differences, the sensors create a catalogue of information via electrical driving forces. For this purpose, a calibrating process is essential for MYO armbands to identify the wearer's motions and gestures.

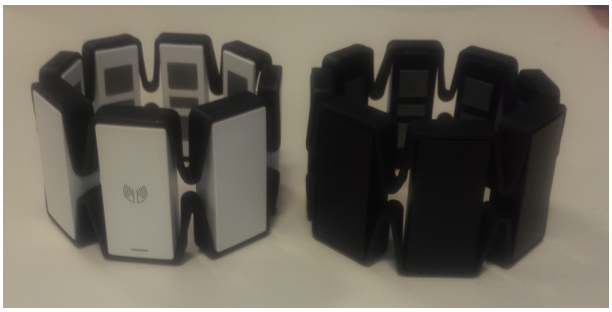


Figure 3.5: Image of MYO Armband

3.6 Dragon NaturallySpeaking

Dragon NaturallySpeaking is a speech recognition software created by Dragon Systems, a company located in Newton, Massachusetts [103]. Dragon NaturallySpeaking helps operators in creating documents, reports, emails, and fill-in forms as well as workflow sheets through verbal commands [103]. The words appear as text in Microsoft Office

Suite, Corel WordPerfect, and all Windows-based applications as they are spoken to the computer. Moreover, operators are able to create voice commands to run applications on the computer in a multiple-step process for convenience.

3.7 Dynamic Movement Primitive

The biomimetic robot expert Ijspeert proposed a nonlinear dynamic system control method in [104]. This method uses a series of linear differentiated equations to model the overall motion of the robot into a nonlinear dynamic attractor model by adding an automatic learning term. This method can model discrete motions (such as writing tasks) as well as rhythmic movement such as drumming [104]. DMP utilizes a comprehensive and dynamic system to express the diagram of motion trajectories. The DMP model is stated as follows [105]:

$$\tau_s \dot{v} = d(g - x) - cv + (g - x_0) f_{target} \tag{2}$$

$$\tau_s \dot{x} = v \tag{3}$$

where $x \in R$ is the Cartesian position, x_0 is the initial position, $v \in R$ is the velocity of the robot end-effector, τ_s is the scale factor which affects the speed of the generated motion, g donates the target position, f_{target} is a forcing term, which is a nonlinear function, d and c are the coefficients for spring and damping, respectively. Additionally, x, v, $\dot{v}$ represent the position, velocity, and acceleration, respectively.

DMP has been developed through investigations and studies that helped in learning about primitive movements to generate a sophisticated model. The concept of DMP can be divided into two categories: the states that use unique formulations based on dynamical structures; and generated trajectories that are constructed via interpolating of factors [106]. DMP consists of 2 components – a converted system, r, and a canonical system, h. The formula for it is given as follows:

$$\dot{s} = h(s) \tag{4}$$

$$\dot{x} = r(x, s, w) \tag{5}$$

where x is the transformed system states, s is the canonical system states and w is the transforming parameters of the canonical system output.

3.8 Gaussian Mixture Model

The GMM actualizes the estimation of the probability density distribution in samples. The estimated model is the weighted sum of finite Gaussian models [107], in which each Gaussian model represents a class. The data in the sample are clustered into several Gaussian models and the probability of each model is then obtained [107]. Subsequently, the largest

probability is selected by the GMM based on the results. The significance of GMM is constructing a series of GMMs to denote joint-density, and then obtaining the probability density and regression function from each GMM.

The GMM formula is defined as,

$$p(x_{exp}) = \sum_{k=1}^{k} p_k p(x_{exp} \mid k) \tag{6}$$

where k denotes the number of the model, p_k represents the weight of the k^{th} Gaussian, and it is also the k^{th} Gaussian probability density function, $p(x_{exp} \mid k)$ is the conditional probability distribution, which includes parameters such as the average value of GMM μ_k and the variance $\sum_k$. To estimate the probability density, p_k, μ_k and $\sum_k$ variables should be available. When the values in the expression are learned, the result of the summation is the probability of the sample x_{exp} belonging to each model.

3.9 Dynamic Time Warping

The DTW is a typical optimization method used to denote the time difference between the test time series and the reference one utilizing the time regular function W(n). It solves the regular function corresponding to the minimum distance of the two templates [6]. This algorithm is based on dynamic programming (DP), and it effectively matches the time series with different lengths [21]. It follows a classic algorithm for speech recognition. The DTW calculates the similarity between two time series by extending and shortening

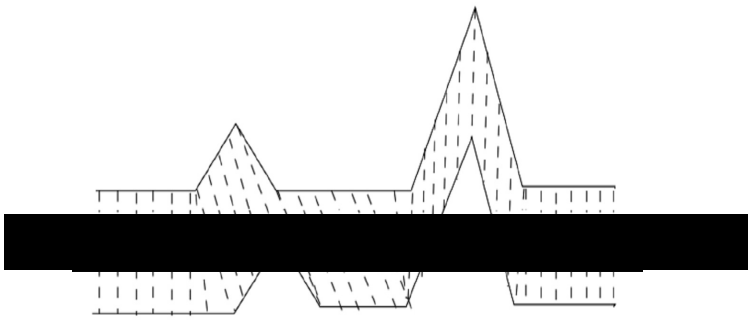


Figure 3.6: Warping example between two time series, modified from [6]

them. As shown in Fig.3.6, the upper and lower solid lines represent two time series, and the dashed lines located between the time series represent similar points. The DTW uses the sum of the distances between all similar points, termed the warp path distance, to measure the similarity between the two time series.

3.10 Kalman Filter

The KF method is used to optimally fuse the real-time, dynamic and low-level extra sensor data. It evaluates the statistical significance for the optimal fusion of data combination by using statistical characteristics of the measurement model. If the system has a linear dynamic model, the system noise and sensor noise can be represented using white noise models that obey the Gaussian distribution. In this regard, KF would provide the unique statistically optimal estimate for the fusion data. The recursive KF reduces the computational burden, and it is divided into two types – continuous-time KF and discrete KF.

The actual physical system is usually continuous, as a consequence, the description of discrete systems often cannot completely replace the continuous-time system. The temporal mathematical model of the system is produced below [108]:

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) + G(t)\omega(t) \\ y(t) &= H(t)x(t) + v(t) \end{aligned} \tag{7}$$

where x and u are n-dimensional state variables; y is m-dimension measurement vector; A is $n \times n$-dimensional system matrix; G and B are $n \times r$-dimensional system matrix; H is $m \times n$-dimensional measurement matrix; ω is the zero-mean white noise vector of the r-dimensional continuous system; v is an m dimensional vector denoting the potential white noise among the measured data.

The continuous-time KF status equation, based on [109] is stated as follows,

$$\begin{aligned} \dot{\hat{x}}(t) &= A(t)\hat{x}(t) + B(t)u(t) + K(t)\left[y(t) - H(t)\hat{x}\right] \\ K(t) &= P(t)H^T(T)r^{-1}(t) \\ \dot{P}(t) &= P(t)H^T(t) + A(t)P(t) - P(t)H^t(t)r^{-1}H(t)P(t) + G(t)s(t)G^T(t) \end{aligned} \tag{8}$$

where K represents the filter gain matrix, $\hat{x}$ is the estimated value of x, and P is the estimated covariance matrix. continuous-time KF is obtained from the measured values of the continuous-time process, and this method is used to estimate the time continuous-time value of the system state variable, which solves the differential equation of the matrix. Moreover, the continuous-time KF does not require complex recursive calculations. The working principle of KF is illustrated in Fig. 3.7.

3.11 RBF Neural Networks

In adaptive control, a neural network is a commonly used as function approximation tool. The unknown function term in the control system is estimated by the neural network. A neural network can be grouped into linear parameterization and nonlinear parameterization, corresponding to RBFNNs and multi-layer neural network (MNNs). RBF neural network consists of an input layer, a hidden layer and an output layer. The input layer to the hidden layer is a nonlinear transformation, and the output layer is a linear combination

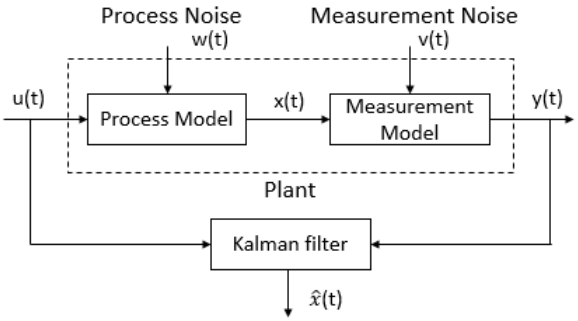


Figure 3.7: Working principle of continuous-time KF

of the output of the hidden layer. RBFNN is not only simple in structure and fast in learning, but also avoids the problems of many nerve layers. As a result, the RBFNN meets the real-time requirements of the control system. Therefore, in this thesis, the RBFNN is used to approximate the unknown terms of the control system.

A continuous function can be approximated using linear parameter RBFNNs, such as $F(z) : R^m \rightarrow R$, over a minimized set $\Omega_z \subset \mathbf{R}^m$. This can be formulated as, shown below in [110]:

$$F(z) = W^{*\mathrm{T}} S(z) + \epsilon_z \; \forall_z \in \Omega_z \tag{9}$$

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 minimal set, l is the NN node number, and ϵ_z is the estimation error. $S(z) = [S_1(\|z-\mu_1\|), \cdots, S_l(\|z-\mu_l\|)]^T$, is the regressor vector, with a radial basis function $S_i(\cdot)$, and μ_i $(i = 1, \cdots, l)$ a central inside $S_i(\cdot)$. The Gaussian functions are presented as follows:

$$S_i(\|z-\mu_i\|) = \exp\left[\frac{-(z-\mu_i)^T(z-\mu_i)}{\varsigma^2}\right] \tag{10}$$

where ς is the variance and $\mu_i = [\mu_{i1}, \mu_{i2}, \cdots, \mu_{im}]^T \in R^m$ represents the center of each receptive area. In case that if there is a large number of neurons, based on [110], the radial basis function (10) can approximate any continuous function with arbitrary precision. The evaluated weight $\hat{W}$ utilized to calculate $F(z) = \hat{W}S(z)$, where $\hat{W}$ is the estimated neural weight, will be detailed determined in the later Chapter.

3.12 Concluding Remarks

In this chapter, the relevant preparatory knowledge of mathematical theory is introduced in detail which lays down a foundation for the following chapters. At the same time, this

thesis introduces the RBFNN, which is used to approximate the uncertain dynamics of the control system. To minimize the undesired effect of sensor noise, we introduce signals fusion from multi-sensors based on KF algorithm. The external devices or sensors and experimental platforms used in this thesis are described in detail.

4 Chapter Four: Development of Kinect based Teleoperation of the NAO Robot

4.1 Introduction

Robotic technologies that deals with the teleoperation capabilities of robots are designed to perform specified tasks. Research on motion capture, human-based tracking systems etc. has attracted great attention in the last decade [14]. In research [15], the author developed a platform of multiple RGBD cameras for the Bayesian object tracking system using a Microsoft Kinect sensor. The kinematic validity of a Kinect-based skeletal tracking system is guaranteed with an upper limb virtual reality rehabilitation system in [19]. In another research [18], a model for tracking fingertips and the centre of the palm is developed based on the Kinect sensor. However, it is also imperative to focus on the research that explores head motion.

Head-tracking technology has a wide range of application, especially in face recognition and authentication, security, surveillance and human-robot interaction. However, it faces challenges such as the low resolution of the human face in videos, changes in illumination and motion blur, amongst other factors [111]. According to [112], a multilayer neural network controller for limb following is developed to overcome those challenges. In [113], a research team establishes a model based on their study results for analysing human motion in order to create a robot arm that mimics the movements of a human arm.

The Kinect sensor, with its in-depth information, has an increasing number of applications in special areas like gaming, entertainment, health and fitness. In early 2014, Microsoft developed a new generation of sensors: Kinect v2. This new advancement comes with a Kinect development kit SDK, which helps Kinect v2 to collect accurate color image information with depth, and thus became a more viable option for the head-following project [20]. Robot visual based detection and tracking systems have been successfully applied in many fields, such as manufacturing industry, military and medical areas. However, because of the complexity of vision, the extraction of visual information needs complex algorithms to support, and the processing process of these image algorithms usually is time-consuming, which makes the real-time performance of the system difficult to meet the satisfaction.

NAO robots are employed in diverse fields such as computer science, mathematics, physics and artificial intelligence [114]. It has also been found useful in schools due to the ease of use [115]. With the popularity of its utilization in the education sector, NAO robot is programmed for various practical applications, for example, one Swedish university introduces voice recognition and visual feedback for students by establishing a laboratory of NAO robots [116]. Commercially, NAO robots have been used in large production launches to assure the tech-compliance. This section includes a discussion of the Kinect v2 camera and NAO robot. Fig.4.1 shows the control block of the system which has been

implemented to verify the effectiveness of our approach.

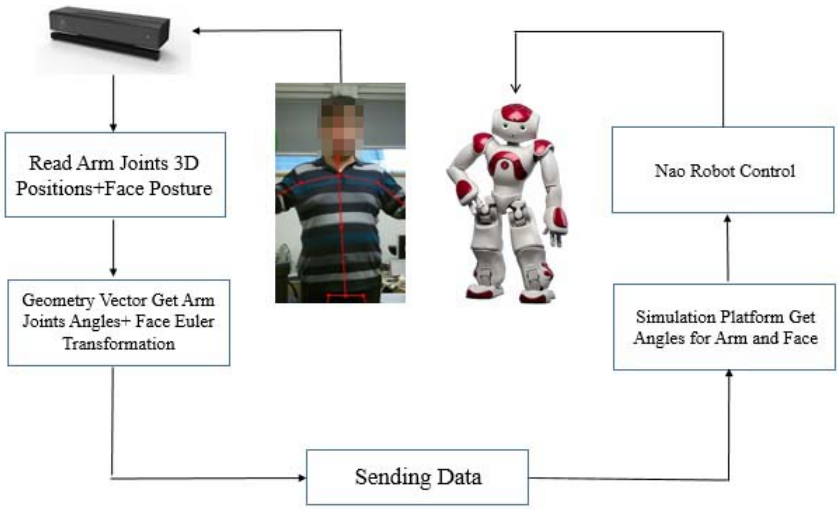


Figure 4.1: Flowchart of the complete experimental process

4.2 Calibration

Before performing the tracking task, the transformation between the NAO robot's coordination and the Kinect's is needed, as shown in (11):

$$X_i = TX_i^{'} \tag{11}$$

where T represents the transformation matrix. Hence, mathematically X_i=$[x_i y_i z_i 1]^T$ and $X_i^{'}$=$[x_i^{'} y_i^{'} z_i^{'} 1]^T$ are the coordinates under both the robot and Kinect systems.

We can identify the transformation matrix T by calculating four points under both the robot and Kinect coordinates systems. Based on the assumption that there are four points – p_1, p_2, p_3 and p_4, and that these coordinates lie in the robot coordinate system and the Kinect coordinate system, respectively. Fig.4.2 shows one of the four points in the relationship between the NAO robot and the Kinect's coordinate system. (12) illustrates the equation to calculate the transfer matrix T from the Kinect coordinates to the robot coordinates.

$$T = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1^{'} & x_2^{'} & x_3^{'} & x_4^{'} \\ y_1^{'} & y_2^{'} & y_3^{'} & y_4^{'} \\ z_1^{'} & z_2^{'} & z_3^{'} & z_4^{'} \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \tag{12}$$

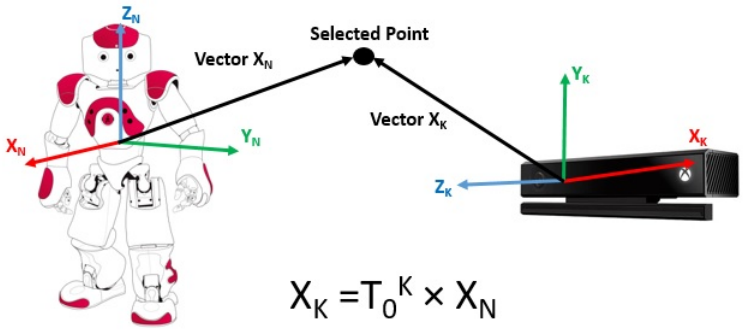
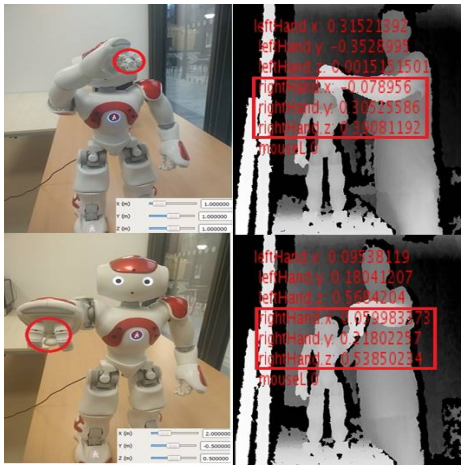
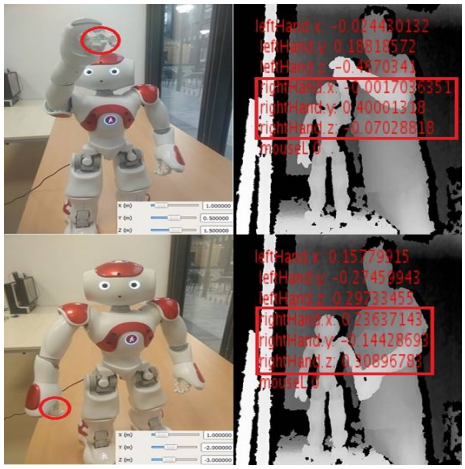


Figure 4.2: Illustration of the relationship between NAO Robot and Kinect coordinate system

In this research, four points were chosen as shown in Fig.4.3. To acquire the positions under the Kinect coordinate system, a person stood in front of a Kinect sensor and held his right hand in the same position as that of the NAO robot. The selected four positions under NAO robot coordinate system are: (1, 1, 1), (2, -0.5, 0.5), (1, 0.5, 1.5), (1, -2, -3), which corresponding to the positions under the Kinect coordination: (-0.0790, 0.3053, 0.3908), (0.0600, 0.2180, 0.5385), (-0.0012, 0.4000, -0.0703), (0.2364, -0.1443, 0.3090), respectively.



(a) : The first and second position in NAO and Kinect coordinate frame



(b) : The third and fourth position in NAO and Kinect coordinate frame

Figure 4.3: Illustration of the four selected positions under both NAO coordinate system and Kinect coordinate system

Substituting all the values on both the NAO and Kinect system into (12), we can calculate the transformation matrix of Kinect underground-based coordinate system T_0^k, based

on the assumption that the NAO robot remains stationary.

$$T_0^k = \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & -0.5 & 0.5 & -2 \\ 1 & 0.5 & 1.5 & -3 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -0.079 & 0.06 & -0.0012 & 0.2364 \\ 0.3053 & 0.218 & 0.4 & -0.1443 \\ 0.3908 & 0.5385 & -0.0703 & 0.309 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \tag{13}$$

Then, we would obtain:

$$T_0^k = \begin{bmatrix} 7.87 & 5.09 & 2.37 & 6.58 \\ -10.27 & -0.4 & -0.73 & -1.26 \\ 1.25 & 9.58 & 1.09 & -0.39 \\ -0 & 0 & 0.0000 & 1.00 \end{bmatrix} \tag{14}$$

Furthermore, the rotation matrix of the Kinect underground-based coordinate system, R_0^k, is obtained on its transformation matrix, which is provided as follows:

$$R_0^k = \begin{bmatrix} 7.87 & 5.09 & 2.37 \\ -10.27 & -0.4 & -0.73 \\ 1.25 & 9.58 & 1.09 \end{bmatrix} \tag{15}$$

4.3 Kinematics Methodology

An image stream of the workspace, is captured by the Kinect v2 colour camera. In order to select a point in the colour image, the operator requires to transform the pose of the end effector of NAO from the Kinect colour space to the NAO coordinate space. The pixels of the colour image are thus transformed into the camera space as follows:

$$[x_c \quad y_c \quad 1]^T = R \cdot [x_k \quad y_k \quad 1]^T + T_{Trans} \tag{16}$$

where x_c and y_c find the position of a transformed colour frame pixel (in camera space), x_k and y_k denote the position of the colour frame pixel, R is the rotational matrix, and T_{Trans} is the translation matrix, between the colour frame and camera space. However, the Kinect v2 depth camera faces inherent camera issues, such as radial distortion (due to the curvature of the lens). Fortunately, the intrinsic parameters are provided in one of the Kinect SDK examples, and can be used to calculate the distortion transformation matrix. At this point, it becomes possible to transform any point in the camera space to NAO's workspace, where the NAO's origin acts as its reference point.

4.3.1 Acquisition of Orientation Angles of Head

By capturing the pose of a user's head, the orientation angles α_f^k, β_f^k, γ_f^k are acquired by the Kinect v2. In order to transmit the data of orientation angles for user's head, the

angles need to be transformed into those under the same coordination, here defined as the reference coordinate, as shown in Fig.4.4. In this section, the flat surface of the Kinect camera is kept parallel to the user's head, such that the value of β_f^k equals zero. Thus, we have the following rotation matrix:

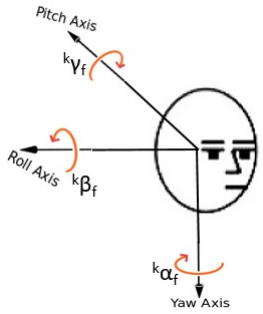


Figure 4.4: Illustration of the conception for human head orientation angles

$$R_k^f(\alpha_f^k, \beta_f^k, \gamma_f^k) = \begin{bmatrix} c\alpha_f^k c\beta_f^k & c\alpha_f^k s\beta_f^k s\gamma_f^k - s\alpha_f^k c\gamma_f^k & c\alpha_f^k s\beta_f^k c\gamma_f^k + s\alpha_f^k s\gamma_f^k \\ s\alpha_f^k c\beta_f^k & s\alpha_f^k s\beta_f^k s\gamma_f^k + c\alpha_f^k c\gamma_f^k & s\alpha_f^k s\beta_f^k c\gamma_f^k - c\alpha_f^k s\gamma_f^k \\ -s\beta_f^k & c\beta_f^k s\gamma_f^k & c\beta_f^k c\gamma_f^k \end{bmatrix} \tag{17}$$

where $R_0^f = R_0^K \times R_K^f$, and the value of R_0^K has already been obtained through calibration, as shown in (15), the rotation matrix R_0^f for the user's head under the reference coordinate system can be found through (18), where we denote the abbreviations $s\alpha_f^k$ for s_a, $s\beta_f^k$ for s_b, $s\gamma_f^k$ for s_g, $c\alpha_f^k$ for c_a, $c\beta_f^k$ for c_b, $c\gamma_f^k$ for c_g.

$$R_0^f = \begin{bmatrix} 7.87c_ac_b - 10.27(c_as_bs_g + s_ac_g) + 1.25(c_as_bc_g + s_as_g) & 5.09c_ac_b - 0.4(c_as_bs_g + s_ac_g) + 9.58(c_as_bc_g + s_as_g) & 2.37c_ac_b - 0.73(c_as_bs_g - s_ac_g) + 1.09(c_as_bc_g + s_as_g) \\ 7.87s_ac_b - 10.27(s_as_bs_g - c_ac_g) + 1.25(s_as_bc_g - c_as_g) & 5.09s_ac_b - 0.4(s_as_bs_g + c_ac_g) + 9.58(s_as_bc_g - c_as_g) & 2.37s_ac_b - 0.73(s_as_bs_g + c_ac_g) + 1.09(s_as_bc_g - c_as_g) \\ -7.87s_b - 10.27c_bs_g + 1.25c_bc_g & -5.09s_b - 0.4c_bs_g + 9.58c_bc_g & -2.37s_b - 0.73c_bs_g + 1.09c_bc_g \end{bmatrix} \tag{18}$$

Finally, the orientation angles of user's head under the basic ground coordinate can be

calculated, which are shown in (19)-(21), where we define $R_0^f = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$.

$$\alpha_f^0 = arctan\frac{r_{21}}{r_{11}} \tag{19}$$

$$\beta_f^0 = arctan \pm (\frac{\sqrt{r_{11}^2 + r_{21}^2}}{r_{31}}) \tag{20}$$

$$\gamma_f^0 = arctan\frac{r_{32}}{r_{33}} \tag{21}$$

4.3.2 Acquisition of Joint Angles of Arm

Kinect sensor emits infrared rays and detects their reflections so that the depth values of each pixel in the field of view can be calculated. In this process, first, information about the object body and shape are extracted. Then with this information, the position of each joint can be obtained as shown in Fig.4.5.

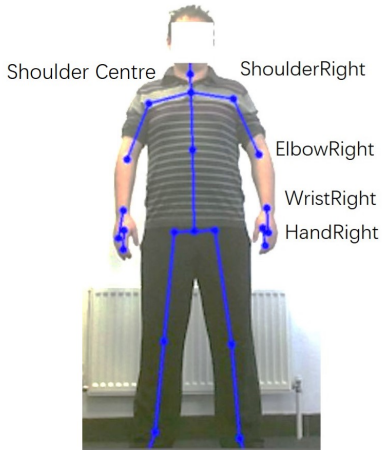


Figure 4.5: Image of body skeleton captured by Kinect

To obtain the angle of rotation of *ShoulderPitch*, we can use joints – *RightShoulder* and *CentreShoulder* in 3D coordinates and the two skeletal nodes in the three-dimensional space form a straight line (l_1), assuming that the 3D coordinates of the right shoulder are (x_1, y_1, z_1) and of centre shoulder, (x_2, y_2, z_2). As the *ShoulderPitch* joint remains unchanged during the rotation in the y-coordinate, only the *xoz* plane will be considered. The formula thus comes out to be:

$$z = k_1 x + b_1 \tag{22}$$

where k_1=$tan\theta_1$ = $\frac{|z_2-z_1|}{x_2-x_1}(x_1 \neq x_2)$, b_1 is not given for calculating the angle of rotation, so the formula is not given here. The angular value between l_1 and the horizontal axis is defined as θ_1 (*ShoulderPitch*) as shown in Fig.4.6, which can be computed by:

$$\theta_1 = arctank_1 = arctan\left(\frac{z_2 - z_1}{x_2 - x_1}\right) \tag{23}$$

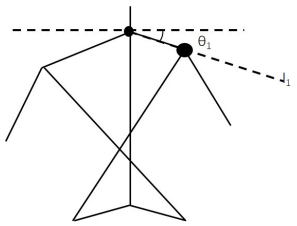


Figure 4.6: Angle of rotation of the *ShoulderPitch*

To obtain the angle of rotation of the *ShoulderRoll*, the 3D coordinates (x, y, z) of joints – *ShoulderRight* and *ElbowRight* will be used to calculate it, assuming that the three-dimensional coordinates of *ElbowRight* are (x_3, y_3, z_3), and the two skeletal nodes in the three-dimensional space form a straight line (l_2). As the *ShoulderRoll* joint remains unchanged during the rotation in the z-coordinate, only the *xoy* plane will be considered. The linear equation is given below:

$$y = k_2 x + b_2 \tag{24}$$

where k_2=$tan\theta_2$ = $\frac{|y_3-y_2|}{x_3-x_2}(x_3 \neq x_2)$, b_2 is not given for calculating the angle of rotation, so its formula is not given here. Assuming that the angle between l_2 and the vertical axis is θ_1, as shown in Fig.4.7, θ_2 is the *ShoulderRoll* rotation angle, the formula will be computed as:

$$\theta_2 = arctank_2 = arctan\left(\frac{y_3 - y_2}{x_3 - x_2}\right) \tag{25}$$

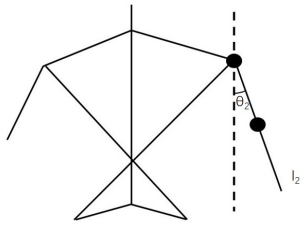


Figure 4.7: Angle of rotation of the *ShoulderRoll*

To obtain the angle of rotation of the *ElbowRoll*, we can use joints of *ElbowRight* and *WristRight* in 3D coordinates. Assuming the 3D coordinates of *WristRight* is (x_4, y_4, z_4),

similarly, there is a straight line constituted by *ElbowRight* and *WristRight* (l_3), the linear equation is given as:

$$y = k_3 x + b_3 \tag{26}$$

where k_3=$tan\theta_3 = \frac{|y_4 - y_3|}{x_4 - x_3} (x_4 \neq x_3)$, b_3 is not given in the calculation of the angle of rotation, so the formula is not given here. Assuming that the angle between l_2 and l_3 is θ_3, as shown in Fig.4.8, θ_3 is the *ElbowRoll* rotation angle, and the formula will be:

$$\theta_3 = arctank_3 = arctan\left(\frac{k_2 - k_3}{1 + k_2 k_3}\right) \tag{27}$$

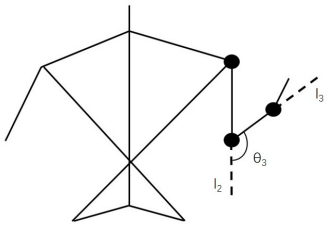


Figure 4.8: Angle of rotation of the *ElbowRoll*

4.4 Experimental Studies

In this section, some experiments have been designed to test the performance of the tracking system, with the help of the software: Kinect SDK for windows, Visual Studio 2013 and OpenCV library. The experimental environment is indoor and well-illuminated. One person stands in front of the Kinect sensor, at a distance of two metres. Because the NAO robot's arm has only 5 DOFs, the *ShoulderPitch*, *ShoulderRoll*, *ElbowRoll*, *ElbowYaw* and *WristYaw* joints of the human arm are chosen for the experiment.

4.4.1 Head Following Experiment

The purpose of the experiment was to test the stability of the head-following system. For this purpose, a command from the program developed by the Kinetic SDK 2.0 was sent to an additional computer which controls the NAO robot to imitate the movement of the human head. As the evidence shown in Fig.4.9, the experiment was conducted four times with different postures in order to get a total performance check. The postures included 30° rotation to the left, 30° rotation to the right, pitching up by 30° and pitching down by 30°. Fig.4.10(a) shown below illustrates the first experiment. The collated data show that head movements were accurately followed.

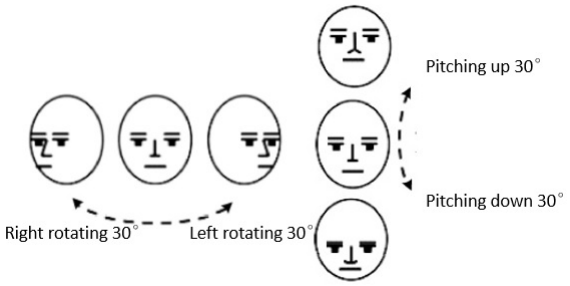


Figure 4.9: Illustration of operator's head for different postures

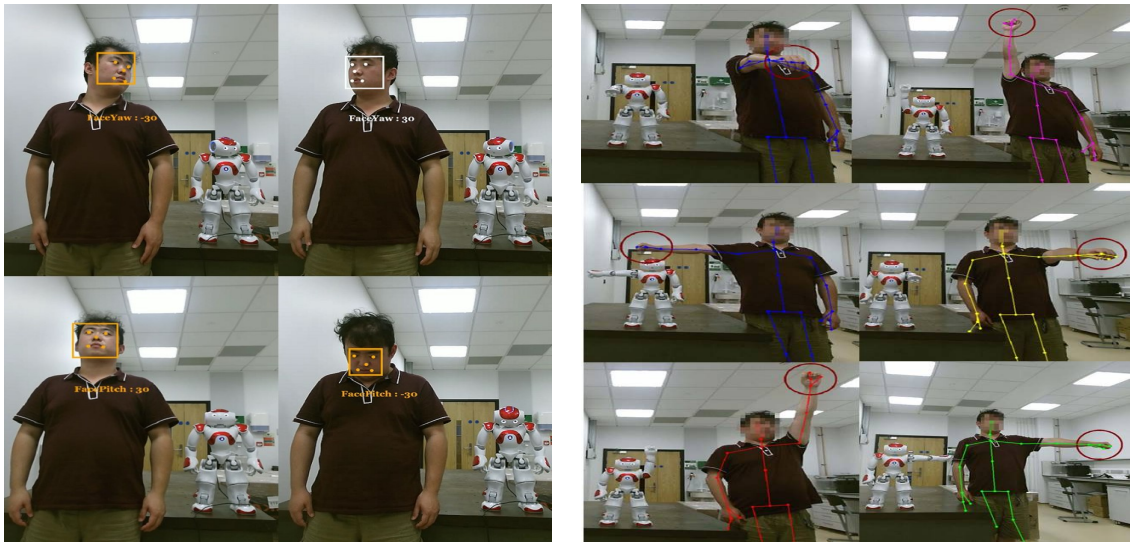
From the Fig.4.11, we can find that the average tracking error of HeadPitch angle (at about 10%) is a bit more than the one of HeadYaw angle (at about 6%). Generally, the head motion of the NAO robot is able to track the one of the human operators. Thus we can conclude that the NAO robot imitates the movement of the human head well in both two orientations.

4.4.2 Limb Following Experiment

In the second experiment, the operator moved his limbs while keeping the rest of his body still. The limbs were moved forward at 90°, raised up at 180° and stretched sideways at 90° at a low speed from the original position to the final position and for both right and left arms. As in the case of the first experiment, after collecting the data, a command was sent to the additional computer which controls the NAO robot to imitate the movements. Fig.4.10(b) shows the process of the second experiment. Fig.4.12 represents the graph illustrating the data collected about the arm movement by both the operator and NAO robot. It can be concluded from the results that the robot can completely follow the movements of the 5 selected joints.

From the plotted graphs, according to the (28), we can conclude that, angles of the selected joints including both the head and arm have the average error at about 10% with the maximum error at about 22%. This means using the Kinect camera sensor alone to track the motion of a human operator shows commonly large errors, hence in the next chapter, the KF based sensor fusion will be employed to minimize the undesired effect due to the noise from the sensors.

$$r_e = \frac{p_{NAO} - p_{Kinect}}{p_{Kinect}} \tag{28}$$



(a) : Process of imitation for NAO robot head following: Rotating 30° to left and right directions and pitching up and down with 30°

(b) : Process of imitation for NAO robot limb following: Putting forwards at 90°, raising up at 180° and stretching sidewards at 90° for both arms

Figure 4.10: Experimental setup of head and limb following tests

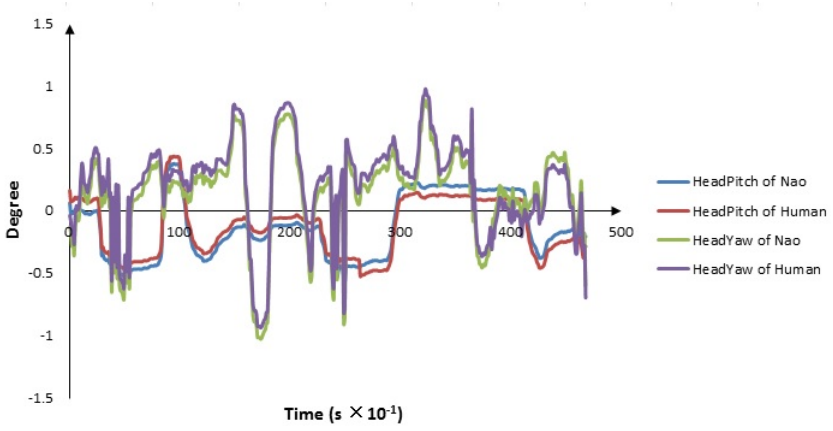


Figure 4.11: Result of the head following experiment for both human and NAO robot

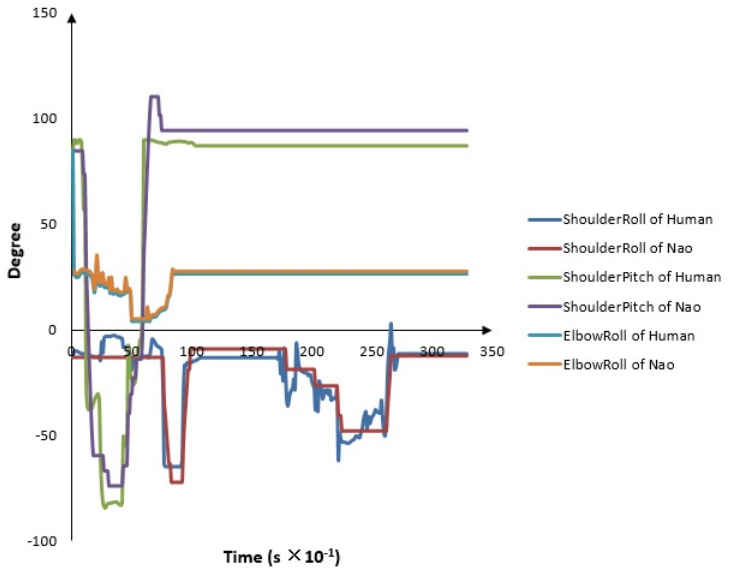


Figure 4.12: Result of the arm following experiment for both human and NAO robot

4.5 Conclusion

A system for tracking a user's head and arm movement while interacting with the NAO robot via a Kinect v2 sensor is developed. The Kinect v2 sensor is used to track the head and arm motion of the human operator which are moved in a natural way. In addition, the kinematics equations adopted in this research are used to transfer the Euler angles of the user's head from the Kinect coordinate frame to the reference coordinate frame in order to calculate all the joints angles of the human arm. Furthermore, we also conducted experiments and drew the curve based on the test data. The results shows that even if the NAO robot can imitate the human user's movements, there are still errors existing. Thus in the following sections, KF algorithm based sensor fusion will be employed to reduce the noise and RBFNNs will also be applied to overcome the unknown dynamics, which will increase the tracking performance of the teleoperation system.

5 Chapter Five: Teleoperation Control of the Baxter Robot using Kalman Filter and Neural Networks

5.1 Introduction

With the rapid advancements in sensor and actuator techniques, the use of robot technology has been incorporated into areas such as control system design, data sensor innovation, bionics and artificial intelligence. According to the research [117], following the accomplishment of modern robots, industrial robots currently attract considerable interest. HRI plays an increasingly important role in the industrial robot application. The robot is thought to have the capacity to adjust to the modern needs. Humans can adapt to various surrounding environments, hence, if robots demonstrate this adaptive ability through teleoperation then they would be able to carry out more complicated/challenging tasks. With robot teleoperation systems, an operator is able to remotely control a robot through the internet [118].

However, there are some potential challenges, such as ground-truth information missing, signal transmission delay due to the restrictions of transfer speed and data transmission protocol. Despite these issues, it is still a critical and helpful instrument in the field of HRI research. Hence, numerous research studies on teleoperation have been done in the works of literature. In [119], a mutual control strategy for the Baxter robot controller is developed, where a strategy for emulating human writings links to a Baxter robot controller. Researchers are able to teleoperate a robot by utilizing electromyography (EMG) signals and a haptic device. In [120], surface electromyography (sEMG) signals have been upgraded to perform teleoperation. The human operators can detect the situation in a haptic way and adjust muscle compression subliminally. In [121], the authors have improved the way for robot teleoperation by hand gesture recognition based on the visual data.

In addition, a direct approach has been developed for the robot to imitate human motion. This approach, coined motion capture technology, is an ideal strategy for skill transfer. To capture human motion, the human body itself has to be followed. There are various methods to achieve human motion capture. The most widely used method is to detect the markers from the body of a human operator. However, it is inconvenient for operation. Another method is to utilize image processing from typical cameras. However, this strategy is not reliable owing to the unstable body location capacities during the imaging process. Other methods include stereo-vision cameras that are applied to motion capture for depth data analysis. Unfortunately, it needs a long processing time, thus it is proved to be ineffective for real-time applications.

As one of the enabling techniques for teleoperation, motion capture primarily incorporates two interfaces, which are remotely wearable device input interface and detecting interface based on a vision system. Several sensors can be utilized for the visual system, such as Leap Motion and Kinect. In [7], human motions are obtained by using the Kinec-

t sensor, and through the utilization of the vector approach, the joint points of a Baxter robot can be ascertained. In [122], motions of welder-related work are captured by the Leap Motion sensor, which is estimated by a soldering robot with teleoperation networks. Moreover, the wearable devices, for example, exoskeleton [123] or joystick, or omnihaptic device [124] are normally used. In this thesis, we investigate a control scheme combining the wearable device MYO armband together with the motion capture system that uses a Kinect sensor, to teleoperate a Baxter robot. The performance of the proposed scheme will be optimized by KF based sensor fusion. In [125], a robust finite-horizon KF is designed for discrete time-varying uncertain systems with both the additive and the multiplicative noise. KF is widely used as it can estimate the past, current and future signal state, even if there are uncertainties in the model.

Filtering is a procedure of signal processing and transformation (removing or reducing undesirable components and enhancing the required characteristics). The above stated steps can be implemented either in software or hardware. KF is a software filtering method minimizing the mean square noise, which is one of the best estimation criterion. Furthermore, KF uses the state space model of signal and noise to estimate the value of the previous time, after that the observed values of the current time are updated. Based on the established equations and observation systems, the algorithm estimates the minimum mean square noise of the signal to be processed. There are methods that are able to approximate and compensate for the uncertainties of the robot dynamics, for example, Fuzzy Logic and NN. The NN control continues to be widely examined in the discrete-time control system [126], [127] as well as the continuous-time control system [128].

Some researchers proposed strategies using Kinect and MYO armbands to obtain the joint angles of human arms and then teleoperate with robots as in [7] and [110]. However, the problem raises that the accurate calculation of the angular data is unavailable owing to the measured noise. Thus, this research develops an optimum strategy to reduce the influence of the noise using KF based sensor fusion. The vector approach is used to calculate the five required joints angles, then the KF algorithm is employed to output a series of more accurate data. MYO armbands worn on the operator's lower and upper arm are utilized to identify and measure the angular velocity of human arm motion. A Kinect based body tracking system is also utilized. Furthermore, the control system is successfully connected to NN by UDP for the teaching and learning processes of the Baxter robot, which promotes the tracking performance.

5.2 Environmental Setup

5.2.1 System Configuration

To delineate the teleoperation of robots utilizing motion capture, an illustrative system is assembled. It consists of the body tracking system, Baxter robot and MYO armbands, as

shown in Fig. 5.1.

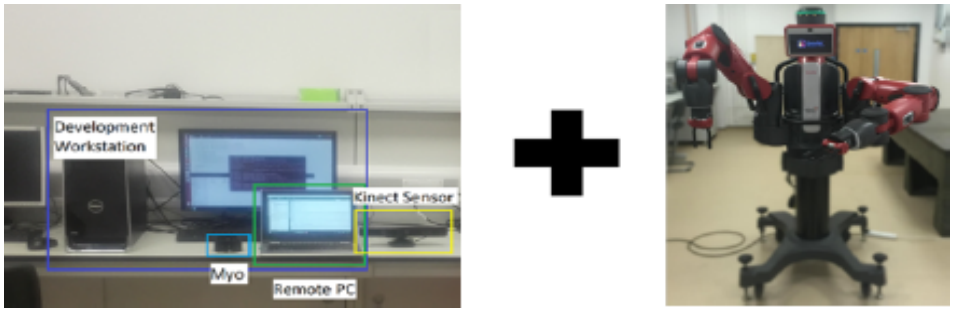


Figure 5.1: The complete experimental teleoperation system

Motion capture is conducted by the Kinect sensor. Kinect is utilized due to its low cost and ability to provide the data required for this research. The Kinect device is connected to a remote computer where processing programming software is used to get the position information from the Kinect sensor.

The experiment is performed on a Baxter robot. The overall experimental system was associated with and controlled by the development workstation, a remote computer with an Ethernet link, as well as a pair of MYO armbands. The principle of the teleoperation system is represented in Fig. 5.2.

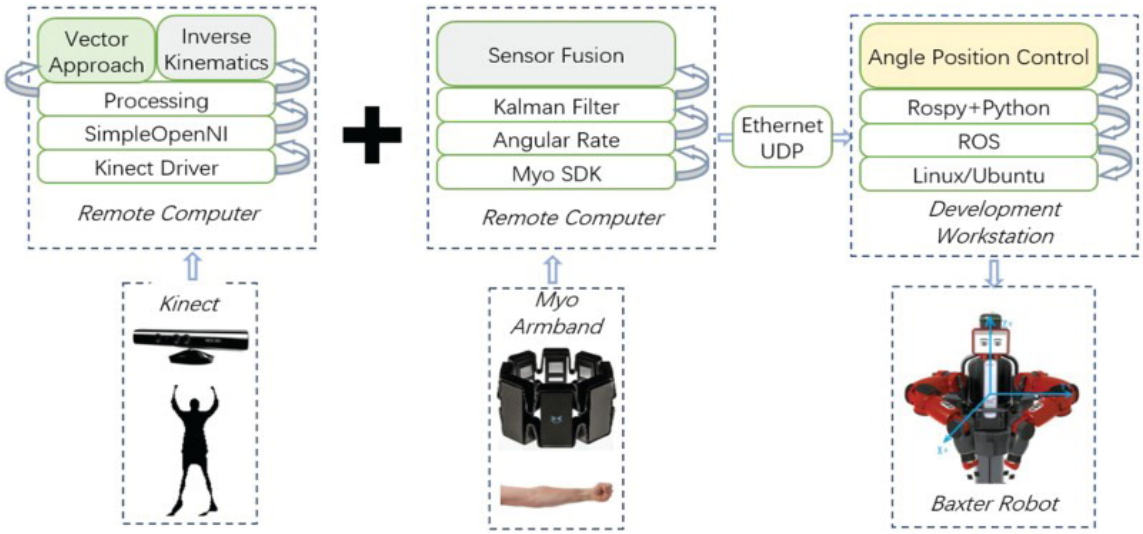


Figure 5.2: Diagram for the rule of research in total

5.2.2 Development Workstation

The first generation of the Kinect sensor, a part of the proposed teleoperation system, is an arrangement of sensors created as a fringe device with the Xbox video game device to track the human motion in 3D space. An RGB camera and a double infrared profundity sensor are located in the front of the Kinect sensor [129]. Utilizing image and depth sensors, Kinect V1 can distinguish the movement of an object/operator. It does not require the human user to wear any additional devices, as shown in Fig.5.3.

As can be seen in Fig.5.3, the depth detector of the Kinect V1 contains two units: the monochrome CMOS sensor and the infrared projector (label 1, Fig.5.3). They work together in the motion capture process. The image depth and RGB data collected via the Kinect sensor is illustrated in Fig. 5.4. The human body can be identified by arranging several straight lines together to show positions and poses in 3D space. Kinect collects data about human joint positions and velocities during teleoperation processes, and then sends them to the robot. In this way, human-robot cooperation is achieved. Compared to the traditional motion tracking device with complex programming, which has high cost and complex setup process, Kinect can be integrated into the control system working with open source codes [130].

Numerous programming projects on Kinect are accessible, including OpenKinect, Open-

Figure 5.3: Image of Kinect sensor: 1. Depth sensors, 2. RGB camera, 3. Motorized base [7]

Figure 5.4: Image of depth and RGB sensor data collected from Kinect V1 modified on [7]

NI, Microsoft Kinect for windows SDK [131]. OpenKinect is free and designed to equip the Kinect with computers and other devices in [132]. OpenNI is able to support a large amount of different devices apart from Kinect in [132], which utilizes NITE to get the skeleton information of the operator, according to [133]. Microsoft Kinect for windows SDK is another commonly used platform produced by Microsoft in [133]. SimpleOpenNI is an OpenNI and NITE wrapper for Processing in [134].

The utilization of programming software is imperative and relies on:
(1) capacity to separate skeletal information;
(2) similarity with different operating systems, for example, Windows and Linux;
(3) clear documentation;
(4) clear-cut and direct approach for quick confirmation of calculations.

After appropriate examination, a processing programming software which fulfils all the prerequisites is utilized. According to [135], operators can program via Kinect with SimpleOpenNI wrapper for OpenNI and NITE, and skeleton information can be collected on both Windows and Linux platforms. Processing is based on Java, hence fundamentally the same syntax can be used. All the functions utilized in this thesis are given below:

PVector: A class to depict a few dimensional vectors, particularly the Geometric vector. The magnitude and direction of each pixel can be obtained using the methods *mag ()* and *heading ()*, according to [136].

***pushMatrix()* and *popMatrix()*:** They can transfer the present transformation matrix into the matrix stack. The *pushMatrix()* can record the current coordinate system information to the stack and *popMatrix()* restores them. The *pushMatrix()* capacity and *popMatrix()* capacity are utilized in conjunction with other transformation functions and can be used for the extent of changes [136].

5.2.3 Robot Operating System and RosPy

Robot Operating System (ROS) is an adaptable operating system for programming robots. According to [137], it contains the tools, conventions and libraries needed for the complicated task in order to simplify robot activities. ROS can be set up under multiple platforms.

RosPy is a related Python customer library for ROS. The RosPy customer API empowers Python software engineers to rapidly interact with parameters, services and topics of ROS. RosPy requires usage speed (i.e. designer time) over runtime execution, hence calculations can be immediately prototyped and examined inside ROS. It is also benefitial for codes which have no critical path such as codes for initialization and configuration. A large amount of ROS instruments are composed in RosPy script to develop introspection abilities. A large number of ROS devices - Rostopic and Rosservice - are based on RosPy [99].

5.2.4 User Datagram Protocol

User Datagram Protocol (UDP) is one of the core members of the Internet protocol suite (the set of network protocols utilized for the Internet). Through this, a PC software can send information, using datagrams, to different PCs with an Internet Protocol (IP). With UDP, unique transmission channels or data paths can be established without prior communications. UDP is suitable for purposes 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. According to [138], UDP is frequently utilized for time-sensitive applications owing to the fact that dropping packets are preferred to waiting for delay packets, where it is difficult to find an alternative for the real time system.

5.3 Motion Capture by Kinect

5.3.1 General Calculation

The motion capture calculations for the upper limb depends on information including distances, locations and joint angles. The distance between two specified points in 2D and 3D space can be calculated by equation (29) and (30).

$$d_{2D} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{29}$$

$$d_{3D} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{30}$$

where (x_1, y_1) and (x_2, y_2) are points in 2D space, d_{2D} is the distance between these two points, (x_1, y_1, z_1) and (x_2, y_2, z_2) are points in 3D space, d_{3D} is the length between these two points.

The angles at all the joints are obtained by the law of cosines. The maximal angle is 180 degrees. While computing the angles among the joints, an extra point is required to define at 180~360 degrees. After capturing the motion, a triangle is drawn by utilizing any two joint points. From the other two points, the third point of the triangle can be obtained. In this scenario, the coordinated statistics for every point of the triangle is known, thus we are able to find out the length of every side, instead of the value of each angle, which is still unknown. As shown in Fig. 5.5, the magnitude of any coveted point can be calculated by applying the law of cosines. Computations for the joint points illustrate the length of sides a, b, c. Similarly, we can also calculate each angle of the triangle using the law of cosines.

5.3.2 Vector Approach

The Kinect sensor can identify every single joint of the human body and supply feedback concurrently. All these directions are transformed into vectors and the particular angles

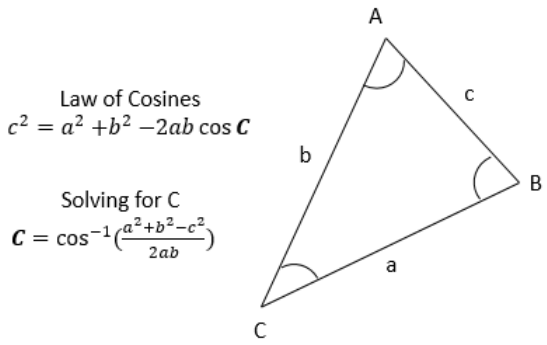


Figure 5.5: Mathematical Principle description

of the joints. The coordinates of the human body joints collected via Kinect are under the Cartesian space. The particular angles from arms are computed. After the mapping process by the Kinect sensor is completed, the data are sent to teleoperate the Baxter.

The five points *ShoulderPitch*, *ShoulderYaw* and *ShoulderRoll* as well as *ElbowPitch* and *ElbowRoll*, shown in Fig. 5.6, are computed from the arm positions data extracted from the Kinect.

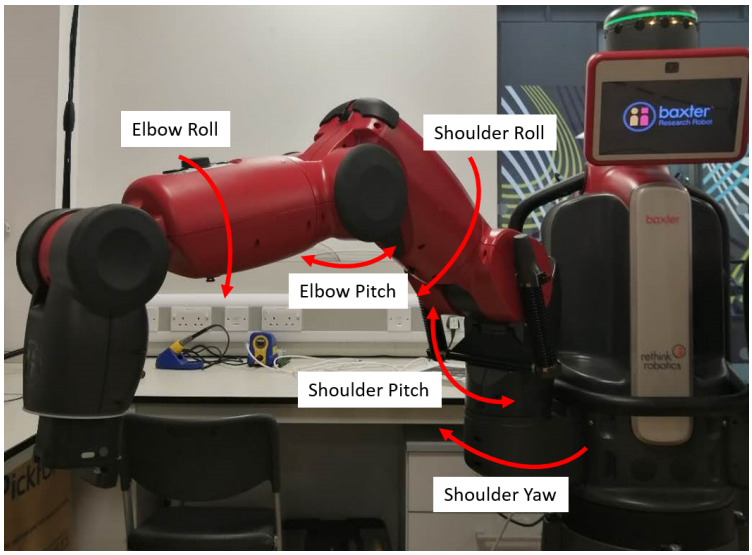


Figure 5.6: Demonstration of all related angles in Vector Approach: *ShoulderPitch*, *ShoulderYaw* and *ShoulderRoll*, *ElbowPitch* and *ElbowRoll*

The computation of vectors is illustrated in Fig. 5.7. According to [7], the intense lines CO and CD represent the upper left and ahead part of the arm. The line BO represents the

distance from the left hip to the left shoulder, and AO represents the length between the right and left shoulder. Directions with coordinated data B_{X+}, B_{Y+} and B_{Z+} shows the axis system of the Kinect sensor in Cartesian space, where point B is the origin.

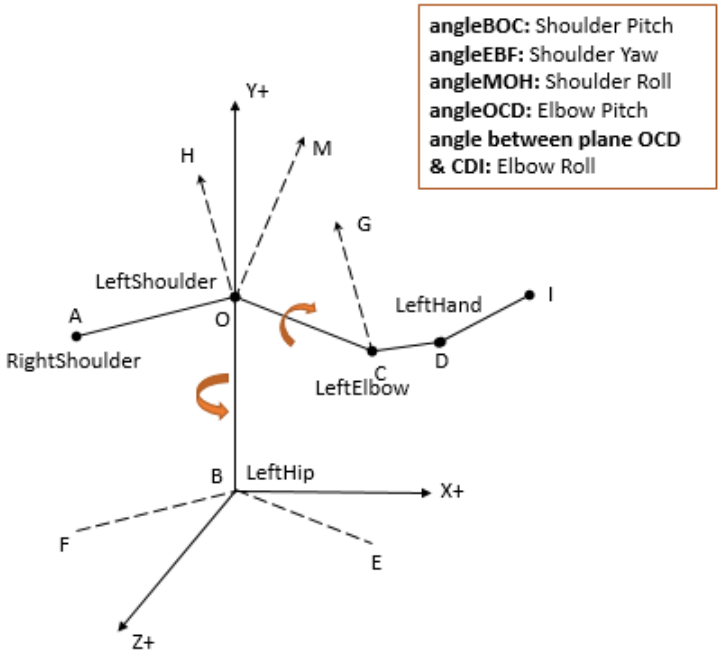


Figure 5.7: The principle of vector approach in mathematical computing

Methodology for Computing *ShoulderPitch* and *ElbowPitch*: As illustrated in Fig. 5.7, according to [7], the angle $\angle BOC$ (*ShoulderPitch*) is obtained by the distance of two points from vectors $\overline{OB}$ to $\overline{OC}$. The pitch angles of the shoulder and elbow are calculated utilizing the three specified joints' position, which are shoulder (point O), elbow (point C) and hip (point B). Delivering these three points using the angle *Of()* function gives feedback on the value for angles, which are directly sent to Baxter. The $\angle OCD$ (*ElbowPitch*), which is the angle among $\overline{CD}$ and $\overline{OC}$, can be computed through sending hand, elbow and shoulder values into the angle *Of()* for working as well [7]. In this methodology, we can use the angle *Of()* command in the Processing software to calculate any angles between two vectors.

Methodology for Computing *ShoulderYaw*: As we can see from Fig. 5.7, according to [7], the angle $\angle EBF$ (*ShoulderYaw*) is obtained by a similar method as utilizing both the shoulder point and elbow point, which are point A, O and C, respectively, where the vectors $\overline{OC}$ and $\overline{OA}$ are grouped together. However, the above mentioned vectors $\overline{OC}$ and $\overline{OA}$ need to be anticipated into the plane XZ. In this way, we are able to obtain the vectors

$\overline{BF}$ and $\overline{BE}$. Angle $\angle EBF$ (*ShoulderYaw*) is the value of the angle between $\overline{BF}$ and $\overline{BE}$, which can be computed by utilizing angle *Of()* command in Processing.

Methodology for Computing *ElbowRoll* and *ShoulderRoll*: The *ElbowRoll* is the angle between plane OCD and CDI, which can be calculated by the angle *Of()* functions. The *ShoulderRoll* is the most difficult to calculate due to the fact that the computing is not straightforward and all the points are given in a 3D plane, the computing method utilized above will not work here [7].

To calculate the required angles, the vector points from elbow to hand, where its plane is opposite to the one from shoulder to elbow (through the shoulder joint too), are noted. This reference vector must be stable in relation to the rest of the body. Thereafter, the reference vector can be computed by intersecting the vector points from shoulder to shoulder, and shoulder to elbow.

According to [7], the normal line from that intersecting point of two vectors is used for continuous calculation. The vector $\overline{OM}$ can be obtained by verifying the intersecting point between vectors $\overline{OC}$ and $\overline{OA}$. The vector $\overline{OM}$ is vertical to the plane expanded by the vectors $\overline{OA}$ and $\overline{OC}$. Clearly, vector $\overline{OM}$ and vector $\overline{OC}$ are vertical from each other.

The normal line vector $\overline{CG}$ can be decided via intersecting vector points $\overline{CD}$ and $\overline{OC}$, which is additionally vertical to vector $\overline{OC}$. At this point, we can obtain the vector $\overline{OH}$ by deciphering vector $\overline{CG}$ along vector $\overline{CO}$ to point O. The angle $\angle MOH$ between vectors $\overline{OH}$ and $\overline{OM}$ is defined as the *ShoulderRoll* [7].

The orientation angles sent by the Kinect can be separated by utilizing *PMatrix3D* with Processing software. The *PMatrix3D* outputs the required rotation matrix as well, where the current coordination framework is well backup into the stack. It is then delivered to the shoulder joint, additionally, the rotation matrix is utilized to transform into the coordination system data. Every single computation in this capacity will be decided within the obtained coordination framework.

After the computation of *ShoulderRoll* and *ElbowRoll* angles, the rotation matrix from the stack can be recovered to obtain the initial coordination framework. The right *ShoulderRoll* is additionally computed with the similar method. Furthermore, a small change has been applied to the vectors coordination system.

Due to the inaccuracy of the function used to calculate roll angles, the error needs to be corrected; every value change of the *ShoulderRoll* keeps along with those of the *ShoulderYaw*. After that, the statistics are plotted into the MATLAB, as seen in Fig. 5.8. After several trials, the error is mostly revised by the following equations:

$$\gamma_s = -\gamma_s - \beta_s/2 - 0.6 \tag{31}$$

where we define the angle of left *ShoulderRoll* as γ_s and the angle of left *ShoulderYaw* as β_s. These returned data are sent to Baxter development platform for further work utilizing UDP protocol. The data packets created by the server are sent through the function introduced above. So far, every single angular value is sent to teleoperate the Baxter robot with

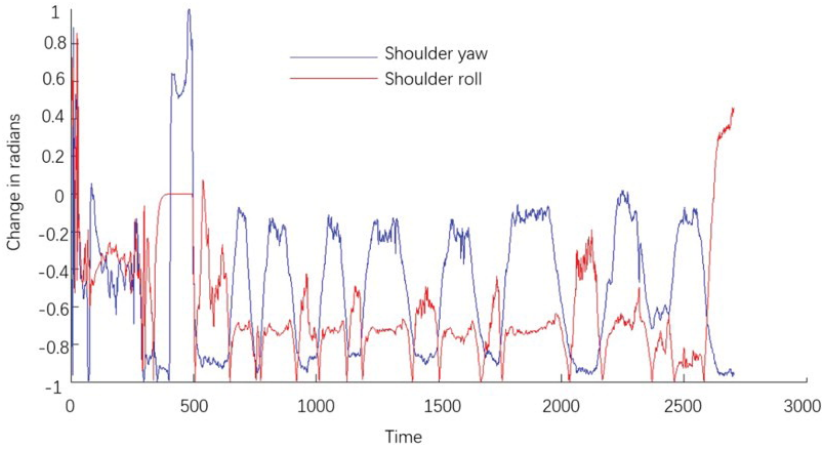


Figure 5.8: Error of Vector Approach

the Python script.

5.4 Measurement of Angular Velocity by MYO Armband

The joint angles are obtained by integrating angle velocity. Any position of the human operator's arms can set to be the initial position, if the joint angles are assumed to be zero, according to [110]. When the operator moves his arm to a new pose P, the rotation angles (joint angles) are pose P with respect to the initial pose in [110].

As shown in Fig. 5.9, the frame (X_1, Y_1, Z_1) represents the orientation of the MYO armband in the initial position. The frame (X_2, Y_2, Z_2) represents the current orientation of the MYO. From the first MYO armband worn on the upper arm, we can obtain three angular velocity $v1_x$, $v1_y$, $v1_z$, which represent *ShoulderRoll*, Pitch and Yaw, respectively. From the second MYO armband worn on the forearm, we can obtain the angular velocity

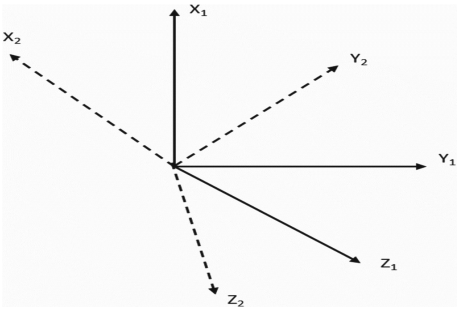


Figure 5.9: The orientation of the MYO in the initial pose and the current pose

$v2_x$, $v2_y$, which represent *ElbowRoll* and *ElbowPitch*.

There would be errors in the joint angular velocity measured by the joint angle. Moreover, the errors will be superimposed and coupled. The shoulder joint error will be superimposed on the elbow joint, resulting in a greater elbow error. Errors will also accumulate as time evolutes. Although the sampling frequency IMU in the MYO is 50 Hz, the resulting angle will have a large difference in value when the joint angle is calculated from the angular velocity integral in the program [110]. Here, the method for the angular velocity is extended from the method of measurement for the angles using MYO armbands mentioned in previous research [110]. In summary, MYO armband is used to measure the angular velocity of each joint, and the Kinect is used to get the angles of each joint.

5.5 Kalman Filtering based Sensor Fusion

The KF method has two basic assumptions:

(1) the sufficiently accurate model of the information process is a linear (or time-varying) dynamic system excited by white noise; (2) the measurement signal contains additional white noise components for each session.

Only if the above assumptions are satisfied, can a KF method be applied. In this thesis, all the data collected from the Kinect sensor and MYO armbands fulfilled the above stated requirements, hence the use of continuous-time KF to fuse the data from different sensors were incorporated.

The equation (mentioned in the previous section) is for the continuous-time KF, wherein, $x(0)$, ω and v are not related to each other, according to [108] and [109], the expected values of the continuous-time KF satisfy the following statistical characteristics,

$$\begin{aligned}
&E\{x(0)\} = m_x(0); E\{\omega(t)\} = 0; E\{v(t)\} = 0 \\
&E\{\,[x(0) - m_x(0)]\,[x(0) - m_x(0)]^T\,\} = P(0) \\
&E\left\{x(0)\omega^T(t)\right\} = 0; E\left\{x(0)v^T(t)\right\} = 0 \\
&E\left\{\omega(t)v^T(t)\right\} = 0; E\left\{\omega(t)\omega^T(\tau_{KF})\right\} = s(t)\delta(t - \tau_{KF}) \\
&E\left\{v(t)v^T(\tau_{KF})\right\} = r(t)\delta(t - \tau_{KF})
\end{aligned} \tag{32}$$

where s is the system noise variance intensity matrix of continuous system; r is the array of measured noise variance intensity; $m_x(0)$ and $P(0)$ are the initial mean value of x and the initial covariance matrix, respectively; $\delta(t - \tau_{KF})$ is the Dirac δ function.

We assume that every single joint in the human arms is taken into account separately for research, which shows that all the KF factors are the first order, hence, here A=1, B=1, G=1 and H=1. Then the KF equations are simplified as below,

$$\begin{aligned}
\dot{x}_i &= x_i + u_i + \omega \\
y_i &= x_i + v
\end{aligned} \tag{33}$$

where in this special case, y_i is the angular position of the number of i joint collected from the Kinect sensor. And u_i is the angular velocity of the number of i joint of the operator's arm motion.

$$\begin{aligned} \dot{\hat{x}}_i &= \hat{x}_i + u_i + k\,(y_i - \hat{x}_i) \\ k &= pr^{-1} \\ \dot{p} &= p - pr^{-1}p + s \end{aligned} \tag{34}$$

where k is the filter gain matrix, p is the estimated covariance matrix, $\hat{x}_i$ is the required (satisfied) data obtained from KF based sensor fusion, which is also the statistic that needs to be sent to the development workstation via UDP.

5.6 Neural Networks Based Control System

The notation of the data output from the result of KF based sensor fusion q_{di}, in addition to the desired joint space trajectory q_d is defined

$$\begin{aligned} q_{di} &= \hat{x}_i \\ q_d &= [q_{d1}, q_{d2}, q_{d3}, q_{d4}, q_{d5}]^T \in R^5. \end{aligned} \tag{35}$$

Here, we apply the NN control procedure to achieve the accurate tracking of the reference joint trajectory. Hence, the controller should be quick, steady and accurate. The reference joint trajectory $q_d \in R^5$ consists of a time series of the angle values that are generated from the motion data filtered by the KF. The angular matrix $q \in R^5$ denotes the actual joint positions during the teleoperation. At that point, as indicated by [110], the dynamics of the robot are given as shown in (36).

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \tau_{ext} \tag{36}$$

where $M(q) \in R^{5\times5}$ is the inertia matrix, $C(q, \dot{q}) \in R^{5\times5}$ is the Coriolis matrix, $G(q) \in R^{5\times1}$ are the gravity terms and τ_{ext} is unmodeled elements owing to the external disturbance and system uncertainties.

Define $z = \dot{e}_q + \Lambda e_q$, $q_r = \dot{q}_d - \Lambda e_q$, where $e_q = q - q_d$, $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)$, λ_n is a positive constant. Here the dynamic equation (36) can be rewritten as (37) from [110].

$$\tau - \tau_{ext} = M(q)\dot{z} + C(q, \dot{q})z + G(q) + M(q)\dot{q}_r + C(q, \dot{q})q_r \tag{37}$$

In [110], an adaptive controller is designed as (38),

$$\tau = \hat{H}(q) + \hat{M}(q)\dot{q}_r + \hat{C}(q, \dot{q})q_r - Kz \tag{38}$$

where $K = diag(k_1, k_2 ... k_i), k_i > \frac{1}{2}$, $\hat{H}(q)$, $\hat{M}(q)$ and $\hat{C}(q, \dot{q})$ are the RBFNN estimates of $G(q) + \tau_{ext}$, $M(q)$ and $C(q, \dot{q})$, respectively.

Then the closed circling framework dynamic equation is given below as (39) [110].

$$M\dot{z} + Cz + Kz = (\hat{M} - M)\dot{q}_r + (\hat{C} - C)q_r + (\hat{H} - H) \tag{39}$$

The RBFNN based approximation approach is applied as follows [110].

$$\begin{aligned} M(q) &= W_M^{*\mathrm{T}} S_M(q) + \epsilon_M \\ C(q,\dot{q}) &= W_C^{*\mathrm{T}} S_C(q,\dot{q}) + \epsilon_C \\ H(q) &= W_H^{*\mathrm{T}} S_H(q) + \epsilon_H \end{aligned} \tag{40}$$

where W_M^*, W_C^* and W_H^* are the constant ideal weight matrix; $S_M(q)$, $S_C(q,\dot{q})$ and $S_H(q)$ are the basis function matrix, and ϵ_M, ϵ_C and ϵ_H are the mismatch uncertainties due to the fact that the number of the hidden neuron is limited.

Then we can write the equation for basis function matrix as follow [110],

$$\begin{aligned} S_M(q) &= \mathrm{diag}\left(S_q, \ldots, S_q\right) \\ S_C(q,\dot{q}) &= \mathrm{diag}\left(\begin{bmatrix} S_q \\ S_{\dot{q}} \end{bmatrix}, \ldots, \begin{bmatrix} S_q \\ S_{\dot{q}} \end{bmatrix}\right) \\ S_H(q) &= \left[S_q^{\mathrm{T}} \ \ldots \ S_q^{\mathrm{T}}\right]^{\mathrm{T}} \end{aligned} \tag{41}$$

where

$$\begin{aligned} S_q &= \left[\phi(\|q - q_1\|) \ \phi(\|q - q_2\|) \ \ldots \ \phi(\|q - q_n\|)\right]^{\mathrm{T}} \\ S_{\dot{q}} &= \left[\phi(\|\dot{q} - \dot{q}_1\|) \ \phi(\|\dot{q} - \dot{q}_2\|) \ \ldots \ \phi(\|\dot{q} - \dot{q}_n\|)\right]^{\mathrm{T}} \end{aligned} \tag{42}$$

The estimates of $M(q)$, $C(q,\dot{q})$ and $H(q)$ can be obtained as (43).

$$\begin{aligned} \hat{M}(q) &= \hat{W}_M^{\mathrm{T}} S_M(q) \\ \hat{C}(q,\dot{q}) &= \hat{W}_C^{\mathrm{T}} S_C(q,\dot{q}) \\ \hat{H}(q) &= \hat{W}_H^{\mathrm{T}} S_H(q) \end{aligned} \tag{43}$$

Now, substituting (43) into (39), then the previous equations are simplified as

$$\begin{aligned} M\dot{z} + Cz + Kz = \tilde{W}_M^{\mathrm{T}} S_M(q)\dot{q}_r \\ + \tilde{W}_C^{\mathrm{T}} S_C(q,\dot{q})q_r + \tilde{W}_H^{\mathrm{T}} S_H(q) \end{aligned} \tag{44}$$

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 $\tilde{W}_H^{\mathrm{T}} = \hat{W}_H^{\mathrm{T}} - W_H^{*\mathrm{T}}$.

For a matrix of $n \times n$, the sum of the elements on the main diagonal of the matrix (the diagonal from the upper left to the lower right) is called the trace of matrix A, denoted $tr(A)$; if A, B are $m \times n$ and $n \times m$ matrices, then $tr(AB) = tr(BA)$ [139]. Here, the Lyapunov candidate function is chosen as,

$$\begin{aligned} V =& \frac{1}{2} z^{\mathrm{T}} M z \\ &+ \frac{1}{2} tr\left(\tilde{W}_M^{\mathrm{T}} Q_M \tilde{W}_M + \tilde{W}_C^{\mathrm{T}} Q_C \tilde{W}_C + \tilde{W}_H^{\mathrm{T}} Q_H \tilde{W}_H\right), \end{aligned} \tag{45}$$

where Q_M, Q_C and Q_H are positive fixed weight matrix. Then, the derivative of V, which shows

$$\dot{V} = z^{\mathrm{T}} M\dot{z} + \frac{1}{2} z^{\mathrm{T}} \dot{M} z + tr\left(\tilde{W}_M^{\mathrm{T}} Q_M \dot{\tilde{W}}_M + \tilde{W}_C^{\mathrm{T}} Q_C \dot{\tilde{W}}_C + \tilde{W}_H^{\mathrm{T}} Q_H \dot{\tilde{W}}_H\right) \tag{46}$$

Utilizing the property: $\dot{M}(q)$-$2C(q,\dot{q})$ are the skew symmetric matrix, and equation (46) becomes:

$$\dot{V} = z^{\mathrm{T}} M\dot{z} + z^{\mathrm{T}} Cz + tr\left(\tilde{W}_M^{\mathrm{T}} Q_M \dot{\tilde{W}}_M + \tilde{W}_C^{\mathrm{T}} Q_C \dot{\tilde{W}}_C + \tilde{W}_H^{\mathrm{T}} Q_H \dot{\tilde{W}}_H\right) \tag{47}$$

The ideal weight matrix W_M, W_C and W_H are constant matrices, we have the following relationship

$$\begin{aligned} \dot{\tilde{W}}_M &= \dot{\hat{W}}_M \\ \dot{\tilde{W}}_C &= \dot{\hat{W}}_C \\ \dot{\tilde{W}}_H &= \dot{\hat{W}}_H, \end{aligned} \tag{48}$$

substituting (44) into the equation (47), then we have:

$$\begin{aligned} \dot{V} = &- z^{\mathrm{T}} Kz - \\ &tr\left[\tilde{W}_M^{\mathrm{T}}\left(S_M(q)\dot{q}_r z^{\mathrm{T}} + Q_M \dot{\hat{W}}_M\right)\right] + \\ &tr\left[\tilde{W}_C^{\mathrm{T}}\left(S_C(q,\dot{q})q_r z^{\mathrm{T}} + Q_C \dot{\hat{W}}_C\right)\right] + \\ &tr\left[\tilde{W}_H^{\mathrm{T}}\left(S_H(q) z^{\mathrm{T}} + Q_H \dot{\hat{W}}_H\right)\right] \end{aligned} \tag{49}$$

According to [110], the upgraded principle is given as follow,

$$\begin{aligned} \dot{\hat{W}}_M &= -Q_M^{-1}(S_M(q)\dot{q}_r z^{\mathrm{T}} + \sigma_M \hat{W}_M) \\ \dot{\hat{W}}_C &= -Q_C^{-1}(S_C(q,\dot{q})q_r z^{\mathrm{T}} + \sigma_C \hat{W}_C) \\ \dot{\hat{W}}_H &= -Q_H^{-1}(S_H(q) z^{\mathrm{T}} + \sigma_H \hat{W}_H) \end{aligned} \tag{50}$$

where $\sigma_M, \sigma_C, \sigma_H$ are pre-designed positive constants.

Combining (50) and (49), then we obtain

$$\begin{aligned} \dot{V} = &- z^{\mathrm{T}} Kz - \sigma_M tr\left(\tilde{W}_M^{\mathrm{T}} \hat{W}_M\right) - \sigma_C tr\left(\tilde{W}_C^{\mathrm{T}} \hat{W}_C\right) \\ &- \sigma_H tr\left(\tilde{W}_H^{\mathrm{T}} \hat{W}_H\right) \end{aligned} \tag{51}$$

Applying Young's inequality, (51) can be extended to

$$\begin{aligned} \dot{V} = &- z^{\mathrm{T}} Kz + \frac{\sigma_M tr(W_M^{*\mathrm{T}} W_M^*)}{2} - \frac{\sigma_M tr(\tilde{W}_M^{\mathrm{T}} \tilde{W}_M)}{2} \\ &+ \frac{\sigma_C tr(W_C^{*\mathrm{T}} W_C^*)}{2} - \frac{\sigma_C tr(\tilde{W}_C^{\mathrm{T}} \tilde{W}_C)}{2} \\ &+ \frac{\sigma_H tr(W_H^{*\mathrm{T}} W_H^*)}{2} - \frac{\sigma_H tr(\tilde{W}_H^{\mathrm{T}} \tilde{W}_H)}{2} \end{aligned} \tag{52}$$

Finally we obtain

$$\dot{V} \leq -\eta V + \kappa \tag{53}$$

where $\eta = \min[2K, \sigma_M/(\lambda_{\max}(Q_M)), \sigma_C/(\lambda_{\max}(Q_C)), \sigma_H/(\lambda_{\max}(Q_H))]$, $\kappa = \frac{1}{2}\mathrm{tr}(\sigma_M W_M^{*\mathrm{T}} W_M^* + \sigma_C W_C^{*\mathrm{T}} W_C^* + \sigma_G W_G^{*\mathrm{T}} W_G^*)$. Since $V > 0$ and κ is the result of the pre-designed constants and weight matrix that we give, the length of $\kappa \leq \eta$, we can have $\dot{V} \leq 0$. As indicated by Lyapunov stability theorem, the closed-loop stability is guaranteed.

5.7 Experimental Studies

The experiment was conducted in a sufficiently illuminated indoor environment with one operator standing two metres in front of the Kinect sensor. As in the previous section, we chose to test the *ShoulderPitch*, *ShoulderYaw*, *ShoulderRoll*, *ElbowPitch* and *ElbowRoll*. After the collation of data, a Baxter robot was simulated in MATLAB and teleoperated by operators.

5.7.1 The KF Based Limb Following Experiment

An operator wearing a pair of MYO armbands faced the Kinect sensor (as seen in Fig. 5.10) doing different arm movements. The operator wore one MYO armband near the centre of the upper arm and the other near the centre of the forearm. The former measured the orientation and angular velocity of the *ShoulderPitch*, *ShoulderYaw* and *ShoulderRoll*. The latter predicted the orientation and the angular velocity of the *ElbowPitch* and *ElbowRoll*. Before the experiments, the MYO armband was calibrated and EMG sensors were warmed up so the MYO armbands could better recognize different hand postures. During the experiment, the operator made no movements except with his arms at a reduced and stable velocity.

5.7.2 The NN Learning Based Limb Following Experiment

In the second experiment, a test experiment was set up to examine the accuracy of the designed control system as shown in Fig.5.11, using NN learning. A heavy object was attached to the end-effector of the Baxter. Meanwhile, the operator, wearing calibrated MYO armbands, stood in front of the Kinect sensor and remotely controlled the end-effector's position.

5.7.3 Experimental Results

Fig. 5.12 illustrates the experimental results of the five selected DOFs with their different trajectories under the KF-based sensor fusion between Baxter and the operator. The Kinect sensor provided the position difference between the robot's real trajectory and the

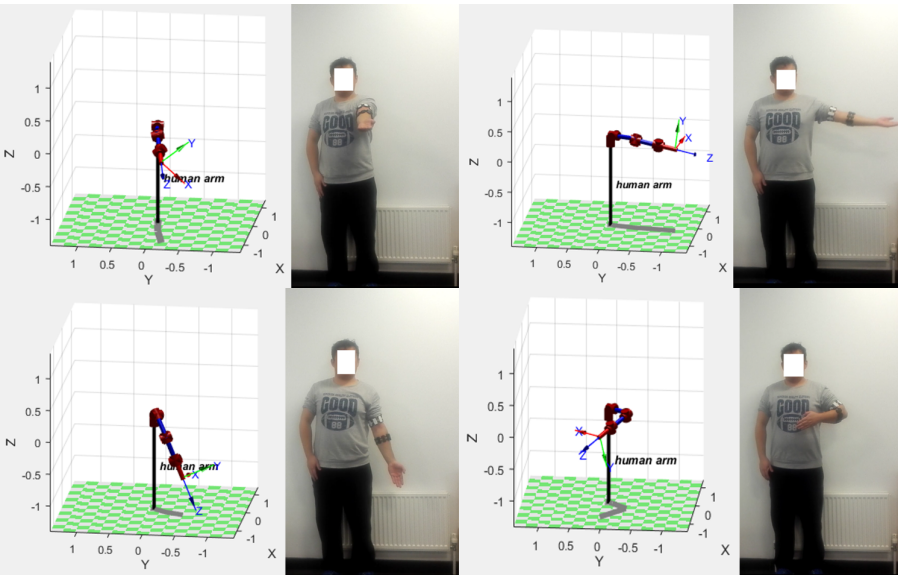


Figure 5.10: Demonstration of the experiment at the different positions

Figure 5.11: Image of designed control system

base point. The MYO armbands provided the angular velocity of those five angles accordingly before they were fused together via KF. The experimental data of the operator's arm motion from Kinect and MYO, and the optimum output from KF based sensor fusion and the angular statistics of the simulated Baxter robot were recorded, respectively, for the test. From the comparison results shown in Fig. 5.12, conclusion is drawn that the total performance of the motion capture system is improved with the KF based sensor fusion.

The use of a Kinect and MYO armband after KF based sensor fusion to teleoperate a Baxter robot was developed and verified. The results shown in Tab.5.1 demonstrate a series of ratios, which are the different values of the 5 angles between those obtained by KF and those directly collected by Kinect. Because the values filtered by KF are optimal, and the noise during the teleoperation process is suppressed. Hence, that ratio is defined as the efficient improvement, which is denoted as r_e as formulated in (54). The ratios

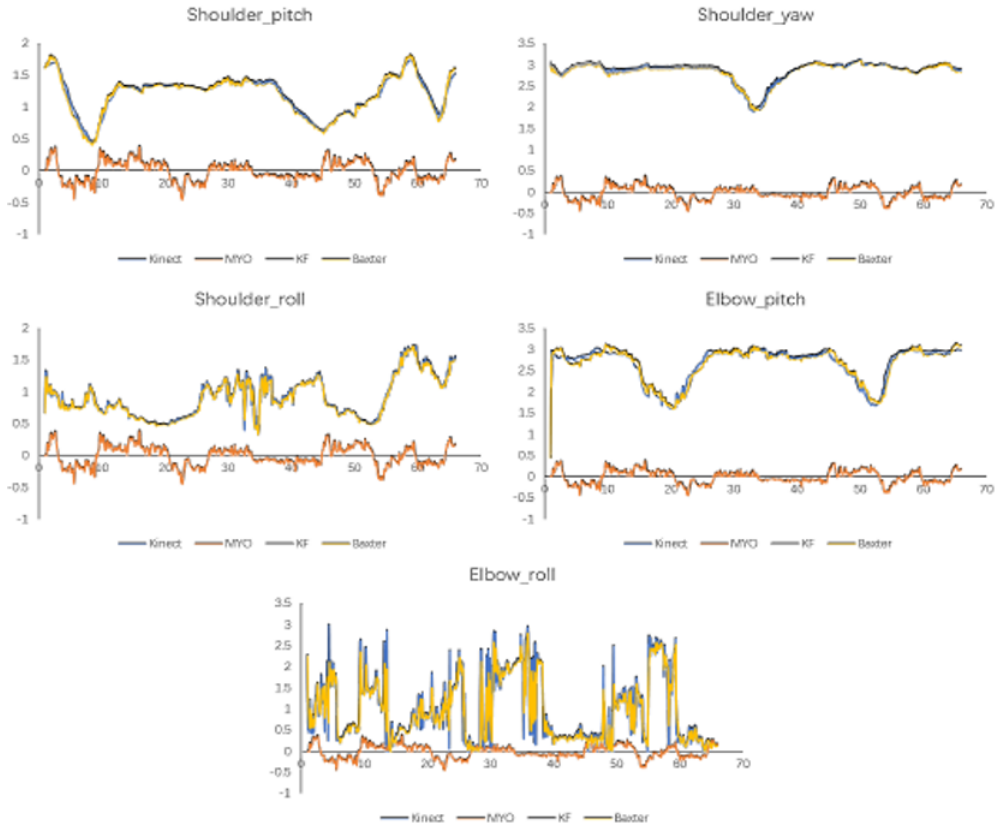


Figure 5.12: Graphical result after the KF based sensor fusion (*ShoulderPitch*, *ShoulderRoll*, *ShoulderYaw*, *ElbowPitch*, *ElbowRoll*)

shown in Tab.5.1 are averagely at 3.158%, 4.086%, 3.442%, 3.269% and 3.673%, standing for the angular positions of *ShoulderPitch*, *ShoulderRoll*, *ShoulderYaw*, *ElbowPitch* and *ElbowRoll*, respectively.

Table 5.1: Table for efficiency improvement of different angular positions

Data	*ShoulderPitch*	*ShoulderRoll*	*ShoulderYaw*	*ElbowPitch*	*ElbowRoll*
Ratio	0.67%	0.1021%	0.348%	0.29%	0.3025%

$$r_e = \frac{p_{KF} - p_{Kinect}}{p_{Kinect}} \tag{54}$$

where p_{KF} and p_{Kinect} are the experimental data of different angular positions obtained from the KF based sensor fusion and directly collected from the Kinect, respectively.

Fig.5.13(c) shows the time series of updating joint parameters in the function approximation method. It can be seen that the adaptive parameters of the trained five joints

converge boundedly with time. Fig.5.13(a) and Fig.5.13(b) give the joint compensation torque curves. The output torque of the NN is mainly to compensate for the uncertainties in the system to achieve the desired control performance. It shows that the uncertainty of the control system appears at a certain moment after the application of the neural network, and the control system can be well compensated by the input torque. The NN learning weighs can be seen in Fig.5.13(d).

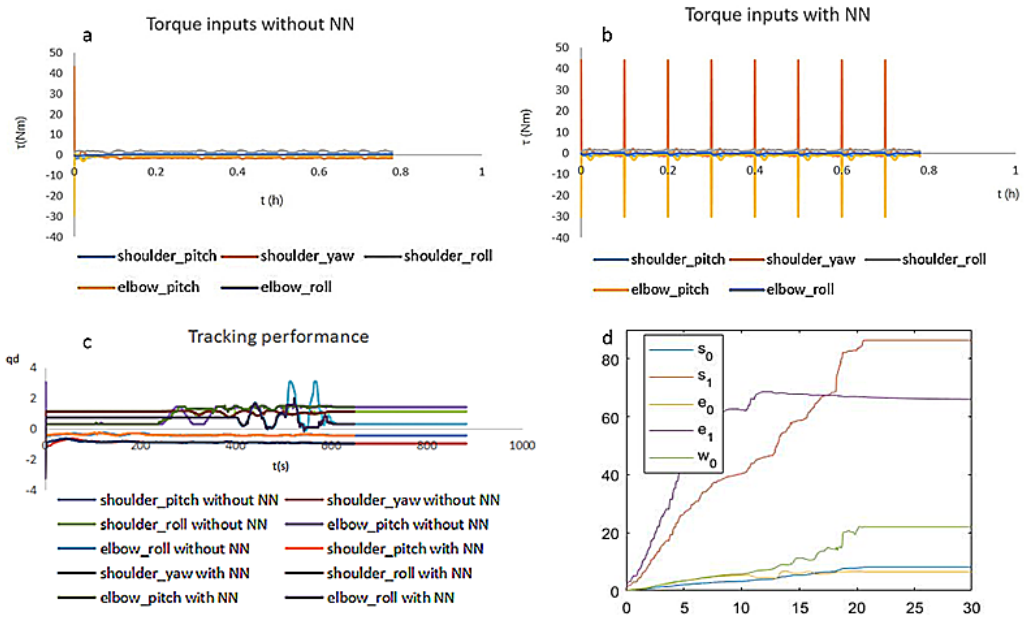


Figure 5.13: Torque inputs of the control system with NN. (a) Torque inputs of the control system without NN. (b) Torque inputs of the control system with NN. (c) Tracking performance of the designed system for both without NN and with NN. (d) NN learning weighs of every single joint.

It can be deduced from the above results that NN learning is a good choice as it has an acceptable error margin during the teaching and learning process, thereby satisfying the proposed design goal.

5.8 Conclusion

Following the last chapter, a KF based sensor fusion has been used to improve the tracking performance. This involved using a Kinect sensor to capture the motion of an operator's arm using the vector approach. The vector approach can accurately calculate the angular data of human arm joints, by selecting 5 out of 7 joints on each arm. In addition, the angular velocity of human operator's arm is measured by a pair of MYO armbands worn on the operator's arm. The data collected is thereafter sent to the Baxter robot for tele-operation. The use of the MYO in this research is due to its portability and capacity for

accurate computation of the values for the angular velocity of the shoulder and elbow motion. It works well with the Kinect sensor, and the application of KF based sensor fusion has greatly enhanced user-accuracy during teleoperation processes.

However, after usage of KF based sensor fusion, the robot has not completely done the motion along with the desired trajectory q_d owing to the existence of dynamic uncertainties. Hence a second experiment was conducted to compare the effect of the unknown payload under NN learning with that without NN learning. The results illustrate that the NN-learning applied control system achieves Baxter teaching with the smallest number of errors.

6 Chapter Six: An Enhanced Teaching Interface for a Robot using DMP and GMR

6.1 Introduction

Robot technology is developing fast and its effective applicability for use in traditional manual tasks has increased. The domains in which use have been found for robots continue to increase exponentially which prompt a need for robots with the ability to move as smoothly as a human [140]. In recent years, with the integration of robots into human daily life, the need for robots to acquire human skills, and teaching by demonstration are the underlying purpose of many research works in both academic and non-academic fields [17]. In [121], a slave-master controller based TbD is developed to retain the kinematic constraint by using the tracking system. A trajectory learning approach for multi-robot interaction for welding tasks is developed in [141]. The ability to transfer human skills, via demonstration, to robots has a huge influence on building robot intelligence, and is an important way to promote robot learning. It helps to successfully overcome the need to manually control the robot to replicate complex human movements.

During the teaching process, the human operator transfers motor skills to the imitator (robot) through recordings of the motions being taught and generalized [142]. In [143], a research team develops a problem-based learning (PBL) method on an autonomous vacuum robot with mechatronics systems. An actuated dynamics technology based on proportion-integral-derivative (PID) controller is proposed in [144], where the unactuated dynamics have been shown to be globally bounded. The acquisition of teaching data is realized through body sensations which convey human-robot dialogue effectively. This has a wide range of usage in intelligent identification and control systems as a natural way for human-robot interaction. Common somatosensory devices including wear systems, such as 3DSuit, data gloves, and optical movement capture systems such as Microsoft's Kinect somatosensory camera, have been widely employed to the HRI research studies.

The GMM is a commonly used clustering algorithm. It approximates the complex distribution. During the process, it first extracts a feature element of each unit in the sample, then uses the GMM to cluster these features as an object, finally obtaining the segmentation result. In [145], authors use continuous myoelectric signals to employ GMMs for multiple limb motion classification. Early in the 19^{th} century, scientists use a finite GMM whose parameters are estimated through the EM algorithm by estimating a probability density function of human skin color [146]. The playback process of a robot includes movement control and movement trajectory reproduction. Trajectory reproduction is employed when the encoded data are sent to regression techniques, such as GMR. Movement control is the result of mapping robot motion trajectories. It is also the playback process obtained from a demonstrator after the learning process. In addition, human-robot skill transfer can be achieved optimally by programming with special characteristic methods,

such as DMP [104].

With the objective of designing robots that can complete operational tasks, such as grabbing objects in dynamic and complicated environments, robots require the ability to overcome and avoid obstacles. For this purpose, the DMP method is used to model and generalize the motion trajectory inside the environment with obstacles via combining the specific planning algorithm with its generalization and stability. In [147], the DMP model combined with the haptic based kinesthetic method is used to comprehend the trajectory generation for the spherical impediment. In [148], the authors argue that the DMP can be represented by defining a function approximator. The method calls for unique obstacle records to calculate the repulsive pressure within the vicinity of the impediment. A GMM is broadly used in pattern recognition and facts evaluation, whereas GMR is a widely-used quantitative analysis approach. The GMM can clean the probability distribution of arbitrary shapes. In [106] and [149], the authors develop a proposed GMM to represent the nonlinear term, which is not required to manually specify the parameters of the basic functions. However, most probability estimation methods are often not able to attain the complete information. For example, though the sample would be known, the Gaussian distribution is unknown. Hence, it is necessary to employ the Expectation - Maximization algorithm (EM) solution.

Machine vision based detection and tracking have attracted wide attention. In [14], a research team develops a method, wherein a Bayesian-based object tracking system with the special focus on Microsoft Kinect devices intelligently schedules a network of multiple RGBD sensors. A kinematics based skeletal tracking system using a Microsoft Kinect sensor with an upper limb virtual reality rehabilitation system is investigated [19]. In [18], a task tracking the fingertips and palm centre are developed using a Kinect sensor. However, due to the unavoidable influence of camera resolution, lens distortion, the illumination intensity of the surrounding environment and data transmission accuracy, the image itself contains a lot of noise, which increases the complexity of image processing and feature matching. In addition, the robot object is a complex time-varying dynamic system, and there is strong coupling between their degrees of freedom. All these indicate that there are still many unsolved problems in the research of robot visual areas, which need to be further studied and discussed, and it is a very valuable subject.

DTW is a measurement of the similarity between two time series. Its usage in speech recognition is to discover if two words constitute the same phrase. In the time collection, the period of the two time series may not be identical, and the DTW calculates the similarity between the two time collections by extending and shortening the time series [150]. In this thesis, the Baxter robot is used to test the proposed teaching method by being commanded to perform obstacle avoidance tasks after the DMP was applied. Then the KUKA LBR robot is used to prove the achievement of our designed teaching method by drawing curves in a horizontal flat paper programmed by recording a sequence of actions taught by a human demonstrator. After that, the DTW and GMR are used to analyse and generalize

Table 6.1: Model characters table using Denavit & Hartenberg's Method [10]

LinkNumber	θ_i	$d_i(m)$	$a_i(m)$	$\alpha_i(rad)$
1	θ_1	0	0	$\pi/2$
2	θ_2	0	0	$\pi/2$
3	θ_3	d_3	0	$\pi/2$
4	θ_4	0	0	$\pi/2$
5	θ_5	d_5	0	$\pi/2$
6	θ_6	0	0	$\pi/2$
7	θ_7	0	l_7	0

the recorded movements. This allows the robot to function as proposed and the validation of our developed teaching interface has been illustrated.

6.2 The Enhanced Teaching Interface

In this section, we investigate the algorithms for teaching process, playback and the generalizing task, including the DMP, the GMR and the DTW.

6.2.1 Calculation of Arm Joint Angles

According to [8], a Denavit-Hartenberg (DH) featured system chart, as can be seen in Fig.6.1, was created to represent the 7-DOF model of our human arm. The DH kinematic parameters of the human arm are indexed in Tab.6.1. In step with the DH approach, the outline of the coordinate frames transformation from body i to border $i-1$ can be calculated.

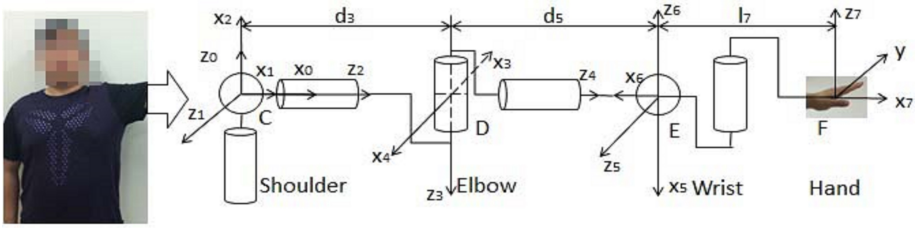


Figure 6.1: Human arm model and its DH coordinate frames [8]

The skeleton data of a human in 3-D positions can be obtained by using the Kinect sensor, which consist of 25 joints and this is shown in the left side of Fig.6.2. The arm model was made visible with a geometry model, which is shown on the right side of Fig.6.2

of our previous work [8]. Next, the point Hip-Left is selected as the origin, at the same time as x-axis is within the identical path of vector $\overrightarrow{AO}$ and y-axis is in conjunction with vector $\overrightarrow{OC}$ [8]. It is easy to align the regular vector of every axis to the base coordinate,$\overrightarrow{X_0}$, $\overrightarrow{Y_0}$ and $\overrightarrow{Z_0}$ [8]:

$$\overrightarrow{X_0} = \frac{\overrightarrow{AO}}{|\overrightarrow{AO}|}; \tag{55}$$

$$\overrightarrow{Y_0} = \frac{\overrightarrow{OC}}{|\overrightarrow{OC}|}; \tag{56}$$

$$\overrightarrow{Z_0} = \overrightarrow{X_0} \times \overrightarrow{Y_0} \tag{57}$$

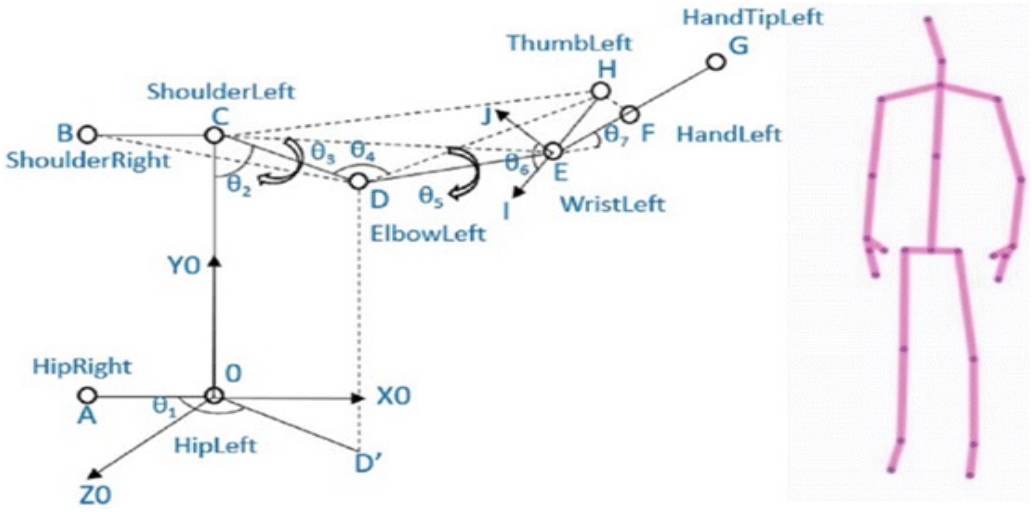


Figure 6.2: Screenshot of the skeleton tracking system and the geometry model for human arm in joint space [8]

In our previous work [8], it was found that the plane COD and plane xOy form the supplement angle θ_1.

$$\theta_1 = \pi - < \overrightarrow{CO} \times \overrightarrow{CD}, \overrightarrow{CB} \times \overrightarrow{CO} > \tag{58}$$

θ_2 is the angle formed by vector $\overrightarrow{CD}$ and y-axis, which is shown as follows [8]:

$$\theta_2 = < \overrightarrow{CO}, \overrightarrow{CD} > \tag{59}$$

Similarly, the plane BCD and plane CDE form the angle θ_3 as defined below [8]:

$$\theta_3 = < \overrightarrow{CB} \times \overrightarrow{CD}, \overrightarrow{CD} \times \overrightarrow{CE} > \tag{60}$$

$$\theta_4 = < \overrightarrow{DC}, \overrightarrow{DE} > \tag{61}$$

θ_5 is the angle between plane CDE and DEH [8].

$$\theta_5 =< \overrightarrow{EC} \times \overrightarrow{ED}, \overrightarrow{ED} \times \overrightarrow{EH} > \tag{62}$$

Angle θ_6 is formed by vector $\overrightarrow{ED}$ and plane EFH [8].

$$\theta_6 = \Pi/2 + < \overrightarrow{EH} \times \overrightarrow{EG}, \overrightarrow{ED} > \tag{63}$$

However, the angle θ_7 is difficult to calculate using the above method, which is the yaw angle of the wrist [8]. Here we define a vector $\overrightarrow{V}$ which is in the plane of EFH and perpendicular to $\overrightarrow{DE}$. θ_7 can thus be measured by forming $\overrightarrow{V}$ and $\overrightarrow{Y_7}$ [8]. While,

$$\overrightarrow{X_7} = \frac{\overrightarrow{EF}}{|\overrightarrow{EF}|}; \tag{64}$$

$$\overrightarrow{Z_7} = \frac{\overrightarrow{EF} \times \overrightarrow{EH}}{|\overrightarrow{EF} \times \overrightarrow{EH}|}; \tag{65}$$

$$\overrightarrow{Y_7} = \overrightarrow{Z_7} \times \overrightarrow{X_7} \tag{66}$$

So now the problem becomes to solve $\overrightarrow{V}$ [8]. We know that $\overrightarrow{V}$ is in the plane of EFH and $\overrightarrow{V}$ is perpendicular to $\overrightarrow{DE}$. Supposing that:

$$\overrightarrow{V} = k_1\overrightarrow{EF} + k_2\overrightarrow{EH} \tag{67}$$

There are:

$$(k_1\overrightarrow{EF} + k_2\overrightarrow{EH}) \times \overrightarrow{DE} = 0; |k_1\overrightarrow{EF} + k_2\overrightarrow{EH}| = 1 \tag{68}$$

Therefore,

$$\theta_7 =< \overrightarrow{V}, \overrightarrow{Y_7} > \tag{69}$$

By using the calculation above, every single joint angle can be obtained. However, two things have to be adhered to ensure that the calculations are accurate:

(1) the left thumb must be in the same plane as the palm; (2) when the angular vectors change from zero to π, then further problems need to be addressed [8].

Those above seven angles and their preliminary positions are shown in Fig.6.1 in the joint space. The proposed geometry vector technique is developed based on the precept of cosine cost of two vectors proven in (62). Furthermore, the posture between two planes may be calculated by giving them regular vector [8].

6.2.2 Data Preprocessing

The DTW algorithm is based totally on the concept of dynamic programming, and it aims to locate the shortest distance and highest quality matching path between two distinct check samples and reference templates. Let us define the reference time collection as $T = \{t_1, t_2, t_3, \cdots, t_i, \cdots, t_{L_1}\}$ and the test sample as $R = \left\{r_1, r_2, r_3, \cdots, r_j, \cdots, r_{L_2}\right\}$, wherein t_i and r_j denote the joint posture values of the time factors, L_1 and L_2 denote the vector lengths. The space matrix $D(i, j)$ has been illustrated from (87) in Section 5.2.1, when the vectors T and R are nonlinearly matched.

In our research, the DTW approach was used to align the recorded patterns by giving a warped characteristic, $W = \left\{w_1, w_2, \cdots, w_p, \cdots, w(p)\right\}$, where $w(p) = (i_p, j_p)$ is the match factor [6]. Here the warped characteristic W is needed to decrease the gap between the check sample vector and the reference template vector. Therefore, the equation is defined as:

$$D = min \sum_{k=1}^{K} d[w(p)] \tag{70}$$

where $d[w(p)] = d[T_i(p), R_j(p)]$ describes the distance measure between the $i(p)^{th}$ feature of the test sample vector and the $j(p)^{th}$ feature of the reference template vector, which is usually characterized by a square measure defined as follows:

$$d[w(p)] = [T_i(p) - R_j(p)]^2 \tag{71}$$

In order to align the two samples, we need to construct a matrix grid of $m \times n$, the matrix element (i, j) represents the distance $d(T_i, R_j)$ between the two points T_i and R_j and each matrix element (i, j) represents the alignment of points T_i and R_j. DTW is aiming to finding a path through several grid points that is representing the aligned points for the two samples to be calculated. Here we firstly outline the minimum cumulative distance among the 2 factors as $D_{Acc}(i, j)$, then we find that [6]:

$$D_{Acc}(i, j) = d(T_i, R_j) + min_{(q_i, q_j)}[D_{Acc}(q_i, q_j)] \tag{72}$$

where (q_i, q_j) belongs to the set of all points within a certain path that exists between points $(1, 1)$ and (i, j). It can be seen from the above components that the minimal cumulative distance of the factor (i, j) is related to not only the local distance $d(T_i, R_j)$ of the eigenvalues T_i, R_j, but also the minimal cumulative distance earlier than this point in the coordinate system [6].

Hence we conclude that $(i, j - 1)$, $(i - 1, j)$ and $(i - 1, j - 1)$ for any point $c(p) = (i, j)$ inside the coordinate system can reach the preceding point of $c(p)$, so the selection of the preceding factor only needs to align with the three above factors. According to the equation below, we can calculate the equal DTW distance among the check pattern vector

and the reference template vector, which is shown as follows:

$$D' = D_{Acc}(L_1, L_2). \tag{73}$$

6.2.3 Trajectory Generation

The canonical system of DMP is an exponential differential equation given by:

$$\tau_s \dot{s} = -\alpha_h s \tag{74}$$

where $\tau_s > 0$ is the temporal scaling factor, $\alpha_h > 0$ is the stability parameter and s is the phase value varying from 0 to 1.

The transformation system is made up of two contents in Cartesian space: a spring damping system and a nonlinear term (see Section 7 of Chapter 3). The transformation function f presents the complex nonlinear system, and it transforms the result of the canonical system, which is given by:

$$f(s) = \sum_{i=1}^{N} w_s l(s) \tag{75}$$

where N is the number of GMM, $w_s \in R$ is the weight of $l(s)$, $l(s)$ are the variable values of the normalized radial basis functions, which can be given as follows:

$$l(s) = \frac{exp(-h_i(s - c_i)^2)}{\sum_{m=1}^{N} exp(-h_m(s - c_m)^2)} \tag{76}$$

where $c_i > 0$ are the centers and $h_i > 0$ are the widths of the Gaussian basis functions. N is the number of the Gaussian functions.

After selecting the start line x_0 and goal g of the canonical system s=0, and integrating the canonical system, we are able to generate motion by the usage of the weight parameter [151]. DMP is used here to obtain the nonlinear transformation characteristic $f(s)$ through skill transfer from the demonstrator. However, there is an issue in creating the transformed system through the usage of more than one verified path. We, thereafter applied the GMM to overcome the above problems.

The GMM is the estimation of the probability density distribution of the samples [152]. The estimated value is the weighted sum of numerous Gaussian models and every Gaussian version represents a class [152]. In this thesis, the joint probability of the nonlinear system, which is the teaching data encoded through GMM, and the records were reconstructed via GMR to generalize the movement trajectory. For any degree of freedom, given j teaching data factors $\xi_j = \left\{f_j, s_j\right\}$, $j \in R$, where s_j and f_j had been defined in the DMP segment, N is the number of records points contained in a single training, every teaching

data ξ_j follows the subsequent probability distribution:

$$p(\xi_j) = \sum_{k=1}^{K} p(k)p(\xi_j \mid k) \tag{77}$$

where $p(k)$ is the prior probability, $p(\xi_j \mid k)$ is the conditional probability distribution, which follows the Gaussian distribution, and K is the number of Gaussian model distribution. Thus, the whole set of teaching data can be expressed by the GMM as follows [153]:

$$p(k) = \pi_k \tag{78}$$

$$p(\xi_j \mid k) = N(\xi_j, \mu_k, \sum\nolimits_k) = \frac{1}{\sqrt{(2\pi)^D \left|\sum_k\right|}} * e^{-0.5(\xi_j-\mu_k)^T \sum_k^{-1}(\xi_j-\mu_k)} \tag{79}$$

where D is the dimension of the GMM encoding the teaching data. Here we use the Bayesian Information Criterion (BIC) method to obtain the value of K [154].

$$\begin{aligned} S_{BIC} &= -L(\xi_j) + \frac{n(K)}{2} lgN \\ L(\xi_j) &= \sum_{j=1}^{N} lg(p(\xi_j)) \\ n(K) &= K - 1 + K(D + \frac{1}{2D(D+1)}) \end{aligned} \tag{80}$$

where $L(\xi_j)$ measures the model's characterization of data, n(K) is the number of free parameters of the model, which is a measure of the complexity of the model.

The GMM parameters need to be determined and are denoted as $\{\pi_k, \mu_k, \sum_k\}$. That is the k^{th} component of prior probability, expectations and variance, respectively. The EM algorithm is used to estimate the GMM parameters, which are obtained by giving the maximum similarity estimation of the parameters in the probability model, expectations and variance, respectively.

The teaching data s_j is used as the query point, and the corresponding spatial value f'_j is estimated by GMR. It is known that $p(\xi_j \mid k)$ satisfies the Gaussian distribution, $\binom{f_{j,k}}{s_{j,k}} \sim N(\mu_k, \sum_k)$, where $\mu_k = \left\{\mu_{f,k}, \mu_{s,k}\right\}$, $\sum_k = \begin{Bmatrix} \sum_{f,k} & \sum_{fs,k} \\ \sum_{sf,k} & \sum_{s,k} \end{Bmatrix}$, and the conditional probability $f_{j,k}$ satisfies the Gaussian distribution as given $s_{j,k}$ [153].

Then we have:

$$f_{j,k} \mid s_{j,k} \sim N(\mu'_{f,k}, \sum\nolimits'_{f,k}) \tag{81}$$

$$\mu'_{f,k} = \mu_{f,k} + \sum\nolimits_{fs,k} \sum\nolimits_{s,k}^{-1} (s_{j,k} - \mu_{s,k}) \tag{82}$$

$$\sum\nolimits'_{f,k} = \sum\nolimits_{f,k} - \sum\nolimits_{fs,k} \sum\nolimits_{f,k}^{-1} \sum\nolimits_{sf,k} \tag{83}$$

where we are able to calculate the variance $\sum_f'$ and the average μ_f' of the k^{th} GMM component, which is shown as follows [153]:

$$\mu_f' = \sum_{k=1}^{K} \eta_k \mu_{f,k}' \tag{84}$$

$$\sum_f' = \sum_{k=1}^{K} \eta_k^2 \sum_{f,k}' \tag{85}$$

$$\eta_k = \frac{p(s_j \mid k)}{\sum_{i=1}^{K} p(s_j \mid i)} \tag{86}$$

where $\mu_f{}'$ is the estimation acquired through the distribution of the expected conditions, and f_j is similar to the reconstruction of area values, known as $\mu_f{}'=f_j{}'$. The generalized data of points ($f_j{}'$,s_j) are not included in the teaching data, but encapsulates all the essential features of the teaching behaviour. Furthermore, under the relevant constraints $\sum_f{}'$, it can generate a smooth and reliable motion trajectory, which results in the effective control of the robot.

6.3 Experimental Studies

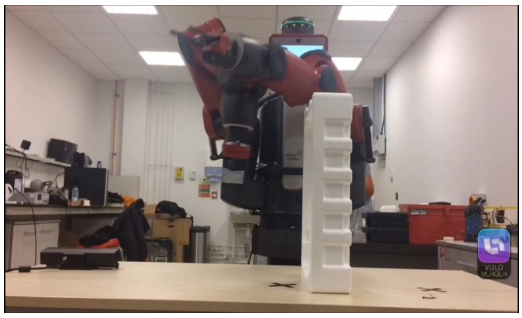
A Baxter robot and a KUKA LBR robot were used in our experiments to verify the effectiveness of the proposed method. As for the experimental platform, the PC operation system was Windows 10. There was also Kinect SDK for windows, Visual Studio 2013 and OpenCV library. The KUKA robot was programmed via Workbench, which is a common modifying platform combined with KUKA robot language (KRL) and Java. The experiments were conducted in a well-lighted environment.

6.3.1 Obstacle Avoidance Experiment

Several tests were designed to test the performance of our designed system by controlling the Baxter robot to navigate a high obstacle. One person, a demonstrator, stood two metres in front of the Kinect sensor. The operator guided the Baxter robot to carry out its tasks via teleoperation, as shown in Fig.6.3.

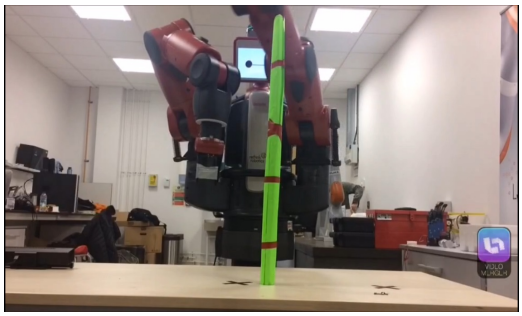
At the same time, those data of each joint of operator's arm is recorded, which is used for the playback of the Baxter robot. After that, we increase the height of the obstacle, it is noticed that the Baxter is not able to pass the obstacle successfully. Hence, the DMP has been employed to generalize the trajectory of the Baxter robot. By doing this, the Baxter is finally able to pass through the obstacle successfully with the increasing height.

(a) Obstacle avoidance by teleoperation with Baxter



(b) Obstacle avoidance by playback

(c) Failed to pass the obstacle with increasing height



(d) Succeeded to pass the obstacle with increasing height after applying DMP

Figure 6.3: Illustration of the obstacle avoidance experiment

6.3.2 Trajectory Generalizing Experiment

For this experiment, we tested a KUKA LBR robot, using recorded data of movements of a demonstrator saved locally and processed using MATLAB. The data generated by the trajectories were sent to a separate computer and used to control the robot arm. A marker pen was connected to draw patterns on a horizontal flat surface. A sine wave was chosen as the reference trajectory for the demonstration, wherein the ability of the designed technique is tested with complex shapes. During the experiment, we used a pre-revealed template shaped on a sheet of A4 paper. Then, a human operator guided the KUKA robot as shown in Fig.6.4. The movement of the robot endpoint was recorded during the demonstration.

The five recorded trajectories in Cartesian space were saved and analysed using the K-means method and EM algorithm to obtain the GMMs. The experimental trajectories were plotted through MATLAB. After that, the DTW was used to align the 5 trajectories, with the first curve selected as the reference to align all the other patterns. GMM was used to encode the trajectories. In the end, the KUKA robot was able to reproduce a generalized

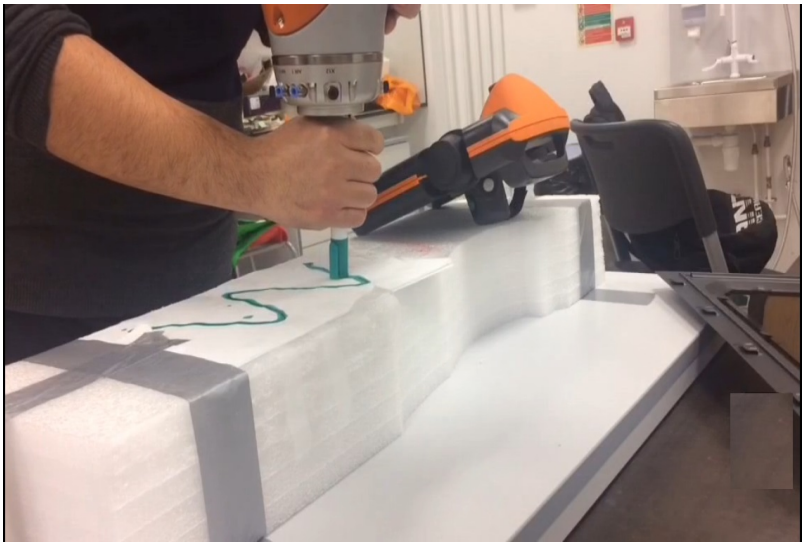


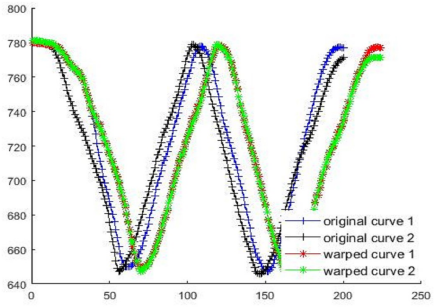
Figure 6.4: The setup of the trajectory generalizing experiment

curve on the vertical surface.

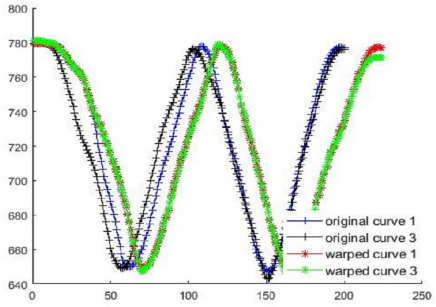
6.3.3 Results

The first group of experiments aimed to verify the performance of the proposed DMP, including the ability to generalize, i.e., spatial scaling, and the learning performance when the demonstration is defective. In this experiment, the demonstration process used joints $s1$, $e0$ and $w2$ fixed, and the angles of the joints $s0$, $e1$, $w0$ and $w1$ which are recorded during the demonstration. Then the demonstration data are used for the training of the modified DMP. The training result is shown in Fig.6.6. It can be seen from the graph that the maximum and the minimum values of *ShoulderPitch* between the data from demonstration and playback, in some specific time point, differ by about 0.4 radians. This implies that the range of the arm motion of the Baxter robot increased at about 0.4, leading to an accurate motion. Here the theoretic explanation is that the *ShoulderPitch* joint angle decides the maximum high that a robot can touch and we use the DMP generalization to increase the moving range of this angle. The motion of joint *ShoulderPitch* is regenerated from the demonstration, which synthesized the features of the demonstration and enabled the robot to perform the obstacle passing task successfully as shown in Fig.6.3(d).

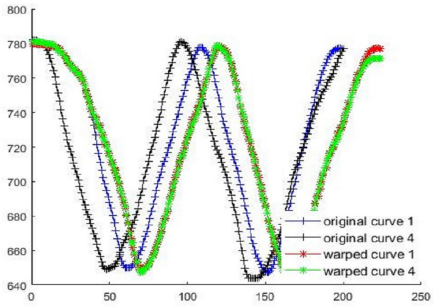
As shown on the left side of Fig.6.7, for the second experiment, five distinctly separate curves were drawn horizontally. It can be concluded that the number of GMMs affected the result trajectories. Hence, to achieve good performance of the generated trajectories, the number of GMM components was pegged at 20. In addition, the optimal result trajectory is shown on the right side of Fig.6.7. By using the GMR method, we transform



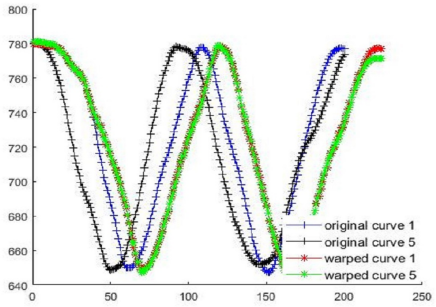
(a) Alignment result between the first and the second curves



(b) Alignment result between the first and the third curves



(c) Alignment result between the first and the forth curves



(d) Alignment result between the first and the fifth curves

Figure 6.5: Illustration of the alignment using DTW

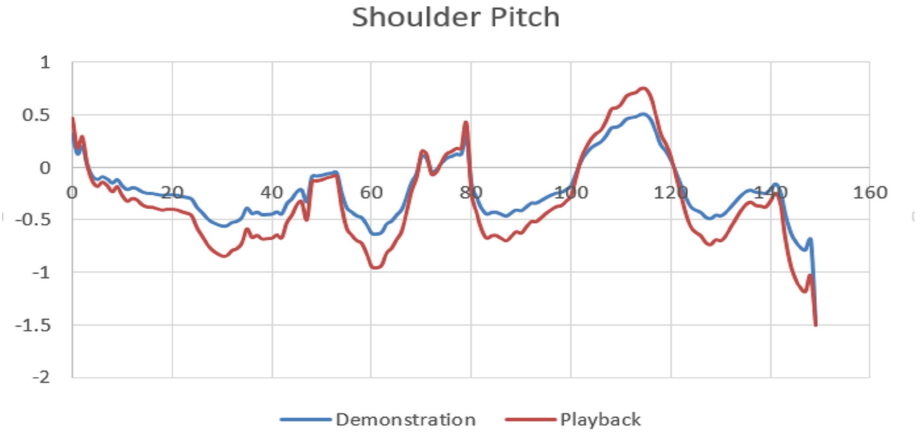


Figure 6.6: The learning and generalization result using the proposed DMP in an obstacle passing task

the data retrieval problem of TbD into a joint distribution estimation problem, which is approximated by a mixture of Gaussians. During the calculation, the key point of the learning process was correlative to the number of points in the sample set of data linearly. Here the prediction process relied on this number.

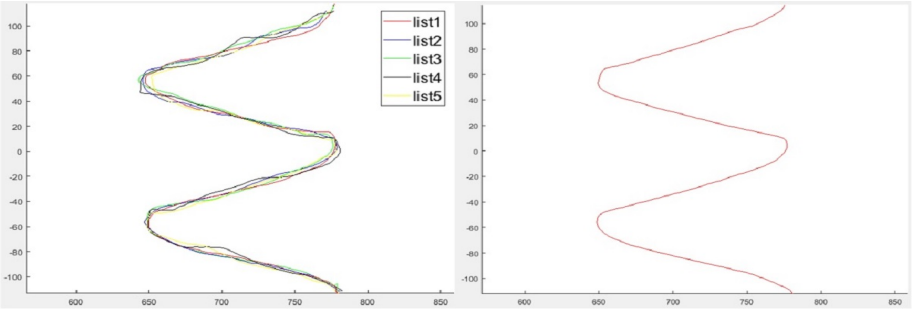


Figure 6.7: The demonstrated trajectories for the sine wave with GMM and the result

After that, we modified the DMP code to apply the spatial and temporal generalization. The generalized curve was then drawn on a vertical flip chart pad by the playback process of the KUKA robot, as shown in Fig.6.8. A smooth curve was retrieved from multiple demonstrations using the modified DMP, where the playback process was achieved at 5 times the speed, proving the proposed temporal generalization. Future works would look into using DMP segmentation for teaching by demonstration-based tasks.

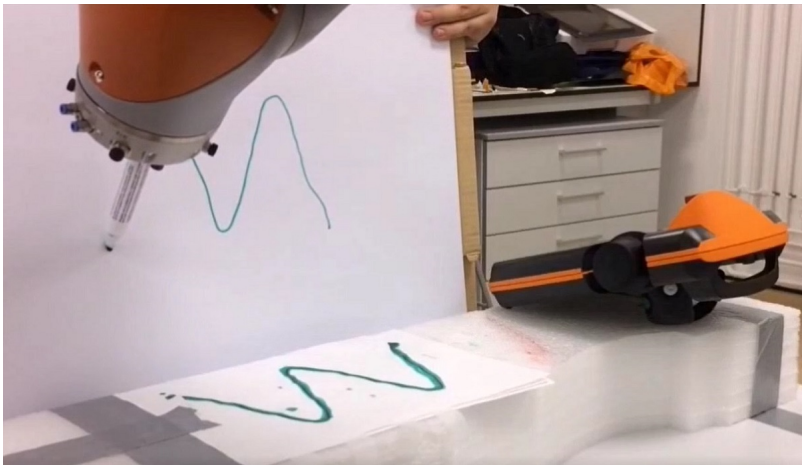


Figure 6.8: Curve on a vertical surface obtained after spatial generalization using the modified DMP

6.4 Conclusion

A GMR and DMP combined with DTW based TbD technology was developed in this thesis, which seeks an effective and superior method for humans to interact with robots. The Kinect V2 sensor was used to teleoperate the Baxter robot for the accurate generation of the motions. For the motion generation, the discrete DMP was selected as the basic motion model, which can achieve the generalization of the motions. To improve the learning performance of the DMP model, the GMM and GMR were employed for the estimation of the unknown function of the motion model. The DMP model was enabled to retrieve a better motion from multiple demonstrations of a specific task. Two experiments have been employed to test the performance of our designed teaching interface. The results verified the effective generalization of the proposed methods. Compared to standard teaching approaches, the proposed teaching interface evaluated the DMP for multiple demonstrations by combining with GMR after all the experimental data had been initially pre-processed by DTW.

7 Chapter Seven: Development of Writing Task Recombination Technology Based on DMP Segmentation via Verbal Command for Baxter Robot

7.1 Introduction

Teaching by demonstration (TbD) technology is fast gaining ground in the field of robotics. This is due to the advancement in the quick and efficient transfer of skills. With the aid of human guides, robots are able to acquire the dexterity required to carry out tasks. This mode of human-robot skill transfer has a number of advantages which include:
(1) TbD does not require a human instructor with expert skills and knowledge;
(2) human-robot skill transfer is achieved in a convenient and efficient manner;
(3) it effectively takes human factors such as flexibility and compliance into account [11].

These benefits facilitate task accomplishment [11]. Character-learning based on TbD technology with trajectory matching is a highly-researched topic in imitation learning and has received considerable attention in the past few decades. Conventional approaches in this area constitute spline-based methods [155], dynamic system methods [156] and probabilistic model methods [157]. The main advantage of robot learning is that it seeks effective control strategies to complete complex motion tasks which are challenging to attain through traditional methods.

Spline-based methods can generate trajectories quickly. However, they are time-dependent, sensitive to interference, do not have the ability to adjust in real-time, and the calculations need to be revised when new data is received. Dynamic system approaches that model discrete and rhythmic motions, are not time-dependent, and they allow for real-time adjustments. It has topological equivalence and is often used as a dynamic movement for high level characterization primitives, such as DMP [158] for the construction of complex modified primitive libraries (complex motions consisting of simple motions represented by a series of primitives). However, DMP is not suitable for direct encoding of complex motions and requires more teaching information (position, speed and acceleration). Probabilistic model methods, such as a hidden Markov model (HMM), GMM, dynamic Bayesian network (DBN), are often used to match trajectory [31]. Probability models have strong coding and noise-processing abilities, excellent robustness, which can deal with high-dimensional problems. GMM, in particular, has a strong ability for encoding and reproducing continuous complex trajectories. Compared to the DMP, it only needs space and position-based teaching information that enables it to be used for imitation learning of complex motions.

The basic theory of GMM is that as long as the number of Gaussian mixtures is sufficiently large, an arbitrary continuous distribution can be approximated by calculating the weighted averages of the Gaussian mixtures with arbitrary precision. It is widely used

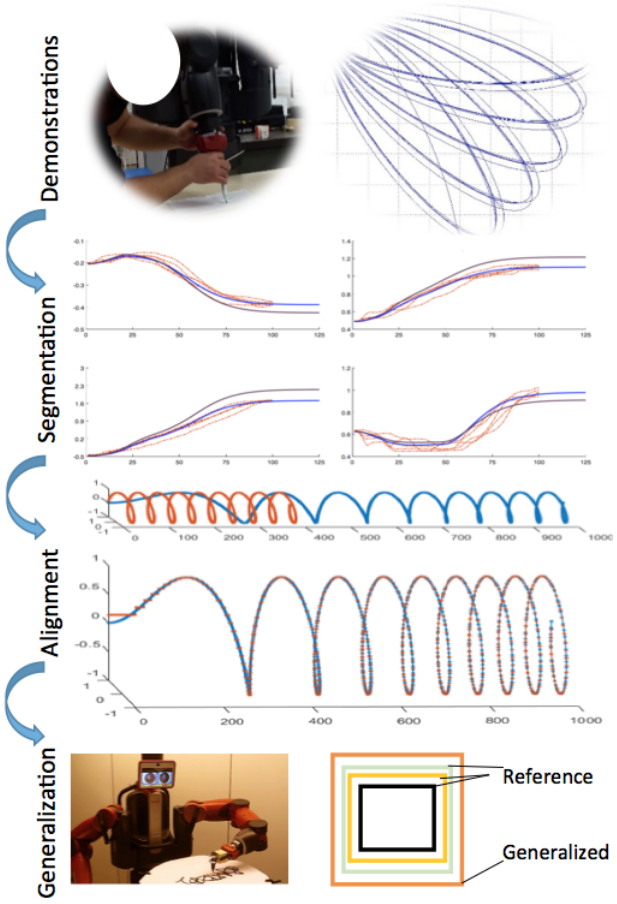


Figure 7.1: Graphical representation of the overview of the proposed technology, modified from [9]

in trajectory generation for robot simulation learning and has a strong behavioural coding capability. For example, in the research [159], by training the teaching data to the robot, learning GMM and stable estimation of the multi-dimensional dynamic system of nonlinear motion are achieved. These cannot be only generalized to reach the unknown position but can also be adjusted online in case of interference. In [160], the framework of variance-based imitation learning is given in the task space, the motion is modelled by using GMM, the expectation maximization (EM) algorithm is used to reconstruct the trajectory, which is generated by using the GMR and the optimization evaluator realizes the imitative learning. In [161] and [162], the authors employ a Gaussian process to establish a random forward model to represent the motion to be simulated with the Kullback-Leibler divergence as an indicator of imitative performance, to ensure effective learning of the data through distribution prediction and to perform trajectory matching, and finally to achieve motion simulation. Content-based retrieval methods have gained significance in motion-

captured data retrieval [163]. During the data matching process, the start frame and the end frame of the search condition sequence are first indexed into the library to select possible alternative segments in the motion capture database, and finally a DTW method is used to calculate the similarity to determine the final search results [164]. With the continued developments in robot research, it has come to be known that robot movement behaviour is highly complex, and requires the robot's learning ability to be more demanding. The traditional algorithms cannot achieve complex movements that are not obtained solely through laws of movement, such as hitting the ball and a writing task [164]. Furthermore, robots are required to have enhanced learning ability so that it can self-compensate, interact with a random dynamic environment, and deal with sudden and unknown situations. However, the existing methods are to predict the model in the finite time domain, and the global stability of the system is difficult to prove.

Writing is a complex task with problems that include continuous complex trajectory characterization and discrete trajectory generation. Some scholars have proposed a control chart model [165] and a recurrent neural network [166] to teach robots writing skills, however, they fall short of fixing the problems. Though the control chart model can generate discrete trajectories, the ability to represent complex trajectories is not enough, and the recurrent neural network can only be used for the reproduction of simple trajectories. Owing to the above problems, a GMM-based teaching method is required that can achieve complex trajectory characterization for acquiring writing skills.

The test of the imitation learning process in this thesis aims to encode the teaching data through GMM, extract the behavior characteristics, and reconstruct the data through GMR, so as to actualize the continuous Chinese character writing skills in trajectories. This method, with the aid of the DMP, is found to effectively solve problems. Based on the basic DMP, a multi-task extension is applied by teaching the robot every single unit of the Chinese characters needed to achieve the regroup tasks after segmentation of trajectories. The Baxter robot successfully learns the writing skills for scribbling Chinese characters with a non-continuous trajectory through good coding ability and generalization performance. The framework that has a basic TbD procedure is divided into four phases: demonstration, segmentation, alignment and generalization [9]. The overview of the framework is shown in Fig.7.1.

7.2 Methodology

In this section, we investigate the proposed teaching process method, playback and the DMP segmentation-based regroup tasks, including the proposed DMP promoted by adding GMM, after the preprocessing using DTW. Here we apply the DTW to match the similarity between the character and the candidate domain, using the weighted distance in both directions as the final distance.

7.2.1 Data Preprocessing using DTW

DTW is an effective time series matching method and is widely used for time series processing and signal processing [164]. Earlier applications were applied in areas such as speech recognition. Given the first character sample C as the reference with a width M and the second character sample Q to be aligned with a width N, the size of the reference area is the approximate as the size of the character to be aligned, $M \approx N$.

Assuming that the writing task is well-structured and continuous, for any two columns where there is no cross-matching occurring in between, and each column can find another to be matched with, continuous results were achieved. The restriction ensures that the $i-th$ column of the sample C and the $j-th$ column of the sample Q have an accumulated distance $D(c_i, q_j)$, and it is only jointly determined by $D(c_i, q_{j-1})$, $D(c_{i-1}, q_j)$, $D(c_{i-1}, q_{j-1})$ and $D(c_i, q_j)$, which is shown as follows [164]:

$$D(c_i, q_j) = min \begin{Bmatrix} D(c_i, q_{j-1}) \\ D(c_{i-1}, q_j) \\ D(c_{i-1}, q_{j-1}) \end{Bmatrix} + d(c_i, q_j) \tag{87}$$

where both $i, j > 1$, $d(c_i, q_j)$ is the distance function between two samples, and the distance between element c_i in column i in C and element q_j in column j in Q, is defined by the Euclidean distance defined as [164]:

$$d(c_i, q_j) = (c_i - q_j)^2 \tag{88}$$

Although there is a difference in writing between the same characters, this difference should be kept within a small local area. Therefore, the global path must be constrained to maintain the invariance of the local structure and accelerate the solution of the problem. It is important to limit the spatial distance between the two columns c_i and q_j to be less than r elements [167].

$$\|i - j\| \leq r \tag{89}$$

$$r = \lceil k * seqL \rceil \tag{90}$$

where k is a constant coefficient and $seqL$ is the length of the feature sequence.

Using the dynamic programming method can accelerate the solution of the horizontal distance $D(c_M, q_N)$. Since the feature sequence length of the sample Q to be aligned and the feature sequence length of the reference domain are the almost the same, and it is not necessary to normalize the accumulated distance $D(c_M, q_N)$ by the sequence length [167]. Note that the length and height of the characters (as the dis-aligned sample and the reference sample) are inconsistent, we used the weighted two-way DTW algorithm to calculate the final distance of two characters, the weight is the length of the sequence itself:

$$dist(C, Q) = D(c_N, q_N) * N + D(c_M, q_M) * M \tag{91}$$

Finally, each reference sample C and the dis-aligned sample Q were arranged in ascending order from $dist(C, Q)$ to obtain a list of candidate regions. Since each sample will have several key points, the overlapping candidate fields need to be eliminated. According to the obtained candidate sample list, for any single character inside, if there is an order prior to, and if the overlapping area ratio exceeds the threshold (here we define it as 0.2), it is eliminated. The final list is the result of the optimization.

7.2.2 Trajectory Generation

The basic idea underlying DMP is to use an easily understood dynamic system to achieve the expression of motion trajectory [168]. The spring damping system is the simplest instance, as shown in (2) of Chapter 3. Here we use a proportional coefficient to scale the trajectory shape when the new target point is farther than the initial target point of the teaching trajectory.

Equation (2) represents a transformation system. Every independent transformation system refers to one degree of freedom, where $f(s)$ is a nonlinear disturbance force and can be generated through learning. A specific expression is as follows:

$$f(s) = \frac{\sum_i \omega_i \varphi_i(s) s}{\sum_i \varphi_i(s)} \tag{92}$$

$$\tau_s \dot{s} = -as \tag{93}$$

$$\varphi_i(s) = exp(-h_i(s - c_i)^2) \tag{94}$$

where φ_i is the Gaussian function, wherein c_i is the center, h_i is the width. By adjusting the weight ω_i, equation (92) can be used to express arbitrary shape trajectories. a is constant, s is a phase parameter with the value from 1 monotonically converges to 0 [13]. We can conclude that during the process, the external factors decrease as it nears the target position g, which ensures the stability of the system to the goal. Equation (93) represents a canonical system, which is used to realize the coupling between multiple transformation systems, and is not directly dependent on time [13].

The policy parameter ω_i can be commonly learned using supervised learning algorithms such as locally weighted regression (LWR). To put it simply, in this thesis, we obtain the weight via finding a proper parameter vector by minimizing the following error, where the f_{target} can be obtained from the equation (2) from the Section 7 of Chapter 3. Combining equations (2) with (92), on the basis of least square method, this locally weighted linear regression problem can be solved efficiently. If J equals the minimum, ω_i is the optimum.

$$J = \sum_s (f_{target} - f(s))^2 \tag{95}$$

After selecting the starting point x_0 and target g of the canonical system s=0, then integrating the canonical system, we can generate a movement by using the weight parameter. The principle of DMP is to obtain the nonlinear transformation function $f(s)$ by learning from the movements of the demonstrator [13]. So far, the spatial value $f(s)$ and the temporal value s have been obtained. However, there is a limitation on the creation of the transformed systems with multiple demonstrated paths, hence we have employed the GMM to overcome the above issues.

The teaching data is acquired by the motion capture system, and the spatial coordinate information of the end-effector of the teacher is obtained. First, dimensionality reduction techniques, such as principal component analysis, are used to perform data preprocessing, and the three-dimensional data is mapped into two-dimensional space to obtain two-dimensional data. Teaching data are inputted into the learning model during the data encoding. In order to simplify the data processing steps, the emphasis is on the representation learning and generalization of the system. The two-dimensional teaching data of this article $\alpha = \{\alpha_s, \alpha_t\}$ are obtained directly from the previous section, where we assume that α_s, α_t, respectively, representing the spatial value and the temporal value of the teaching information. The GMM consists of multiple Gaussian distributions of the value α of each element [169],

$$P(\alpha) = \sum_{i=1}^{K} \omega_i \eta(\alpha; \mu_i, \sum_i) \tag{96}$$

$$\eta(\alpha; \mu_i, \sum_i) = \frac{1}{\sqrt{(2\pi)^{n/2} |\sum_i|^{1/2}}} e^{-1/2(\alpha-\mu_i)^T \sum_i^{-1}(\alpha-\mu_i)} \tag{97}$$

where K is the number of Gaussian functions. The larger the K value is, the better the model can represent a complex situation; ω_i, μ_i and $\sum_i$ are the weights, mean values, and covariance matrices of the $i{-}th$ Gaussian distribution, which are required to be determined.

The EM algorithm is used to estimate the parameters of the GMM (ω_i, μ_i and $\sum_i$), and the parameters are learned by searching the parameters in the probability model [170]. The EM algorithm is commonly employed to estimate the parameter with hidden variables. The above is a maximum likelihood estimation problem. This algorithm continuously improves the lower bound of the likelihood function and optimizes the parameters [170]. The training process is very sensitive to the initial value of parameters, which need to be initialized by k-means clustering. After the parameters (ω_i, μ_i and $\sum_i$) are determined, the GMM model can be learned based on the data from demonstrations, and thus the skill is encoded by the GMM model.

In this research work, GMR is used to reconstruct the teaching data via GMM learning, by doing this, the generalized output is obtained. α_t of teaching data is used as a searching point, and its according spatial value α'_s is estimated by GMR. Note that $\eta(\alpha, \mu_i, \sum_i)$ meet

the Gaussian distribution [171]:

$$\begin{pmatrix}\alpha_{s,i}\\ \alpha_{t,i}\end{pmatrix} \sim \eta(\alpha, \mu_i, \sum_i) \tag{98}$$

where $\mu_i = \{\mu_{s,i}, \mu_{t,i}\}$, $\sum_i = \begin{Bmatrix}\sum_{s,i} & \sum_{st,i}\\ \sum_{ts,i} & \sum_{t,i}\end{Bmatrix}$, and the conditional probability of $\alpha_{s,i}$ satisfies the Gaussian distribution at the given $\alpha_{t,i}$, i [171]. Then we have

$$\mu'_{s,i} = \mu_{s,i} + \sum\nolimits_{st,i} \sum\nolimits_{t,i}^{-1} (\alpha_{t,i} - \mu_{t,i}) \tag{99}$$

$$\sum\nolimits'_{s,i} = \sum\nolimits_{s,i} - \sum\nolimits_{st,i} \sum\nolimits_{t,i}^{-1} \sum\nolimits_{ts,i} \tag{100}$$

and then the average μ'_s and variance $\sum'_s$ of the number i of GMM components can be calculated as follows [171]:

$$\mu'_s = \sum_{i=1}^{K} \eta_i \mu'_{s,i} \tag{101}$$

$$\sum\nolimits'_s = \sum_{i=1}^{K} \eta_i^2 \sum\nolimits'_{s,i} \tag{102}$$

$$\eta_i = \frac{p(\alpha_t \mid i)}{\sum_{n=1}^{K} p(\alpha_t \mid n)} \tag{103}$$

where the mean $\mu_s{}'$ is the required teaching data reconstruction value ($\mu_s{}' = \alpha_s{}'$), and finally the generalized data points $\alpha' = (\alpha_s{}', \alpha_t)$ and the variance memory for extracting task constraints $\sum_s{}'$ can be obtained. The generalized data points are not included in the teaching data, but they encapsulate all the essential features of the teaching behaviour. Under the relevant constraint of $\sum_s{}'$, smooth and reliable motion trajectories can be generated to achieve effective control of the robot.

7.3 Experimental Studies

7.3.1 Experimental Setup

A Baxter robot is used to verify the effectiveness of the proposed method. A marker pen is attached to the gripper of the robot. The operator physically guides the Baxter to write a Chinese character on a flat paper. The experimental setup is shown in Fig.7.2.

The experimental platform, Visual Studio 2013 and OpenCV library, are used within a Windows 10 operation system. The experimental environment was indoor and adequately illuminated. During the teaching process, the operator demonstrated how to write the Chinese character “Mu” five times. In doing this, we had four separate single primitives,

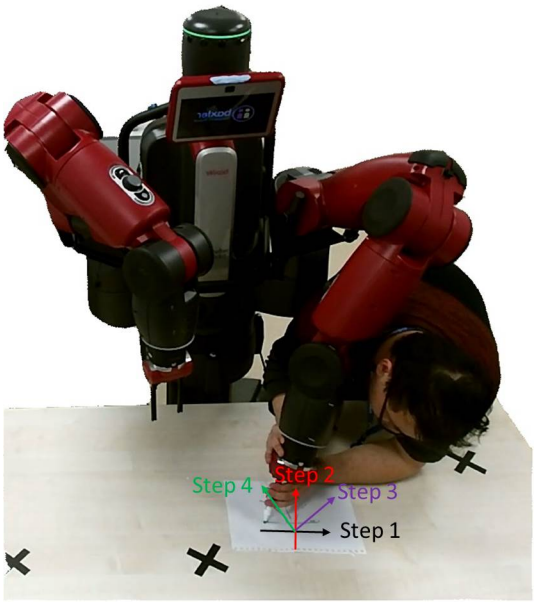


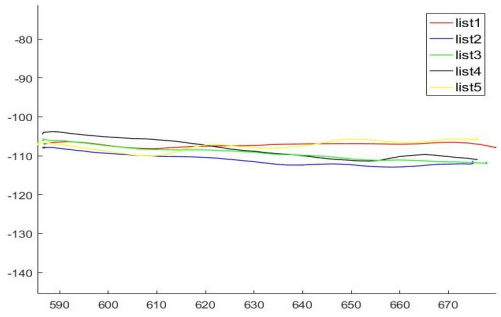
Figure 7.2: The experimental setup for the Chinese character writing task. Step 1: across stroke; Step 2: vertical stroke; Step 3: left-falling stroke; Step 4: right-falling stroke

which were generalized by DMP to regroup other Chinese words. There was a self-made implementation running on a remote PC to control the recording and playback of the trajectories of the Baxter robot. This was done by defining any text for the locally outputted trajectory files. In addition, Dragon NaturallySpeaking was installed on the remote PC, allowing transfer from voice to text signal, and the generation of robot motion control commands via UDP.

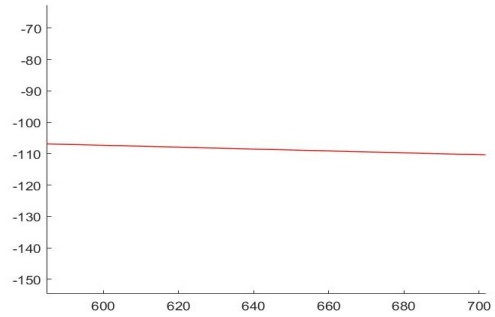
The demonstration process is repeated five times with the joint W_2 fixed, and the trajectories of the joints S_0, E_1, W_0 and W_1 with time were recorded. The demonstration data was thereafter used to train the modified DMP. The parameters of the DMP model were set as: $\tau_s = 1$, $K = 20$, $a = 8$. The GMM has a strong trajectory coding ability for complex tasks. This research used the JAVA based codes to acquire the writing data in the teaching mode, applied the GMM-based imitative learning for writing skills, and obtained the generalized output through the GMR, which was performed automatically by the Baxter robot. Based on the operator's demonstrated strokes, the second, third and fourth strokes of the Chinese character "Mu" were chosen to be generalized.

7.3.2 Results and Analyses

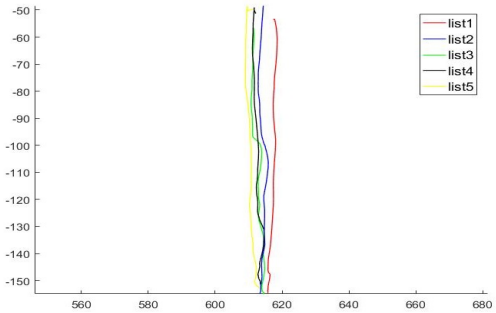
The experimental trajectories were plotted using MATLAB. The five recorded movement trajectories for each stroke were saved in Cartesian space, where the K-means method was



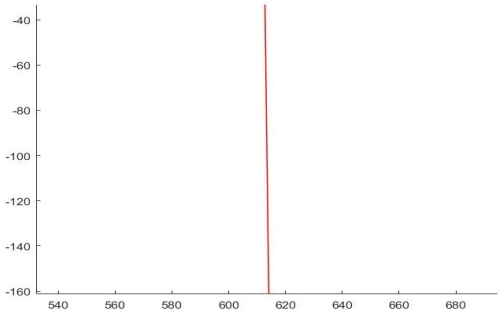
(a) : The 5 demonstrated trajectories for the across stroke



(b) : The result of trajectory for the across stroke using GMR



(c) : The 5 demonstrated trajectories for the vertical stroke

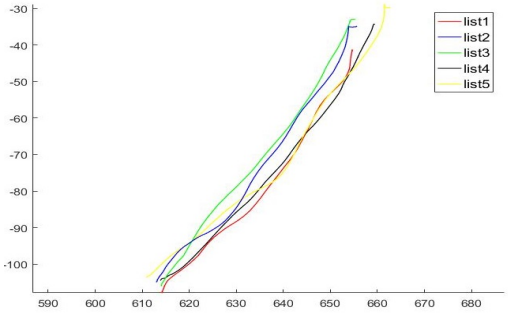


(d) : The result of trajectory for the vertical stroke using GMR

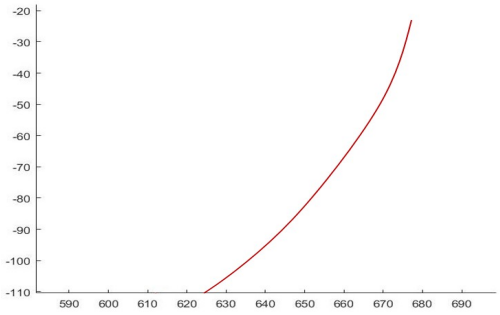
used to initial the analysed data. Thereafter, the EM algorithm was used to obtain the GMMs. After which the DTW method was used to align the five recorded trajectories, where the first curve was chosen as the reference to be aligned with others.

It can be seen from Fig.7.3 that b,d,f,h are GMR-reconstructed trajectories using MATLAB and Fig.7.3a,c,e,g are the five demonstrated trajectories. Here we take the second step to be spatially generalized, which is the across stroke. Using the GMM-based imitative learning, the trajectory can be continuously used to write the Chinese character “Mu”. The blue dotted line is the teaching trajectory by demonstration, the solid black line is the generated trajectory and the red solid line is the generalized result after DMP and GMM coding. Next, the first stroke and all the other generalized strokes are able to form a new Chinese character “Bu” by using the verbal commands orderly as shown in Fig.7.4. The demonstration process recorded the variable data of six joints S_0, S_1, E_0, E_1, W_0 and W_1. The joint W_2 was fixed to the value 0 and these data were used to train the modified DMP. The training results are shown in Fig.7.5.

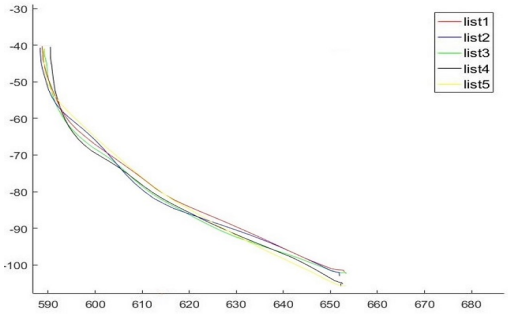
Obviously there are maximum and the minimum values of all the joints angles between



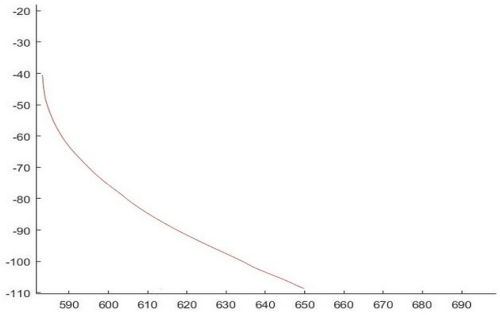
(e) : The 5 demonstrated trajectories for the left-falling stroke



(f) : The result of trajectory for the left-falling stroke using GMR



(g) : The 5 demonstrated trajectories for the right-falling stroke



(h) : The result of trajectory for the right-falling stroke using GMR

Figure 7.3: The demonstrated and reconstructed trajectories of the "Mu" character strokes, the x axis represents x direction and the y axis represents y direction

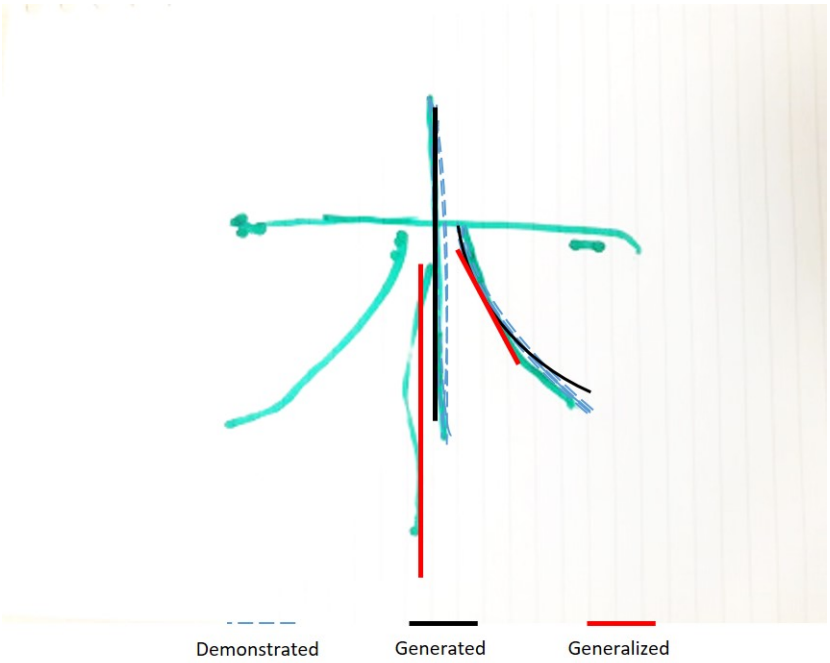
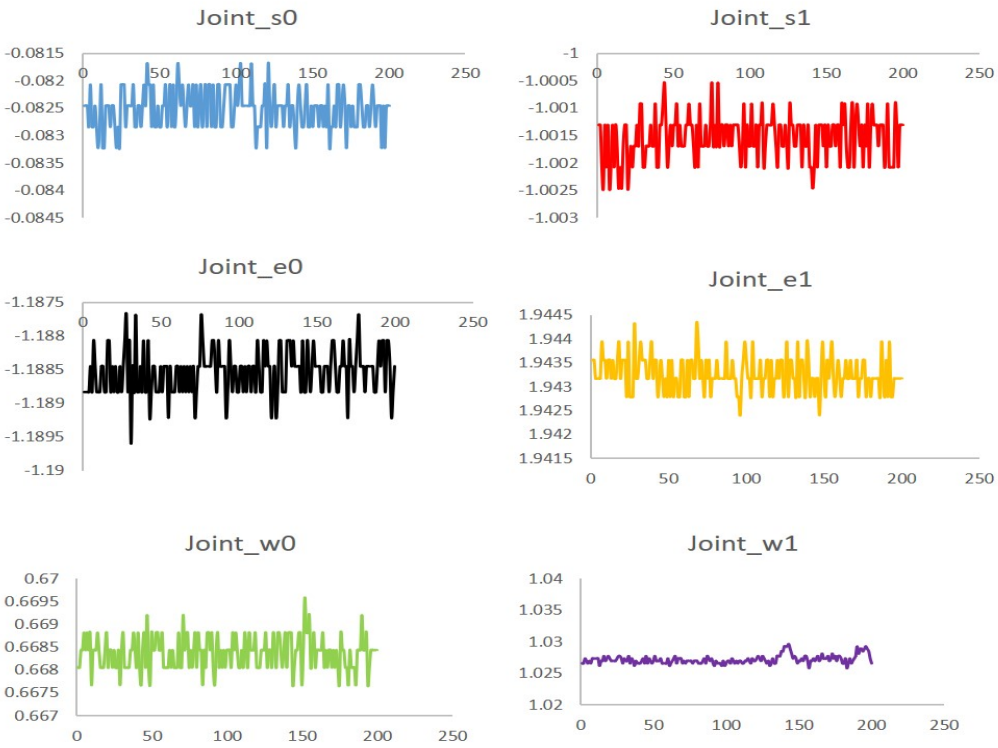


Figure 7.4: The initial and generalized Chinese character

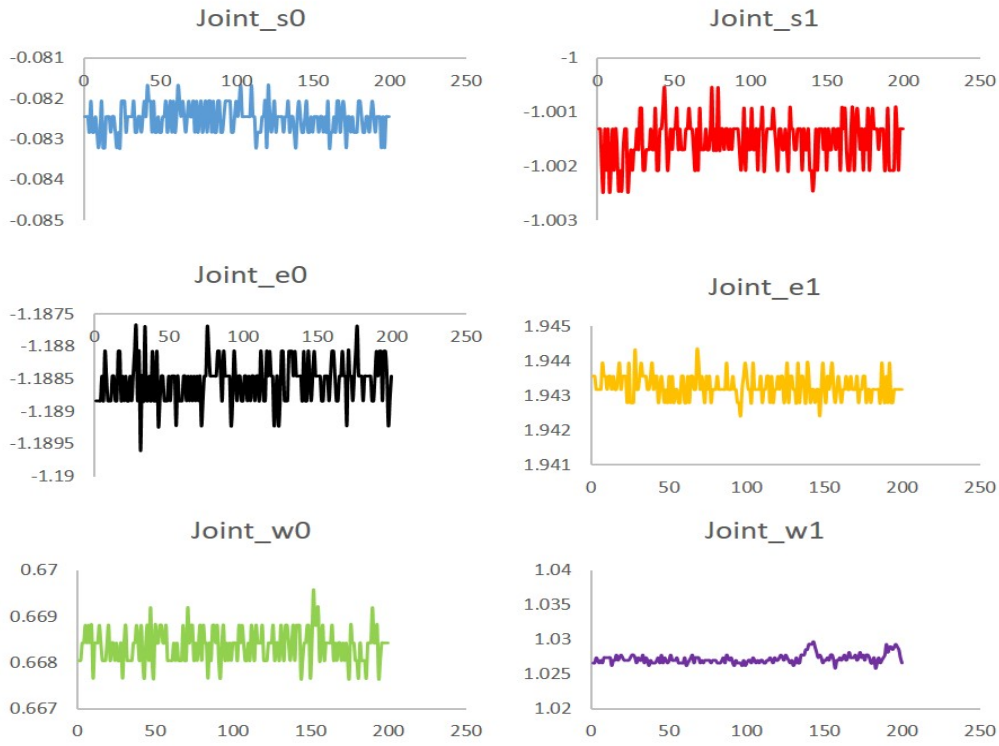
the demonstration and generalization. In some special time points, spaced about 0.04 radians apart (i.e. the range of the arm movement of the robot in those two situations differs at about 0.04), could lead to different arm motions in two different positions. The movement of joints S_0 to W_1 is regenerated through the teaching process, which enabled the robot to perform the Chinese character writing task successfully as shown in Fig.7.4, and synthesizes the features of our proposed technology as well. As shown in Fig.7.3, smooth curves are obtained from multiple demonstrations using the modified DMP. Hence, the robot acquired the writing skill after the human demonstration.

7.4 Conclusion

A Chinese character recombination technology based on DMP segmentation using verbal command for a Baxter robot is developed in this thesis. The technology performs well in continuous writing of trajectories with generalizations. In the motion generation part, we chose the discrete DMP as the basic motion model because it can generalize the motion trajectories. To promote the learning efficiency of the DMP model, we employ the GMM and GMR to estimate the unknown function of the model referring to the movement. With this modification, the DMP model is able to create better movement (under the multiple demonstrations based tasks). Through this method, the simultaneous encoding of multiple teachings effectively reduces the influence of errors, which, in turn, improves the stability



(a) : The value of angles for the initial vertical stroke



(b) : The value of angles for the generalized vertical stroke

Figure 7.5: The comparison of the joints angles for the vertical stroke of the “Mu” character with or without DMP, the x axis represents time (in seconds) and the y axis represents joint angles (in radians)

of the system.

Our proposed method can be extended to achieve multi-task learning and improve the system's co-processing capabilities. For this purpose, Dragon NaturallySpeaking software has been installed on the PC and provides an efficient voice interaction framework for TbD, facilitating a higher-level robot learning.

8 Chapter Eight: Conclusions

8.1 Summary of Contributions

The unprecedented advancement in modern technology has led to an ever-increasing demand for the rapid development in robotic technology. The complexity of today's robot system and the complexity of the work environment influence a robot's functioning in executing tasks and speeding performance. State-of-the-art, intelligent and controlled robots that can function autonomously have not yet been developed. In terms of the autonomous, safe and accurate control of robots that operate in complex environments there are still major challenges to be overcome. This thesis has studied those challenges from the perspective of human-robot skill transfer. The main content of this thesis is the design of the skill transfer method between industrial robots and human beings based on learning algorithms. Industrial robots are working in unknown and complex environments. The design of interactive control method needs to meet the requirements of high accuracy and safety.

This thesis first introduces the research status of industrial robots, then summarizes and analyses the research status, advantages and disadvantages of human-robot interaction control, machine learning methods and control methods in the field of robotics. In this thesis, common problems in robot teaching and behaviour reproduction, such as safety, control accuracy, adaptability and self-expansion, are studied in depth. This thesis studies the challenges of a human-robot skill transfer perspective. This has been done via investigating a learning algorithm design validated on a NAO robot, Baxter robot and KUKA LBR iiwa robot, respectively. The main contents of this research are summarized as follows:

1. Chapter one provides an introduction to robotic research and a short discussion of related publications;

2. Chapter two is a review of the literature related to the scope of the study. This review covers literature on the development of industrial robotics, the relating human-robot interaction, machine learning in robot areas and robot controller design;

3. Chapter three covers a discussion of robot platforms, equipment including the Kinect sensor and the MYO armband and basic mathematical methodologies including the DMP, GMM, GMR, DTW algorithms;

4. Chapter four illustrates a tracking system using a Kinect sensor to teleoperate with a NAO Robot. The operator is able to move his head and arm in a natural way to interact with the NAO robot, which imitates the operator's actions in real-time;

5. Chapter five covers the investigation of a NN learning and KF based TbD technique for a Baxter robot to further increase the accuracy of the teleoperation process by fusing the data collected from the MYO armband and Kinect v2 sensor. The NN system is employed to compensate for the uncertain dynamics that exist in the learning process.

6. In chapter six, an enhanced teaching interface for a robot using DMP and GMR is

developed. A Baxter robot is commanded to go around obstacles set at different heights. Thereafter, the KUKA LBR iiwa robot is used to draw curves in a horizontal plane with a playback in a vertical pad at different speeds.

7. Chapter seven demonstrates a robot writing technology inspired by DMP based segmentation and speech recognition. This involves the operator teaching the robot to write a Chinese character via verbal commands and robot playback via spatial and temporal generalization.

In addition, the novelties of this research project are as follows:

1. This thesis presents and designs a robot control system which integrates vision technology and RBFNN approximation technology to solve the problem of remote control for a robot manipulator. Finally, the method is employed on the Baxter robot platform.

2. Through the combination of DMP, GMR GMM and DTW algorithms, the robot has the ability to self-adapt and generalize spatially/temporally, which greatly reduces the time of teaching and training for robots, improves work efficiency. It is validated in the Baxter and KUKA iiwa robots.

3. According to different working needs, different teaching methods have been designed: for production of hazardous chemicals, the real-time remote teleoperation based human-robot skill transfer has been applied; for those producing operations with high-precision requirements, such as binding, cutting, the physical teaching by demonstration method can be applied.

4. The transfer of human motor skills is not just one-way based, such as Chinese writing tasks, human operators teach writing skills to robots, robots can still transfer the same skills to humans in the learning stage, such as children, disabled people.

5. Combining voice interaction with human-robot skill transfer to achieve verbal commands such that controlling the robot to complete specific tasks is more convenient.

8.2 Future Works

Although this thesis has achieved some research results in solving the problem of skill transfer between humans and robots, there are still some aspects to be improved:

1. This research work was limited by the inability to carry it out in an actual problem-solving situation even though there were three robots available. Hence, in the future, an actual industrial application or task based the research will be further carried out.

2. In addition, the dynamic model provided is not reliable outside the workspace where it was verified and tested. The controller developed in this work needs physical verification to determine how it will deal with actual environmental disturbances.

3. There exists a chattering phenomenon in extracting human motion information in visual technology. If the chattering signal is used as the input signal of the controller, it will not only reduce the control performance of the system, but also cause mechanical wear

and tear over time, which will lead to instability of the system. Therefore, it is necessary to solve the problem of input signal jitter.

4. As suggested previously, there is a need for research on the practical uses to which the different types of sensors and algorithms can be adopted. For example, the fuzzy logic can be used for the sensor fusion instead of the KF.

References

[1] Hyojoo Son, Changwan Kim, Hyoungkwan Kim, Seung Heon Han, and Moon Kyum Kim. Trend analysis of research and development on automation and robotics technology in the construction industry. *KSCE Journal of Civil Engineering*, 14(2):131–139, 2010.

[2] Atsushi Watanabe, Yoshiharu Nagatsuka, and Jun Mizuno. Robot teaching device, May 13 2008. US Patent 7,373,220.

[3] Xu Junyan, Zhang Peiren, and Cheng Junfei. Real-time trajectory tracking control of mobile robot based on backstepping time-varying state feedback and pid control method. *Electric Machines and Control*, 8(1):35–38, 2004.

[4] Zhiyong Yang, Jiang Wu, and Jiangping Mei. Motor-mechanism dynamic model based neural network optimized computed torque control of a high speed parallel manipulator. *Mechatronics*, 17(7):381–390, 2007.

[5] Clemens Amon, Ferdinand Fuhrmann, and Franz Graf. Evaluation of the spatial resolution accuracy of the face tracking system for kinect for windows v1 and v2. In *Proceedings of the 6th Congress of the Alps Adria Acoustics Association*, 2014.

[6] François Petitjean, Germain Forestier, Geoffrey I Webb, Ann E Nicholson, Yanping Chen, and Eamonn Keogh. Dynamic time warping averaging of time series allows faster and more accurate classification. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 470–479. IEEE, 2014.

[7] Hitesh Reddivari, Chenguang Yang, Zhaojie ju, Peidong Liang, Zhijun Li, and Bin Xu. Teleoperation control of baxter robot using body motion tracking. In *Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on*, pages 1–6, 2014.

[8] Chunxu Li, Chenguang Yang, Peidong Liang, Angelo Cangelosi, and Jian Wan. Development of kinect based teleoperation of nao robot. In *2016 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 133–138. IEEE, 2016.

[9] Chenguang Yang, Chao Zeng, Peidong Liang, Zhijun Li, Ruifeng Li, and Chun-Yi Su. Interface design of a physical human–robot interaction system for human impedance adaptive skill transfer. *IEEE Transactions on Automation Science and Engineering*, 15(1):329–340, 2018.

[10] Peidong Liang, Lianzheng Ge, Yihuan Liu, Lijun Zhao, Ruifeng Li, and Ke Wang. An augmented discrete-time approach for human-robot collaboration. *Discrete Dynamics in Nature and Society*, 2016, 2016.

[11] Chenguang Yang, Chao Zeng, Yang Cong, Ning Wang, and Min Wang. A learning framework of adaptive manipulative skills from human to robot. *IEEE Transactions on Industrial Informatics*, 2018.

[12] Jorge Solis, Simone Marcheschi, Antonio Frisoli, Carlo Alberto Avizzano, and Massimo Bergamasco. Reactive robot system using a haptic interface: an active interaction to transfer skills from the robot to unskilled persons. *Advanced robotics*, 21(3-4):267–291, 2007.

[13] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference*, pages 763–768, 2009.

[14] Hanxuan Yang, Ling Shao, Feng Zheng, Liang Wang, and Zhan Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011.

[15] Florian Faion, Simon Friedberger, Antonio Zea, and Uwe D Hanebeck. Intelligent sensor-scheduling for multi-kinect-tracking. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3993–3999. IEEE, 2012.

[16] Michael C Nechyba and Yangsheng Xu. Human skill transfer: neural networks as learners and teachers. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 3, pages 314–319. IEEE, 1995.

[17] Chunxu Li, Chenguang Yang, Jian Wan, Andy SK Annamalai, and Angelo Cangelosi. Teleoperation control of baxter robot using kalman filter-based sensor fusion. *Systems Science & Control Engineering*, 5(1):156–167, 2017.

[18] Jagdish L Raheja, Ankit Chaudhary, and Kunal Singal. Tracking of fingertips and centers of palm using kinect. In *Computational intelligence, modelling and simulation (CIMSiM), 2011 third international conference on*, pages 248–252. IEEE, 2011.

[19] Tao Gordon, Philippe Archambault, and Mindy Levin. Evaluation of kinect skeletal tracking in a virtual reality rehabilitation system for upper limb hemiparesis. In *Virtual Rehabilitation (ICVR), 2013 International Conference on*, pages 164–165. IEEE, 2013.

[20] Cha Zhang and Zhengyou Zhang. Calibration between depth and color sensors for commodity depth cameras. In *Computer Vision and Machine Learning with RGB-D Sensors*, pages 47–64. Springer, 2014.

[21] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*, 2010.

[22] John Bares, Martial Hebert, Takeo Kanade, Eric Krotkov, Tom Mitchell, Reid Simmons, and William Whittaker. Ambler: An autonomous rover for planetary exploration. *Computer*, 22(6):18–26, 1989.

[23] Anirban Mazumdar, Martin Lozano, Aaron Fittery, and H Harry Asada. A compact, maneuverable, underwater robot for direct inspection of nuclear power piping systems. In *2012 IEEE International Conference on Robotics and Automation*, pages 2818–2823. IEEE, 2012.

[24] WC Flannigan, Gabriel M Nelson, and Roger D Quinn. Locomotion controller for a crab-like robot. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 1, pages 152–156. IEEE, 1998.

[25] John E Bares and David S Wettergreen. Dante ii: Technical description, results, and lessons learned. *The International Journal of Robotics Research*, 18(7):621–649, 1999.

[26] Zhongcheng Wu, Fei Shen, Dezhang Xu, and Huaguo Zhou. Networked transducer interface module design and its application in bionic robotic sensing system. In *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*, volume 6, pages 4759–4762. IEEE, 2004.

[27] Davis Meike and Leonids Ribickis. Energy efficient use of robotics in the automobile industry. In *Advanced Robotics (ICAR), 2011 15th International Conference on*, pages 507–511. IEEE, 2011.

[28] Scott A Green, Mark Billinghurst, XiaoQi Chen, and J Geoffrey Chase. Human-robot collaboration: A literature review and augmented reality approach in design. *International journal of advanced robotic systems*, 5(1):1, 2008.

[29] Torgny Brogårdh. Present and future robot control developmentan industrial perspective. *Annual Reviews in Control*, 31(1):69–79, 2007.

[30] Chunxu Li, Chenguang Yang, and Cinzia Giannetti. Segmentation and generalization for writing skills transfer from humans to robots. *Cognitive Computation and Systems*, 2019.

[31] Chunxu Li, Chenguang Yang, Zhaojie Ju, and Andy SK Annamalai. An enhanced teaching interface for a robot using DMP and GMR. *International Journal of Intelligent Robotics and Applications*, pages 1–12, 2018.

[32] Chunxu Li, Chenguang Yang, Andy Annamalai, Qingsong Xu, and Shaoxiang Li. Development of writing task recombination technology based on dmp segmentation via verbal command for baxter robot. *Systems Science & Control Engineering*, 6(1):350–359, 2018.

[33] Chunxu Li, Chenguang Yang, Jian Wan, Andy Annamalai, and Angelo Cangelosi. Neural learning and kalman filtering enhanced teaching by demonstration for a baxter robot. In *2017 23rd International Conference on Automation and Computing (ICAC)*, pages 1–6. IEEE, 2017.

[34] Songde Ma. Integration technology for advanced manufacturing: R&d in china 863 program. In *IEEE International Conference on Integration Technology*, 2007.

[35] Richard Volpe and Pradeep Khosla. A theoretical and experimental investigation of explicit force control strategies for manipulators. *Automatic Control IEEE Transactions on*, 38(11):1634–1650, 1993.

[36] Loulin Huang, Shuzhi Sam Ge, and Tongheng Lee. Fuzzy unidirectional force control of constrained robotic manipulators. *Fuzzy Sets & Systems*, 134(1):135–146, 2003.

[37] Haifa Mehdi. Stiffness and impedance control using lyapunov theory for robot-aided rehabilitation. *International Journal of Social Robotics*, 4(1):107–119, 2012.

[38] Anders Blomdell, Isolde Dressler, Klas Nilsson, and Anders Robertsson. Flexible application development and high-performance motion control based on external sensing and reconfiguration of abb industrial robot controllers. In *Proc. ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications*, pages 62–66. Citeseer, 2010.

[39] Auke Jan Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4):642–653, 2008.

[40] Christine Connolly. A new integrated robot vision system from fanuc robotics. *Industrial Robot: An International Journal*, 34(2):103–106, 2007.

[41] Thanh Vo et al. Autonomous packaging robot. 2010.

[42] Lianzheng Ge, Jian Chen, and Ruifeng Li. Feedforward control based on fourier series trajectory fitting method for industrial robot. In *Control And Decision Conference (CCDC), 2017 29th Chinese*, pages 4890–4894. IEEE, 2017.

[43] Michael A Goodrich, Alan C Schultz, et al. Human–robot interaction: a survey. *Foundations and Trends® in Human–Computer Interaction*, 1(3):203–275, 2008.

[44] Christoph Bartneck and Jodi Forlizzi. Shaping human-robot interaction: understanding the social aspects of intelligent robotic products. In *CHI'04 Extended Abstracts on Human Factors in Computing Systems*, pages 1731–1732. ACM, 2004.

[45] Neville Hogan. Controlling impedance at the man/machine interface. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 1626–1631. IEEE, 1989.

[46] Russell H Taylor. A perspective on medical robotics. *Proceedings of the IEEE*, 94(9):1652–1664, 2006.

[47] Allison M Okamura. Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot: An International Journal*, 31(6):499–508, 2004.

[48] Yoshihiro Nakabo and Masatoshi Ishikawa. Visual impedance using 1 ms visual feedback system. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 3, pages 2333–2338. IEEE, 1998.

[49] Timo Oksanen, Jari Kostamo, Petro Tamminen, and Johannes Tiusanen. Robot competition as a teaching and learning platform. In *Proc. the 18th IFAC World Congress, Milano, August*, pages 5176–5181, 2011.

[50] Takayuki Yazawa, Masago Shiba, Hiroto Nakajima, and Masashi Fujiwara. Industrial robot, July 12 2016. US Patent 9,387,584.

[51] UG Student. Robot truck loading by unique matrix routine. *Development*, 2(5), 2015.

[52] Rainer Bischoff, Johannes Kurth, Günter Schreiber, Ralf Koeppe, Alin Albu-Schäffer, Alexander Beyer, Oliver Eiberger, Sami Haddadin, Andreas Stemmer, Gerhard Grunwald, et al. The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing. In *Robotics (ISR), 2010 41st international symposium on and 2010 6th German conference on robotics (ROBOTIK)*, pages 1–8. VDE, 2010.

[53] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *IEEE International Conference on Robotics & Automation*, 2017.

[54] Mark Pfeiffer, Michael Schaeuble, Juan Nieto, Roland Siegwart, and Cesar Cadena. From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. 2016.

[55] Tai Lei and Liu Ming. Deep-learning in mobile robotics - from perception to control systems: A survey on why and why not. 2016.

[56] Juergen Schmidhuber. Deep learning in neural networks: an overview. *Neural Netw*, 61:85–117, 2015.

[57] Vincent Franois-Lavet, Raphael Fonteneau, and Damien Ernst. How to discount deep reinforcement learning: Towards new dynamic strategies. *Computer Science*, 2015.

[58] Juan C. Caicedo and Svetlana Lazebnik. Active object localization with deep reinforcement learning. In *IEEE International Conference on Computer Vision*, 2015.

[59] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(1):1334–1373, 2015.

[60] Alessandro Giusti, Jerome Guzzi, Ciresan Dan, Fang Lin He, Juan Pablo Rodriguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jurgen Schmidhuber, and Gianni Di Caro. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics & Automation Letters*, 1(2):661–667, 2017.

[61] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with large-scale data collection. *International Journal of Robotics Research*, (10), 2016.

[62] Ke Li, Yue Lei Zhang, and Zhi Xiong Li. Application research of kalman filter and SVM applied to condition monitoring and fault diagnosis. In *Applied Mechanics and Materials*, volume 121, pages 268–272. Trans Tech Publ, 2012.

[63] Sridhar Seshagiri and Hassan K Khalil. Output feedback control of nonlinear systems using rbf neural networks. *IEEE Transactions on Neural Networks*, 11(1):69–79, 2000.

[64] Carlo Cecati, Janusz Kolbusz, Paweł Różycki, Pierluigi Siano, and Bogdan M Wilamowski. A novel rbf training algorithm for short-term electric load forecasting and comparative studies. *IEEE Transactions on industrial Electronics*, 62(10):6519–6529, 2015.

[65] Marko Robnik-Šikonja. Data generators for learning systems based on rbf networks. *IEEE transactions on neural networks and learning systems*, 27(5):926–938, 2016.

[66] Rui Yang, Poi Voon Er, Zidong Wang, and Kok Kiong Tan. An rbf neural network approach towards precision motion system with selective sensor fusion. *Neurocomputing*, 199:31–39, 2016.

[67] Yukiko Hoshino, Tsuyoshi Takagi, U Di Profio, and Masahiro Fujita. Behavior description and control using behavior module for personal robot. In *IEEE International Conference on Robotics & Automation*, 2004.

[68] Reza N Jazar. *Theory of applied robotics: kinematics, dynamics, and control*. Springer Science & Business Media, 2010.

[69] Philippe Poignet and Maxime Gautier. Nonlinear model predictive control of a robot manipulator. In *Advanced Motion Control, 2000. Proceedings. 6th International Workshop on*, pages 401–406. IEEE, 2000.

[70] Yong Feng, Xinghuo Yu, and Zhihong Man. Non-singular terminal sliding mode control of rigid manipulators. *Automatica*, 38(12):2159–2167, 2002.

[71] Mark Uebel, Ioannis Minis, and Kevin Cleary. Improved computed torque control for industrial robots. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 528–533. IEEE, 1992.

[72] J Adongo Ochier, Clementina D Mladenova, and Peter C Müller. An approach to automatic generation of dynamic equations of elastic joint manipulators in symbolic language. *Journal of Intelligent and Robotic Systems*, 14(2):199–218, 1995.

[73] Antal K Bejczy. Robot arm dynamics and control. *Jet Propulsion Laboratory Technical Memo*, pages 33–669, 1974.

[74] Antal K Bejczy and Richard Paul. Simplified robot arm dynamics for control. In *Decision and Control including the Symposium on Adaptive Processes, 1981 20th IEEE Conference on*, pages 261–262. IEEE, 1981.

[75] SaFid M Megahed. *Principles of robot modelling and simulation*. John Wiley & Sons, Inc., 1993.

[76] George Lee, Bowen Lee, and Ricahrd Nigam. Development of the generalized d'alembert equations of motion for mechanical manipulators. In *Decision and Control, 1983. The 22nd IEEE Conference on*, pages 1205–1210. IEEE, 1983.

[77] King Sun Fu, Rafael Gonzalez, and Lee George. Robotics control, sensing, vision, and intelligence. 1987.

[78] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2009.

[79] Philip Mckerrow. *Introduction to robotics*. Addison-Wesley Longman Publishing Co., Inc., 1991.

[80] Yongli Huang and Seiji Yasunobu. A general practical design method for fuzzy pid control from conventional pid control. In *Ninth IEEE International Conference on Fuzzy Systems. FUZZ-IEEE 2000 (Cat. No. 00CH37063)*, volume 2, pages 969–972. IEEE, 2000.

[81] Puren R Ouyang, Wen-Jun Zhang, and Fang-Xiang Wu. Nonlinear pd control for trajectory tracking with consideration of the design for control methodology. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 4126–4131. IEEE, 2002.

[82] Guilin Yang, I-Ming Chen, Weihai Chen, and Wei Lin. Kinematic design of a six-dof parallel-kinematics machine with decoupled-motion architecture. *IEEE transactions on robotics*, 20(5):876–887, 2004.

[83] Fathi Ghorbel and Ruvinda Gunawardana. A validation study of pd control of a closed-chain mechanical system. In *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, volume 2, pages 1998–2004. IEEE, 1997.

[84] Zhen Gao, Dan Zhang, and Yunjian Ge. Design optimization of a spatial six degree-of-freedom parallel manipulator based on artificial intelligence approaches. *Robotics and Computer-Integrated Manufacturing*, 26(2):180–189, 2010.

[85] Yunxing Su, Binye Duan, and Chunhong Zheng. Nonlinear pid control of a six-dof parallel manipulator. *IEE Proceedings-Control Theory and Applications*, 151(1):95–102, 2004.

[86] Warren E Dixon, Darren M Dawson, Erkan Zergeroglu, and Aman Behal. Adaptive tracking control of a wheeled mobile robot via an uncalibrated camera system. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(3):341–352, 2001.

[87] Se-Han Lee, Jae-Bok Song, Woo-Chun Choi, and Daehie Hong. Position control of a stewart platform using inverse dynamics control with approximate dynamics. *Mechatronics*, 13(6):605–619, 2003.

[88] Guangfeng Liu, Yulie Wu, Xuezhong Wu, Yingying Kuen, and Zhixing Li. Analysis and control of redundant parallel manipulators. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, pages 3748–3754. IEEE, 2001.

[89] Hui Cheng, Yiu-Kuen Yiu, and Zexiang Li. Dynamics and control of redundantly actuated parallel manipulators. *IEEE/ASME Transactions on mechatronics*, 8(4):483–491, 2003.

[90] Guanfeng Liu and Zexiang Li. A unified geometric approach to modeling and control of constrained mechanical systems. *IEEE Transactions on robotics and automation*, 18(4):574–587, 2002.

[91] Farhad Aghili. A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: applications to control and simulation. *IEEE Transactions on Robotics*, 21(5):834–849, 2005.

[92] Hag Seong Kim, Young Man Cho, and Kyo-II Lee. Robust nonlinear task space control for 6 dof parallel manipulator. *Automatica*, 41(9):1591–1600, 2005.

[93] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jér Ome Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, and Bruno Maisonnier. Mechatronic design of nao humanoid. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 769–774. IEEE, 2009.

[94] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jerome Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, and Bruno Maisonnier. The nao humanoid: a combination of performance and affordability. *CoRR abs/0807.3223*, 2008.

[95] Jason Kulk and JS Welsh. A low power walk for the nao robot. *preprint, available at*, 2008.

[96] Chenguang Yang, Huaiwei Wu, Zhijun Li, Wei He, Ning Wang, and Chun-Yi Su. Mind control of a robotic arm with visual fusion technology. *IEEE Transactions on Industrial Informatics*, 2017.

[97] Andrew D Wilson, Jarvis A Schultz, Alex R Ansari, and Todd D Murphey. Dynamic task execution using active parameter identification with the baxter research robot. *IEEE Transactions on Automation Science and Engineering*, 14(1):391–397, 2017.

[98] Cliff Fitzgerald. Developing baxter. In *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, pages 1–6. IEEE, 2013.

[99] Steve Cousins. Ros on the pr2 [ros topics]. *IEEE Robotics & Automation Magazine*, 17(3):23–25, 2010.

[100] Günter Schreiber, Andreas Stemmer, and Rainer Bischoff. The fast research interface for the kuka lightweight robot. In *IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010)*, pages 15–21. Citeseer, 2010.

[101] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.

[102] Jarrett Webb and James Ashley. *Beginning Kinect Programming with the Microsoft Kinect SDK*. Apress, 2012.

[103] James Altman. Taming the dragon effective use of dragon naturallyspeaking speech recognition software as an avenue to universal access. *Writing & Pedagogy*, 5(2):333–348, 2014.

[104] Stefan Schaal, Jan Peters, Jun Nakanishi, and Auke Ijspeert. Learning movement primitives. In *Robotics research, the eleventh international symposium, Springer*, pages 561–572, 2005.

[105] Stefan Schaal. Dynamic movement primitives-a framework for motor control in humans and humanoid robotics. In *Adaptive motion of animals and machines*, pages 261–280. Springer, 2006.

[106] Affan Pervez and Dongheui Lee. Learning task-parameterized dynamic movement primitives using mixture of gmms. *Intelligent Service Robotics*, 11(1):61–78, 2018.

[107] Hongyu Ren, Da Xu, and Wenxin Li. Modeling the uncertainty in finger-vein authentication by the gaussian mixture model. In *Proceedings of the 2017 4th International Conference on Biomedical and Bioinformatics Engineering*, pages 39–42. ACM, 2017.

[108] Narjes Davari, Asghar Gholami, and Mohammad Shabani. Multirate adaptive Kalman filter for marine integrated navigation system. *The Journal of Navigation*, pages 1–20, 2016.

[109] Simo Särkkä and Arno Solin. On continuous-discrete cubature Kalman filtering. *IFAC Proceedings Volumes*, 45(16):1221–1226, 2012.

[110] Chenguang Yang, Junshen Chen, and Fei Chen. Neural learning enhanced teleoperation control of baxter robot using IMU based motion capture. In *Automation and Computing (ICAC), 2016 22nd IEEE International Conference on*, pages 389–394, 2016.

[111] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35(4):399–458, 2003.

[112] Frank L Lewis, Aydin Yesildirek, and Kai Liu. Multilayer neural-net robot controller with guaranteed tracking performance. *Neural Networks, IEEE Transactions on*, 7(2):388–399, 1996.

[113] Masanobu Yamamoto and Kazutada Koshikawa. Human motion analysis based on a robot arm model. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 664–665. IEEE, 1991.

[114] Alina Ninett Panfir, Răzvan Gabriel Boboc, and Gheorghe Leonte Mogan. Nao robots collaboration for object manipulation. In *Applied Mechanics and Materials*, volume 332, pages 218–223. Trans Tech Publications Ltd, 2013.

[115] Jason Kulk and James S Welsh. Evaluation of walk optimisation techniques for the nao robot. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 306–311. IEEE, 2011.

[116] Athanasia Louloudi, Ahmed Mosallam, Naresh Marturi, Pieter Janse, and Victor Hernandez. Integration of the humanoid robot nao inside a smart home: A case study. In *Proceedings of the Swedish AI Society Workshop (SAIS). Linköping Electronic Conference Proceedings*, volume 48, pages 35–44, 2010.

[117] Elena Garcia, Maria Antonia Jimenez, Pablo Gonzalez De Santos, and Manuel Armada. The evolution of robotics research. *IEEE Robotics & Automation Magazine*, 14(1):90–103, 2007.

[118] Kerstin Dautenhahn. Socially intelligent robots: dimensions of human–robot interaction. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1480):679–704, 2007.

[119] Chenguang Yang, Sai Chang, Peidong Liang, Zhijun Li, and Chun-Yi Su. Teleoperated robot writing using EMG signals. In *Information and Automation, 2015 IEEE International Conference on*, pages 2264–2269, 2015.

[120] Chenguang Yang, Junshen Chen, Zhijun Li, Wei He, and Chun-Yi Su. Development of a physiological signals enhanced teleoperation strategy. In *Information and Automation, 2015 IEEE International Conference on*, pages 13–19, 2015.

[121] Chen Li, Hongbin Ma, Chenguang Yang, and Menyin 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)*, pages 1469–1474, 2014.

[122] YuKang Liu, YuMing Zhang, Bo Fu, and Ruigang Yang. Predictive control for robot arm teleoperation. In *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, pages 3693–3698, 2013.

[123] Inseong Jo, Younkyu Park, and Joonbum Bae. A teleoperation system with an exoskeleton interface. In *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1649–1654, 2013.

[124] Zhangfeng Ju, Chenguang Yang, Zhijun Li, Long Cheng, and Hongbin Ma. Teleoperation of humanoid baxter robot using haptic feedback. In *Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on*, pages 1–6. IEEE, 2014.

[125] Fuwen Yang, Zidong Wang, and YS Hung. Robust kalman filtering for discrete time-varying uncertain systems with multiplicative noises. *IEEE Transactions on Automatic Control*, 47(7):1179–1183, 2002.

[126] Chenguang Yang, Shuzhi Sam Ge, Cheng Xiang, Tianyou Chai, and Tong Heng Lee. Output feedback NN control for two classes of discrete-time systems with unknown control directions in a unified approach. *IEEE Transactions on Neural Networks*, 19(11):1873–1886, 2008.

[127] Yan-Jun Liu, CL Philip Chen, Guo-Xing Wen, and Shaocheng Tong. Adaptive neural output feedback tracking control for a class of uncertain discrete-time nonlinear systems. *IEEE Transactions on Neural Networks*, 22(7):1162–1167, 2011.

[128] Weisheng Chen and Licheng Jiao. Adaptive tracking for periodically time-varying and nonlinearly parameterized systems using multilayer neural networks. *IEEE Transactions on Neural Networks*, 21(2):345–351, 2010.

[129] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *The International Journal of Robotics Research*, 31(5):647–663, 2012.

[130] Adi Sucipto, Agung Harsoyo, and Pranoto Hidaya Rusmin. Implementation of gesture recognition on aquarium application. In *System Engineering and Technology (ICSET), 2012 IEEE International Conference on*, pages 1–4, 2012.

[131] Maged N Kamel Boulos, Bryan J Blanchard, Cory Walker, Julio Montero, Aalap Tripathy, and Ricardo Gutierrez-Osuna. Web GIS in practice x: a microsoft kinect natural user interface for google earth navigation. *International journal of health geographics*, 10(1):1, 2011.

[132] Norman Villaroman, Dale Rowe, and Bret Swan. Teaching natural user interaction using OpenNI and the Microsoft Kinect sensor. In *Proceedings of the 2011 conference on Information technology education*, pages 227–232. The Journal of the ACM, 2011.

[133] Jordi Bolíbar. Kinect audio-runner: Audio feedback for improving performance in long-distance running. *Master of Science Thesis, KTH Royal Institute of Technology, Stockholm, Sweden*, 2012.

[134] Marco Ronchetti and Mattia Avancini. Using kinect to emulate an interactive whiteboard. *MS in Computer Science, University of Trento*, 2011.

[135] Jared St Jean. *Kinect Hacks: Tips & Tools for Motion and Pattern Detection*. "O'Reilly Media, Inc.", 2012.

[136] Enrique Ramos. Processing. In *Arduino and Kinect Projects*, pages 35–60. Springer, 2012.

[137] Ryan Calo. Open robotics. *Maryland Law Review*, 70(3), 2011.

[138] Craig Partridge and Stephen Pink. A faster udp (user datagram protocol). *IEEE/ACM Transactions on Networking*, 1(4):429–440, 1993.

[139] Yuguang Fang, Kenneth A Loparo, and Xiangbo Feng. Inequalities for the trace of matrix product. *IEEE Transactions on Automatic Control*, 39(12):2489–2490, 1994.

[140] Cheng S Chin and Keng M Yue. Application of an intelligent table-top vacuum robot cleaner in mechatronics system design education. *Journal of Robotics and Mechatronics*, 23(5):645, 2011.

[141] Sonia Chernova and Manuela Veloso. Teaching multi-robot coordination using demonstration of communication and state sharing. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, pages 1183–1186. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[142] Sylvain Calinon, Danilo Bruno, Milad S Malekzadeh, Thrishantha Nanayakkara, and Darwin G Caldwell. Human–robot skills transfer interfaces for a flexible surgical robot. *Computer methods and programs in biomedicine*, 116(2):81–96, 2014.

[143] Cheng Chin and Keng Yue. Vertical stream curricula integration of problem-based learning using an autonomous vacuum robot in a mechatronics course. *European Journal of Engineering Education*, 36(5):485–504, 2011.

[144] Cheng Siong Chin, MW Shing Lau, Eicher Low, and GG Lee Seet. A robust controller design method and stability analysis of an underactuated underwater vehicle. *International Journal of Applied Mathematics and Computer Science*, 16:345–356, 2006.

[145] Yonghong Huang, Kevin B Englehart, Bernard Hudgins, and Adrian DC Chan. A gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses. *IEEE Transactions on Biomedical Engineering*, 52(11):1801–1811, 2005.

[146] Ming-Hsuan Yang and Narendra Ahuja. Gaussian mixture model for human skin color and its applications in image and video databases. In *Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 458–467. International Society for Optics and Photonics, 1998.

[147] Petar Kormushev, Sylvain Calinon, and Darwin G Caldwell. Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. *Advanced Robotics*, 25(5):581–603, 2011.

[148] Freek Stulp, Gennaro Raiola, Antoine Hoarau, Serena Ivaldi, and Olivier Sigaud. Learning compact parameterized skills with a single regression. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pages 417–422. IEEE, 2013.

[149] Micha Hersch, Florent Guenter, Sylvain Calinon, and Aude G Billard. Learning dynamical system modulation for constrained reaching tasks. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 444–449. IEEE, 2006.

[150] Abdullah Mueen and Eamonn Keogh. Extracting optimal performance from dynamic time warping. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2129–2130. ACM, 2016.

[151] Neville Hogan and Dagmar Sternad. Dynamic primitives of motor behavior. *Biological cybernetics*, pages 1–13, 2012.

[152] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

[153] Jianjun Yu, Yusen Men, Xiaogang Ruan, and Shaoqiong Zhao. The research and implementation of behavior imitation system about nao robot based on kinect. *CAAI Transactions on Intelligent Systems*, 2:006, 2016.

[154] Kenneth P Burnham and David R Anderson. Multimodel inference: understanding aic and bic in model selection. *Sociological methods & research*, 33(2):261–304, 2004.

[155] Lingyun Hu, Changjiu Zhou, and Zengqi Sun. Estimating biped gait using spline-based probability distribution function with q-learning. *IEEE Transactions on Industrial Electronics*, 55(3):1444–1452, 2008.

[156] Satoshi Kurosawa, Hidehisa Nakayama, Nei Kato, Abbas Jamalipour, and Yoshiaki Nemoto. Detecting blackhole attack on aodv-based mobile ad hoc networks by dynamic learning method. *IJ Network Security*, 5(3):338–346, 2007.

[157] Simon Thompson, Takehiro Horiuchi, and Satoshi Kagami. A probabilistic model of human motion and navigation intent for mobile robot path planning. In *2009 4th International Conference on Autonomous Robots and Agents*, pages 663–668. IEEE, 2009.

[158] Rui Huang, Chenguang Yang, Fan Yang, and Zhijun Li. Robot hand learning from multiple demonstrations using dynamics motor primitives. In *Advanced Mechatronic Systems (ICAMechS), 2017 IEEE International Conference*, pages 1–6, 2017.

[159] Andrej Gams, Auke J Ijspeert, Stefan Schaal, and Jadran Lenarčič. On-line learning and modulation of periodic movements with nonlinear dynamical systems. *Autonomous robots*, 27(1):3–23, 2009.

[160] Manuel Muhlig, Michael Gienger, Sven Hellbach, Jochen J Steil, and Christian Goerick. Task-level imitation learning using variance-based movement optimization. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference*, pages 1177–1184, 2009.

[161] Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[162] Wei Wang, Baoju Zhang, Dan Wang, Yu Jiang, Shan Qin, and Lei Xue. Anomaly detection based on probability density function with kullback–leibler divergence. *Signal Processing*, 126:12–17, 2016.

[163] Michael S Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2(1):1–19, 2006.

[164] Shunyi Yao, Ying Wen, and Yue Lu. Hog based two-directional dynamic time warping for handwritten word spotting. In *Document Analysis and Recognition (ICDAR), 2015 13th IEEE International Conference*, pages 161–165, 2015.

[165] Hector De la Torre Gutierrez and DT Pham. Estimation and generation of training patterns for control chart pattern recognition. *Computers & Industrial Engineering*, 95:72–82, 2016.

[166] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[167] Yaodong Zhang, Kiarash Adl, and James Glass. Fast spoken query detection using lower-bound dynamic time warping on graphical processing units. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference*, pages 5173–5176, 2012.

[168] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.

[169] Mingzhi Li, Zhiqiang Ma, Yong Shan, and Xiaoyan Zhang. Adaptive background update based on gaussian mixture model under complex condition. *Jisuanji Yingyong/ Journal of Computer Applications*, 31(7):1831–1834, 2011.

[170] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.

[171] Sylvain Calinon and Aude Billard. Statistical learning by imitation of competing constraints in joint space and task space. *Advanced Robotics*, 23(15):2059–2076, 2009.