

**A NEURAL NETWORK BASED ADAPTIVE ARM MOTION  
GENERATION FOR ASSISTIVE HUMANOID ROBOTS**

人間型介助ロボットのニューラルネットによる適応アーム動作生成

**ZULKIFLI MOHAMED**

**Dissertation submitted in partial fulfilment for the degree of**

Doctor of Philosophy

**Graduate School of Science and Engineering**

**University of Toyama**

**February 2014**

## **Acknowledgements**

*Syukur Alhamdulillah.* It would not have been possible to write this doctoral thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

First and foremost I would like to express my deepest gratitude to my PhD advisors, Professor Genci Capi for excellent guidance, caring, patience and providing me with an excellent atmosphere for doing research. He has been very helpful in anyway, supportive and kind since the first day I arrived in Toyama. His endless support not only in my research but also to my family and my life here in Toyama. Without his guidance, it would be almost impossible for me to complete my PhD.

I would like to thank to all Intelligent Robotics Laboratory members especially to Toda sensei for having me as one of his student, Kitani sensei for helping me on translating all the paper works and for sharing his knowledge and experience, Mano for being my only roommate for the past three years and it would have been a lonely lab without him. Many thanks to Ishikawa and Takeuchi for helping me on the mobile platform and also Shirakawa-san from Mechanical Engineering Workshop for helping me on the machining part of the robot.

Thanks to all my Malaysian friends in Toyama for the best experience ever, for helping and guiding me in so many things, to my officemate in the Faculty of Mechanical Engineering UiTM Shah Alam, especially the top management lead by the Dean, Professor Ahmed Jaffar and not forgotten Azli, Azhan, Alias, Afzan, Halim, Bidin, Hafiz, Man, Azzeim, Skano, Arai, Fazli, Che We, Liana and Pak Lah.

I would also like to thank to the Malaysian Government specifically Ministry of Education Malaysia and Universiti Teknologi MARA for giving me this oppor-

tunity to do my PhD here in University of Toyama by providing financial support to me and my family for the past three and a half years.

A special thanks to my beloved wife, sons and daughter, my mom, brothers and sister and to my late father for endless support and doa. My doa will always for all of you.

*This dissertation is lovingly dedicated to my wife, Ida Rohayu, for always being there for me.*

*Her support, encouragement, doa and constant love have sustained me throughout my life.*

## **Abstract**

In the recent years the number of ageing population has been increasing. In the coming decades, this scenario will lead to high demand for health care and financial resources. The ability to do house hold chores such as cleaning, picking the trash or taking a bottle of water from the kitchen becomes much harder for the elderly. One of the solutions is by having personal assistive robots to help with the daily activities and task.

As humanoid robots are expected to operate in human environments they are expected to perform a wide range of tasks. Therefore, the robot arm motion must be generated based on the specific task. In this thesis, we propose an optimal arm motion generation satisfying single and multiple criteria. In our method, we evolved neural controllers that generate the humanoid robot arm motion in dynamic environment optimizing three different objective functions: minimum time, minimum distance and minimum acceleration. An advantage of proposed method is that in a single ran of MOEA multiple neural controllers are generated. In addition, the same neural controller can be employed to generate the robot motion for a wide range of initial and goal positions. The robot motion generation in dynamic environments is also considered.

A new mobile humanoid robot system has been developed in our lab to test the performance of the proposed method. The robot consists of two main parts, the upper body for object manipulation and the mobile platform for robot navigation. The humanoid robot has ten degrees of freedom in both arms, two degrees of freedom head and two grippers. The robot is equipped with two cameras and two laser range finders for assisting the robot while navigating, localization and object recognition.

We tested the evolved neural controller in a simulated and on the real robot. The results show good performance. Although the results in simulated environment and on the real robot have some differences, the robot is capable of maintaining its trajectory and completes the task successfully.

## Abstract (Japanese)

近年、世界的に高齢化が進んでおり、高齢者に対する身体的ないし経済的な補助が急務となっている。若年者は、清掃やゴミ出し、物の運搬といった家事を容易に行うことができるが、高齢者は、身体能力の低下のためにこれらの動作を行うことが困難となる。よって、高齢者を日常的に補助するための、個人用介助ロボットの需要が高まっている。

介助ロボットは、一つの動作を繰り返し行うものよりも、様々な動作を行うことが期待されるが、ロボットの腕部の動作は特定の動作を基準として生成されることが好ましい。本論文では、ロボットの腕部動作生成のための最適化手法について検討した。本論文における提案手法は、最少時間と距離、加速度の 3 つの目的関数を最適化することで、動的環境におけるロボットの腕部動作を生成することができる。また、1 つの多目的進化型アルゴリズム (Multi-objective Evolutionary Algorithm: MOEA) によって複数のニューラルコントローラを生成することが可能である。加えて、ニューラルコントローラはロボットの動作開始地点と終了地点を変更しても、利用することができる。

本提案手法を実装するための人型ロボットを構築した。ロボットは、物体を操作するための上半身と移動のための下半身の 2 つから構成されている。上半身の自由度は、2 自由度の頭部と把持部を含む 10 自由度である。また、センサとしてステレオカメラと 2 つのレーザレンジファインダが設置されており、これらを用いて、物体認識と定位、移動を行うことができる。

提案手法を検証するために、ニューラルコントローラをシミュレーション環境で進化させ、ロボットに実装した。結果として、ロボットは良好に腕部動作を生成することができた。シミュレーション環境と実環境では差があるものの、ロボットはシミュレーション環境で獲得した腕部の軌道を保ちつつ、良好にタスクを実行することができた。

## Table of Contents

Acknowledgements .....	i
Abstract .....	iv
Abstract (Japanese).....	vi
Table of Contents.....	viii
List of Tables .....	xi
List of Figures.....	xii
1 INTRODUCTION.....	1
1.1 Background.....	2
1.2 Research Goal.....	3
1.3 Achievements .....	3
1.4 Thesis Outline.....	3
2 LITERATURE REVIEW .....	5
2.1 Assistive Humanoid Robots .....	8
2.1.1 System Integration.....	11
2.2 Arm Motion Generation .....	13
2.2.1 Arm Motion as an Optimization Problem .....	15
3 MOBILE HUMANOID ROBOT PLATFORM .....	18
3.1 Kinematics Analysis .....	19
3.1.1 Denavit-Hartenberg (DH) Parameters .....	19
3.1.2 Inverse Kinematics .....	21
3.2 Mechatronics Design & Hardware Architecture.....	23
3.2.1 Upper Body .....	23
3.2.2 Mobile Platform.....	26
3.2.3 Electronic Components .....	27
3.2.3.1 DC Motors .....	27
3.2.3.2 AC Motors .....	27

3.2.3.3	Servo Motors .....	28
3.2.3.4	Laser Range Finder (LRF).....	29
3.2.3.5	Potentiometers .....	31
3.2.4	Software Architecture.....	31
3.2.4.1	MATLAB Simulation .....	31
3.2.5	System Integration.....	32
3.3	Kinesiology of the Robot Motion.....	35
4	EVOLUTION OF NEURAL CONTROLLERS .....	38
4.1	Problem Formulation.....	38
4.1.1	Robot Arm Tasks Selection .....	39
4.1.2	Arm Motion Generation .....	42
4.1.3	Proposed Method.....	43
4.2	Neural Networks.....	44
4.3	Genetic Algorithms (GA) .....	46
4.3.1	Multi-Objective GA.....	47
4.3.2	Objective Functions.....	47
4.3.2.1	Arm Motion Objective Functions .....	48
4.3.2.2	Minimum Execution Time (MT) .....	48
4.3.2.3	Minimum Distance (MD) .....	49
4.3.2.4	Minimum Acceleration (MA).....	50
4.3.2.5	Minimum Angular Acceleration (MAA) .....	51
4.3.3	Robot Navigation.....	51
4.3.3.1	Environment 1.....	51
4.3.3.2	Environment 2.....	52
4.3.3.3	Environment 3.....	53
4.4	Arm Motion Generation in Dynamic Environment.....	54
5	SIMULATION AND EXPERIMENTAL RESULTS .....	57
5.1	Single Objective Optimization .....	57

5.1.1 Simulation Results.....	58
5.1.2 Experimental Results.....	61
5.2 Multi Objective Optimization.....	67
5.2.1 Two Objective Optimization .....	67
5.2.1.1 Simulation Results: Minimum Time & Minimum Distance: $(f_1-f_2)$ .....	67
5.2.1.2 Simulation Results: Minimum Time & Minimum Acceleration: $(f_1-f_3)$ ....	69
5.2.1.3 Simulation Results: Minimum Distance & Minimum Acceleration: $(f_2-f_3)$	71
5.2.1.4 Experimental Results: Two Objective Optimization.....	73
5.2.2 Three Objective Optimization .....	78
5.3 Neural Controllers Performance and Obstacle Avoidance.....	83
5.4 Robot Navigation.....	94
6 CONCLUSION AND DISCUSSION .....	105
6.1 Future Work .....	106
REFERENCES .....	108
APPENDIX A:.....	117
APPENDIX B:.....	119
APPENDIX C:.....	120

## List of Tables

Table 3- 1 D-H Parameters.....	20
Table 4-1 Summary of genetic algorithm parameter.....	46
Table 5-1 Simulation parameters.....	86

## List of Figures

Figure 1-1 Changes of population pyramid (a) World (b) Japan (c) Malaysia. ....	6
Figure 1-2 Age structure population by country.....	7
Figure 1-3 Developed mobile humanoid robot (a) TWENDY-ONE (b) ARMAR III (c) RIBA (d) Dynamaid (e) Snackbot (f) PR2. ....	7
Figure 3-1 Coordinate frame of the robot upper body.....	19
Figure 3- 2 Inverse kinematics analysis of the robot hand.....	22
Figure 3-3 Right hand design of the mobile humanoid robot. ....	24
Figure 3-4 Left hand design of the mobile humanoid robot. ....	25
Figure 3-5 Mobile platform.....	26
Figure 3-6 DC Motors and motor drivers. ....	26
Figure 3-7 AC Motors. ....	27
Figure 3-8 Servo motors.....	28
Figure 3-9 Laser range finder. ....	28
Figure 3-10 (a) Obstacle detection via LRF1 (mobile platform) (b) Spray can detection via LRF2 at table height (upper body). ....	30
Figure 3-11 Potentiometers. ....	31
Figure 3-12 (a) Robot arm simulator (b) GUI.....	33
Figure 3-13 Developed mobile humanoid robot. ....	34
Figure 3- 14 System Integration.....	35
Figure 3- 15 Robot head movements. ....	36
Figure 3- 16 Robot arm movements.....	36
Figure 3-17 Robot hand flexion and extension movements.....	37
Figure 3-18 Robot hand supination, semiprone and pronation movements.....	37

Figure 4-1 Task 1: Placing a bottle on the table.....	39
Figure 4-2 Experimental setup for reaching the table.....	40
Figure 4-3 Task 2: Reaching a spray can on the table and picking motion to a holding position.....	41
Figure 4- 4 Task 3: Reaching a spray can on the table while avoiding obstacle.....	41
Figure 4-5 FFNN for mobile platform. ....	45
Figure 4-6 FFNN for robot arm. ....	45
Figure 4-7 GA Cycle. ....	48
Figure 4-8 Robot navigation in environment 1. ....	52
Figure 4-9 Robot navigation in environment 2. ....	53
Figure 4-10 (a) Robot navigation in environment 3 (b) Colour landmark.....	54
Figure 4-11 Obstacle detection regions for different obstacle shape. ....	55
Figure 5-1 Experiment 1: Placing a bottle on the table.....	58
Figure 5-2 Execution time comparison for each objective function. ....	59
Figure 5-3 Total trajectory comparison for each objective function. ....	60
Figure 5-4 Total velocity comparison for each objective function. ....	60
Figure 5-5 Comparison of robot hand (a) Trajectory (b) Velocity profile. ....	62
Figure 5-6 Video capture of the experiment.....	63
Figure 5-7 Task execution times for robot hand (a) Without bottle (b) With bottle...	64
Figure 5-8 Execution time comparison for simulation, without bottle and with bottle for (a) MT (b) MD (c) MA (d) MAA.....	65
Figure 5-9 Execution time comparison for each objective functions in $x$ , $y$ and $z$ -axis (a) Simulation (b) With load. ....	66
Figure 5-10 Pareto fronts of MT-MD objective functions.....	69

Figure 5-11 Robot arm motion for NN1, NN2, NN3 and NN4 neural controllers of MT-MD objective function.....	69
Figure 5-12 Pareto fronts of MT-MA objective functions.....	70
Figure 5-13 Robot arm motion for NN1, NN2, NN3 and NN4 neural controllers of MT-MA objective function.....	71
Figure 5-14 Pareto fronts of MD-MA objective functions.....	72
Figure 5-15 Robot arm motion for NN1, NN2, NN3 and NN4 neural controllers of MD-MA objective function. ....	72
Figure 5-16 Robot arm motion for $f_1, f_2$ and $f_3$ neural controllers of single objective function neural controllers. ....	73
Figure 5-17 Robot arm joint trajectories for (a) $\theta_1$ (b) $\theta_2$ (c) $\theta_3$ .....	75
Figure 5-18 MT-MA joint trajectories for (a) $\theta_1$ (b) $\theta_2$ (c) $\theta_3$ . ....	76
Figure 5-19 Pareto comparison of two and three objective functions optimization for (a) MT-MD (b) MT-MA (c) MD-MA.....	77
Figure 5-20 Pareto front of MT-MD-MA objective functions optimization.....	78
Figure 5-21 Pareto front of MT-MD-MA objective functions optimization (a) $f_2$ - $f_1$ view (b) $f_3$ - $f_1$ view (c) $f_3$ - $f_2$ view.....	79
Figure 5-22 Robot hand motion adapting NN1, NN2 and NN3 solutions.....	81
Figure 5-23 Video capture of robot hand motion adapting NN1, NN2 and NN3 solutions.....	81
Figure 5-24 Joint angular displacement comparison between simulation and experiment of the robot arm (a) $\theta_1$ (b) $\theta_2$ (c) $\theta_3$ . ....	82
Figure 5-25 Selected neural controllers for right hand (a) Picking motion (b) Holding motion. ....	84

Figure 5-26 Selected neural controllers for left hand (a) Picking motion (b) Holding motion. ....	85
Figure 5-27 Simulation results for random goal position 1 and 2 for environment 1. ....	86
Figure 5-28 Simulation results for random goal position 1 and 2 for environment 2. ....	88
Figure 5-29 Simulation results for random goal position 1 and 2 for environment 3. ....	88
Figure 5-30 Mobile humanoid robot moving toward the table. ....	89
Figure 5-31 Experimental results for random goal position 1 and 2 (a) Environment 1 (b) Environment 2 (c) Environment 3. ....	90
Figure 5-32 Obstacle avoidance results for (a) Simulation (b) Real robot. ....	92
Figure 5-33 Trajectory comparison between simulation results and the real robot for (a) <i>x</i> -axis (b) <i>y</i> -axis and (c) <i>z</i> -axis. ....	93
Figure 5-34 Robot navigation (a) Simulation (b) Real environments. ....	94
Figure 5-35 Robot navigation in environment 1 (a) Simulation (b) Experiment. ....	96
Figure 5-36 Robot navigation in environment 2 (a) Simulation (b) Experiment. ....	97
Figure 5-37 Robot navigation in environment 3 (a) Simulation (b) Experiment. ....	98
Figure 5-38 Robot hand simulation. ....	100
Figure 5-39 Mobile robot approaching the table. ....	100
Figure 5-40 Spray can detection via laser range sensor and reaching motion of the robot hand. ....	101
Figure 5-41 Reaching and picking the spray can motion of the robot hand. ....	101
Figure 5-42 Comparison of the robot hand motion between simulation and experiment (GP1) for (a) <i>x</i> -axis (b) <i>y</i> -axis (c) <i>z</i> -axis. ....	103

Figure 5-43 Comparison of the robot hand motion between simulation and  
experiment (GP2) for (a)  $x$ -axis (b)  $y$ -axis (c)  $z$ -axis..... 104

---

# CHAPTER 1

---

## 1 INTRODUCTION

The increasing number of ageing population becomes one of the main factor for research in the field of mobile humanoid robot is still relevant and important as it can potentially be applied and used in the hospital, rehabilitation centre, elderly people centre and or in our own home. The main problem with these domestic robots is the price is too high and unaffordable by normal people. This is the reason, researches in this area are still going on and numbers of mobile humanoid robots are still being built in recent years. The main focus is to come out with the best robot that can perform domestic task, has high mobility, safe to co-exist with the human and affordable. A lot of factors and constraints have to be considered in developing a mobile humanoid robot such as the safety features, robustness, flexibility, manoeuvrability, ability to understand human instructions, ability to process informations, the task performance speed and the most important aspect is the cost of building it.

Basically, a mobile humanoid robot consists of three main parts, the main body, mobile platform, vision and sensory system. Each of these parts has their own function and contribution to the developed mobile humanoid robot and need to be integrated in order to have a complete and functional system. The main body is designed to perform the manipulation task such as picking, placing, holding, pushing and moving and object. The design should consider both mechanical and electrical parts such as the

kinematics analysis, number of degrees of freedom, mechanical and electronic components and their positions, safety features and the limitation of the robot motion.

The vision and sensory system are very important in order to guide the robot while navigating in the environment and avoiding obstacles, to determine the object or obstacle size, shape and position for the robot manipulation and also for receiving instruction from human. The last part is the mobile platform and it becomes the base for the upper body and other components such as battery and PC. The main function of the mobile platform is to navigate around the environment and moving to the desired location while avoiding obstacles.

Another important issue in developing a mobile humanoid robot is to have an optimized arm motion when performing a specific task. A human can easily move their hand to perform a task with high speed and accuracy but for a robot to do the same task a lot of considerations need to be made. Three basic criteria in arm motions should be considered; the time, distance and speed which are very important in order to have a good performance in task execution.

## **1.1 Background**

In performing domestic task such as cleaning, picking and carrying food or household item from one room to another, a mobile humanoid robot are required to have high mobility, and ability to manipulate object in optimized manner. There are numbers of constraints and consideration need to be made in order to perform these tasks such as unknown environment, obstacles avoidance, object recognition and manipulation, the execution time, safety features and understanding human instruction.

In this thesis we investigate the performance of the proposed optimal neural controller in simulation environment and we verified the performance on the newly developed mobile humanoid robot. Single and multi-objective evolutionary algorithms are proposed to generate optimal neural controllers for the robot arm motion to perform simple domestic task. Arm motion generation in dynamic environment is also considered in this thesis. The performance of evolved neural controllers is tested in the mobile humanoid robot developed in our lab.

## **1.2 Research Goal**

The main goal of this thesis is to develop an intelligent humanoid robot able to perform several tasks. The humanoid robot selects the appropriate neural controller from a set of pre-evolved neural controllers for the specific tasks. Another objective is to apply multi-objective GA for simultaneous evolution of neural controllers.

## **1.3 Achievement**

We have developed a humanoid robot able to generate the arm motion satisfying several objective functions. The proposed optimized neural controllers show good results both in simulated and real robot implementation. In addition, arm motion generation in dynamic environment where obstacles are present is also considered. The developed mobile humanoid robot navigates in the environment to reach the target location and performs the task.

## **1.4 Thesis Outline**

The thesis is organized as follows; in Chapter 2, the literature review and related works are explained in details. The previous works and the current development of mobile humanoid robot research are discussed in this chapter. Chapter 3 introduces

our newly developed mobile humanoid robot. The kinematics analysis, detail design process, software architecture, mechanical and electronic components and the kinesiology of the robot motion are explained. The evolution of neural controllers and problem formulation are discussed in Chapter 4. Chapter 5 presents the results and performance verifications on simulated environment and on the real robot. All the findings are discussed in details. Chapter 6 concludes the thesis and discusses the future work that can be drawn on this research.

---

# CHAPTER 2

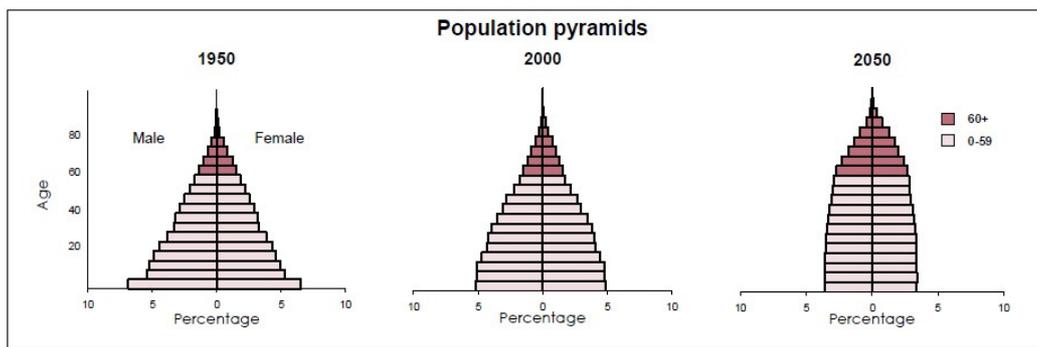
---

## 2 LITERATURE REVIEW

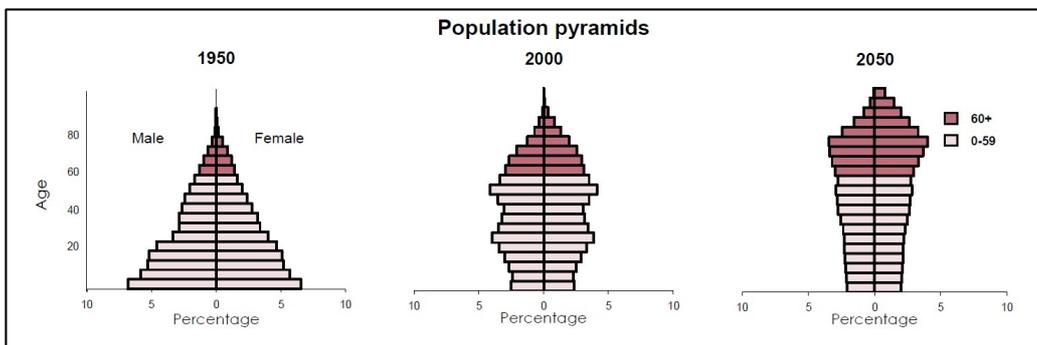
In the twenty-first century, the percentage of ageing population is increasing not only in the developed country and apparently in the developing country as well. Nearly in the future, all countries will be facing similar situation although the level of intensity and time frames are varies between each countries. This scenario will lead economic, political and social problem such as extension of retirement and pension, rising medical cost and increasing demand in health care service (Kose 1997). Fig. 1-1(a), Fig. 1-1(b) and Fig. 1-1(c) show the comparison of the ageing population of the World, Japan and Malaysia, respectively.

Statistics show that in less than 40 years, the world will have an increasing number of elderly people especially in developed country such as Japan and South Korea, but in developing country such as Malaysia the problem is less critical. Fig. 1-2 shows that, Japan has the highest number of ageing population in the world. In 1990, Japan population was approximately 124 million and 12% was over 65 years and older. In 1986 the percentage is increased to 23.6% and it is estimated to be 30% in 2020. It can be read in the newspapers and reports that the number of ageing people in Japan will further increase and it is going to be 40% in 2050. This problem is getting worse with the decreasing number of birth rate and it is expected to be more than 20% reduction by 2050. This scenario will increase the needs of assisting and taking care of the elderly every year.

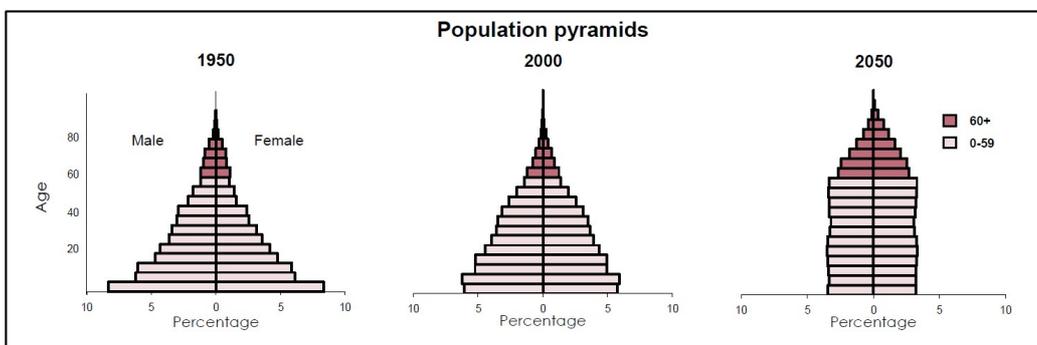
To address this problem, a variety of assistive robots have been proposed to assist the elderly in their everyday life such as eating, drinking, cleaning and for emergencies. The main problem with these service robots is the cost of owning them. Today, the need of having a service robot in each of the elderly house is a must. Researches in this area need to be continued in order to make the service robot more affordable and flexible.



(a)



(b)



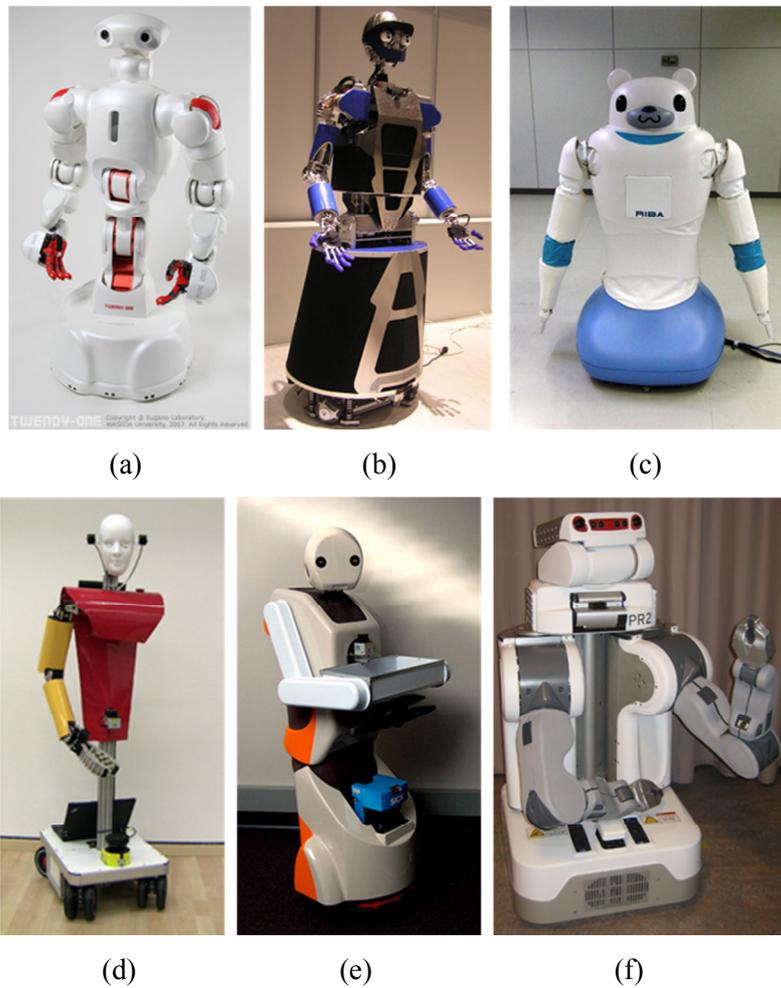
(c)

**Figure 1-1** Changes of population pyramid (a) World (b) Japan (c) Malaysia.

Country	Age structure population by country (%)					
	2010			2050 (projection)		
	0-14 years	15-64	65 and over	0-14 years	15-64	65 and over
Japan .....	* 13.2	* 63.7	* 23.1	8.6	51.8	39.6
Korea, Rep. of .....	16.4	72.4	11.1	13.2	54.0	32.8
Italy .....	14.1	65.6	20.4	14.3	53.0	32.7
Germany .....	13.5	66.1	20.4	14.5	54.6	30.9
China .....	19.5	72.4	8.2	13.5	61.0	25.6
France .....	18.4	64.8	16.8	17.6	57.5	24.9
Canada .....	16.4	69.5	14.1	16.2	58.9	24.9
Sweden .....	16.5	65.2	18.2	17.3	58.1	24.6
U.K. ....	17.4	66.0	16.6	17.2	59.2	23.6
Russia .....	15.0	72.2	12.8	16.9	60.0	23.1
Brazil .....	25.5	67.5	7.0	14.7	62.8	22.5
U.S.A. ....	20.1	66.9	13.1	18.8	60.0	21.2
India .....	30.6	64.5	4.9	19.0	67.6	13.5

Source: Statistics Bureau, MIC; Ministry of Health, Labour and Welfare; United Nations.

**Figure 1-2** Age structure population by country.



**Figure 1-3** Assistive mobile humanoid robot (a) TWENDY-ONE (b) ARMAR III (c) RIBA (d) Dynamaid (e) Snackbot (f) PR2.

The research and development of mobile humanoid robots has started since the era of Leonardo da Vinci (Rosheim 2006). Since then, there are numbers of humanoid robots that had been developed by private companies and universities. Among the established mobile humanoid robots that were developed are Waseda University's TWENDY-ONE (Iwata & Sugano 2009), Karlsruhe Institute of Technology's ARMAR III (Albers et al. 2006), University of Bonn's Dynamaid (Stuckler, Schreiber, et al. 2009), Snackbot from Carnegie Mellon University (Lee et al. 2009), RIBA from RIKEN-TRI (Mukai et al. 2010), PR2 from Georgia Institute of Technology (Chen et al. 2013), Pearl robot, a collaboration research from three different universities, University of Michigan, University of Pittsburgh and Carnegie Mellon University (Pollack et al. 2002), Assistant Robot from University of Tokyo (Yamazaki et al. 2012) and Care-O-Bot from Fraunhofer Institute of Manufacturing and Automation (Schraft et al. 1998). Fig. 1-3 shows some of the developed mobile humanoid robots.

## **2.1 Assistive Humanoid Robots**

Mobile humanoid robots have been developed for helping elderly people in their daily life activities for example assisting them with the household chores, in the hospital or in a care centre. Iwata & Sugano (2009) have developed a sophisticated symbiotic robot, TWENDY-ONE which provides a physical support to the elderly. The robot is developed to attend the elderly in their home environment especially in the kitchen. The robot has five basic features in the design namely, safety, friendliness, dexterity, high power and mobility. The robot consists of a head, a pair of compact passive mechanism arms, a pair of anthropomorphic hands and omnidirectional wheel. Tactile force sensors are placed on the robot body for safety when the robot is in contact with humans. A similar assistant robot, ARMAR has been de-

veloped by Karlsruhe Institute of Technology has the capability of moving from one room to another and picking a specific object commanded by the user. The robot consist of twenty five mechanical degrees of freedom, 4 dofs body, a pair of anthropomorphic arms with 7 dofs each, two simple grippers and 3 dofs head (Asfour et al. 2000). An improved version, ARMAR II and ARMAR III has been developed with improve hands, vision system and design. The newly developed hands, increased the capability of the robot manipulating and performing more complicated task such as opening a door and manipulating different shape household object (Asfour et al. 2006).

A household helper name Dynamaid has been developed not only for the elderly people but it can be a personal helper to human or doing our daily chores such as sweeping and cleaning. This robot has one head, an anthropomorphic arm, a gripper and four individual steerable wheels (Stuckler, Schreiber, et al. 2009; Stuckler, Grave, et al. 2009). In recent years, a new version of Dynamaid has been developed with an improved gripper design and vision system. Cosero, the improved version of Dynamaid has the ability to perform more complicate task such as opening the refrigerator door and opening a bottle cap (Holz et al. 2013; Nieuwenhuisen et al. 2013).

In a different environment, a nursing robot system called RIBA is developed to assist medical staff and nurses in the hospital or in a care centre. RIBA is developed to perform heavy physical task that requires human contact such as transferring patient from the bed to the wheelchair and vice versa. It needs caregiver to monitor while the robot performs the task and to make sure the patient's safety (Mukai et al. 2010). A similar robot RI-MAN, has been developed by (Odashima et al. 2006) has a similar

function as RIBA, assisting nurses in the hospital for lifting and moving the patient. One advantage of RI-MAN is the robot utilised soft touch sensors and react to the amplitude and location of the external forces thus increasing its safety features while in contact with human. In a similar environment, another mobile humanoid robot, PR2 is developed to assist patients or visitors in a hospital to the desired location by holding its hand. The robot has the ability to recognized the command from the users and bring them to the desired room. PR2 not only can guide human in the hospital but also can perform a domestic task such as selecting and picking a household object and opening the drawers and doors (Chen & Kemp 2011; Chen et al. 2013; Chen et al. 2010). Dario et al. (1996) proposed three type of robot in assisting patients and humans in the hospitals and institutions (URMAD), assisting the elderly and disable people at home (MOVAID) and a wheelchair integrated with robot arm (IMMEDIATE).

Pearl is another elderly assisting robot has been developed by three different universities. This robot reminds people about routine activities such as drinking, eating, taking medicine and also guiding them through the environment. Preliminary testing has been carried out at the Longwood Retirement Community Centre (Pollack et al. 2002). In a different environment, a semi-autonomous snack serving robot has been developed in Carnegie Mellon University to serve healthy snack to humans in a university building. The robot able to move around the university building and taking orders and send it ordered snack to the users. The robot has the ability to move around the university, communicate with users, understand the command and deliver the product (Lee et al. 2009; Lee & Forlizzi 2009).

Arnold is another mobile humanoid robot developed by (Bergener et al. 1999) that have one anthropomorphic arm and capable of interacting with human. The robot has the ability to move in dynamics environment, entering a room, manipulating object and avoiding obstacle. The robot has two cameras with small field of view and a wide angle view to mimic human eye. These cameras are for the robot navigation and object manipulation. A similar type of robot with one anthropomorphic arm has been developed by (Natale et al. 2007). The upper torso humanoid robot is designed to have precise 3D reaching for object manipulation, where the reaching capability of the robot did not rely on the arm and the head kinematics.

Another type of robot, Magilla with different capabilities of having a human sensory system has been developed by (Coelho et al. 2000). In their work the haptic and visual sensorimotor are cognitively integrated to have a human like sensory system specifically for grasping behaviour in human infant. Saika robot developed by JSK Laboratory, University of Tokyo is another robot developed for assisting human in daily activities (Konno et al. 1997). The robot has modularized features to reduce the developing cost and light weight with only eight kilogram. All the motors are kept inside the body arm and torso for better appearance. Lopes et al. (2004) has proposed a human imitating upper torso robot that can recognize and follow human motion via its vision system. The robot consists of one anthropomorphic arm that has the same degrees of freedom as a human. The robot is developed for rehabilitation purpose.

### **2.1.1 System Integration**

In order to perform a domestic task such as getting a bottle of juice from the kitchen or guiding humans to the desired location, numbers of considerations need to be made such as understanding the command, determining the environment, obstacles

avoidance, object recognition and manipulation, safety features and many others. Chen & Kemp (2011; 2010) has proposed a direct contact between human and robot to guide the human from one location to another, for example leading a child by the hand or assisting nurses a hospital. A direct physical interface is introduced to enables a user to influence the robot behaviour by making contact to the robot body. Testing has been done in a real hospital with volunteered nurses to test the performance of the robot. In their work, a combination of MEKA Robotic arms, an Omni directional Segway and a linear actuator by Festo are used as complete robotic system. The robot wrist is equipped with six axis force sensor to sense the input from human or while holding their hand. A nurse robot, PEARL has been developed by (Pollack et al. 2002) for assisting the elderly. Two main purpose of PEARL is to remind people about routine activities such eating, drinking, bathing and taking medicine and the other function is to guide them through their environment. A high level control architecture, partially observable Markov decision process (POMDP) has been adapted in this robot.

In other work by (Stuckler, Grave, et al. 2009), a personal robot for helping household chores has been developed. The robot Dynamaid adapted four schemes for navigation namely Fast Simultaneously Localization and Mapping (FastSLAM), localization, path planning and safe local navigation. Holz et al. (Holz et al. 2013), has proposed global-to-local control strategy to navigate the developed robot, Cosero from the transport box to the processing place for bin picking. A rough estimation of the exact environment in the form of 2D is employed. The approximate location is navigated globally and the robot is locally aligned with the transport box and the processing place. Adaptive Monte Carlo Localization is used to estimate the robot's pose in a given grid map using a laser range finder. A\* search is applied to find the shortest

obstacles free path from the estimate position to the target location. For the arm manipulation, LBKPIECE is utilised in order to maximised the performance and reduce the execution time.

## **2.2 Arm Motion Generation**

In order to perform daily life activities, a mobile humanoid robot is required to have the ability to locate and manipulate object. Therefore, they have to perform a wide range of tasks, such as picking an object and giving it to the human, removing an unnecessary object, opening and closing the door. The wide range of robot task requires different robot motion strategies. In addition, because there are an infinite number of trajectories connecting the robot hand position with the goal location, the robot has to select the best trajectory and speed in order to complete the task successfully.

Many different methods and approaches have been proposed in the past decades on humanoid robot arm motion generation. A minimum time trajectories robot arm motion has been proposed by (Sahar & Hollerbach 1986). In their work, a general solution is proposed to solve minimum time trajectory path which involves joint space tessellation, a dynamic time scaling and a graph search. The full dynamics of the arm movement and actuator constraints are incorporated. With these features, the arm motion while avoiding obstacle can be easily generated. Flash & Hogans (1985) had proposed minimum hand jerk criteria where the position vector of the hand is defined with respect to the Cartesian coordinate system. Differentiating the position three times will define the jerk of the hand. The arm motion from its initial to the goal position is generated by minimizing the time integral of the square magnitude of jerk. Rosenbaum et al. (1995) has proposed a similar approach of generating arm motion by

minimum angle jerk. The coordinate movement of the arm and trunk using optimization criteria defined in the joint space.

Minimum torque change criterion has been introduced by Uno et al. (1989), where control objects are the joint links plan in an intrinsic dynamic- mechanical space. The hand trajectory properties are reproduced based on the arm dynamics, posture, external forces and motion duration. An improve version of minimum torque change has been proposed by (Uno, Y. Kawato, M. Suzuki 1989). In minimum commanded torque change, the incorrect values of the inertia and viscosity as in minimum torque change have been improved. The same approach has been proposed by (Nakano et al. 1999) using representation of motor commands controlling the muscles. Kawato et al. (1990) utilized minimum torque change to produce a multi joint arm motion while avoiding obstacles and passing through points. Wada et al. (2001) has compared the performance of all four optimal theories namely minimum hand jerk, minimum angle jerk, minimum torque change and minimum commanded torque change. In his study, the minimum commanded torque change show the optimum results and has the closest trajectories to human.

In recent years, there are numbers of arm motion generation technique, Vahrenkamp et al. (2008) suggested Rapid-Exploring Random Trees (RRTs) which can adapt the number of active degree of freedom used in robot motion thus improving the performance and quality of the trajectories. The numbers of degree of freedom used are optimized using RRT. In an eight degree of freedom robot, RRT determines the optimum number of joints needed to complete the task. An improved version of RRT, Rapidly Exploring Dense Tree (RDT) has been proposed by (Vahrenkamp et al. 2011), with addition of automatic adjustment collision detection system.

Ang et al.(2009) proposed a minimum time motion planning of robot arm using Pareto based multi-objective Bees Algorithm for a SCARA robot. Four different operators are used to optimize the cubic splines trajectories thus minimizing the travelling time of the robot which are discrete recombination, intermediate recombination, line recombination and path redistribution and relaxation.

### **2.2.1 Arm Motion as an Optimization Problem**

During everyday task performance, humans move their arms in different ways satisfying different constraints. For example to move a cup of coffee, human arm need to move such as to minimize the acceleration in order not to spill the coffee. For more complicated tasks such as drawing a straight line or pushing a non-rigid object, the kinematics constraints such as velocity and acceleration are required to be decreased resulting in a longer execution time. Such scenarios inspired researchers to adapt the similar approach in generating robot hand motion for specific tasks.

For a robot hand to have a human like motion, similar characteristics and approaches of motion generation need to be properly considered. To address this problem, a single objective optimization Genetic Algorithm (GA) for robot trajectory planning and collision avoidance has been proposed (Rana & Zalzala 1996; Wang & Zalzala 1996; Pires & Machado 2000). Rana & Zalzala (1996) proposed an open loop minimum time planning of a two link planar robot manipulator. The optimization is done via evolutionary algorithm to have a minimum time motion and simultaneously avoiding obstacles. A similar approach of utilising genetic algorithm in searching the optimal robot manipulator path has been proposed by Wang & Zalzala (1996). In their study, the motion of six dof robotic arm is optimized in terms of minimum time. In other work by Pires & Machado (2000), GA is chosen to minimize the distance trav-

elled by a three link planar robot manipulator based on its kinematics and dynamics. The motion generations of these works are based on direct kinematics of the robot manipulator, which is proven better than motion generated using inverse kinematics by (Chen & Zalzal 1997). Based on direct kinematics of a two link planar robot manipulator, Kubota et al. (1997), generates robot manipulator collision free motion using virus evolutionary genetic algorithm optimizing the distance from its initial to goal position.

In recent decade, multi-objective evolution of robot arm motion generation has been proposed. Pires et al. (2007), proposed a multi objective motion generation for two and three dof planar robot manipulator optimizing two and five objective criteria. The five chosen objectives are; minimum joint traveling distance, minimum joint velocity, minimum Cartesian distance, minimum Cartesian velocity and minimum energy. The robot manipulator trajectory is minimized via GA adopting direct kinematics.

In other works by Ramabalan et al. (2008), two multi-objective evolutionary algorithms (MOEA); elitist non-dominated sorting genetic algorithm (NSGA-II) and multi-objective differential evolution (MODE) are proposed to generate the motion of two robots, Cartesian and PUMA 560. Both robot end effectors are required to do pick and place operation in the workspace avoiding three obstacles. The two objective functions selected to be optimized are the travelling time and consumed energy. In order to select the Pareto optimal front, normalized weighting objective function and fuzzy membership function are used. Liu et al. (2011) proposed an improved version non-dominated differential evolution (NSDE) to generate the two and three degree of freedom (dof) planar redundant manipulator motion. Multiple objectives namely

singularity avoidance, obstacles avoidance and joint limit avoidance are chosen to be optimized. Rehman et al. (2010) have proposed MOGA for generating parallel kinematics machine motion while considering three objectives optimization. The optimal path of the three dof parallel kinematics machine is generated optimizing the minimum electric energy used by the actuators, maximum torque and minimizing the shaking force.

In the next chapter, our developed mobile humanoid robot is discussed in terms of its kinematic analysis, mechatronics design, software, arm motion generation and the system integration.

---

# CHAPTER 3

---

## 3 MOBILE HUMANOID ROBOT PLATFORM

Most of the previously developed mobile humanoid robot are well established, having good design, robust and has the ability to perform domestics and complicated task. In this study, we developed a new upper torso mobile humanoid robot for assisting the elderly and disable people. It has the basic of human motion and sufficient enough to do simple task. The upper part is attached to a mobile platform for more flexibility and mobility. Mobile platform is chosen based upon the stability and its simplicity over a pair of leg (Mohamed & Capi 2012). The upper body has ten degrees of freedom hands, a pair of simple grippers and two degrees of freedom head as in Fig. 3-1.

In this chapter, the detail descriptions of the developed mobile humanoid robot are discussed in terms of its kinematics analysis, mechatronics design, software configuration and kinesiology of the robot movements. This robot is originally designed to assist elderly or handicap people in everyday life chores. The design of the mobile humanoid robot are considering the safety features, ability to navigate throughout the environment, obstacles avoidance, ability to perform simple manipulation, position determination and object recognition (Mohamed & Capi 2012). Later in this thesis, the proposed neural controllers will be implemented on this mobile humanoid robot to investigate its performance.

### 3.1 Kinematics Analysis

The robot hand positioning is very important in order to perform a task with high accuracy and stability. Proper kinematics analysis is required to determine the current and goal position of the robot hand. The direct and inverse kinematics analysis of the robot arm is discussed in this section and in Appendix A.

#### 3.1.1 Denavit-Hartenberg (DH) Parameters

There are many methods to determine the direct kinematics of a robot arm and one of the most established methods is Denavit-Hartenberg (DH) analysis. This is the reason DH analysis is utilized in this thesis to determine the direct kinematics of the robot hand. In DH analysis, the homogeneous transformation matrix can be determined, which specifies the position and orientation of the robot hand with respect to the base as in Fig. 3-1 (Sciavicco & Siciliano 2001; Jazar 2007; Asada & Slotline 1986; Spong et al. 1993).

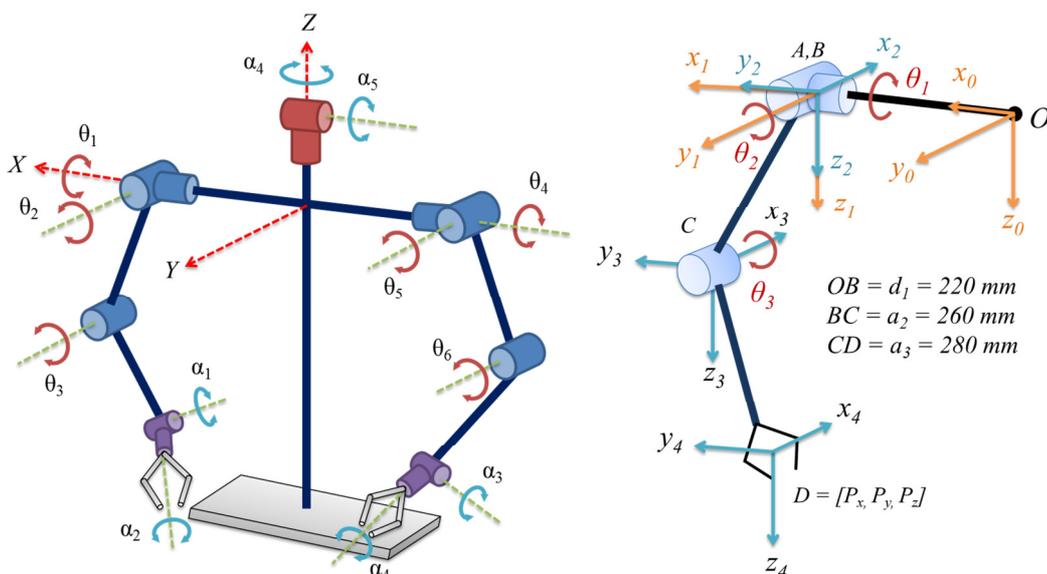


Figure 3-1 Coordinate frame of the robot upper body.

Implementing DH convention and following the procedures, the D-H parameters for the robot hand assigned frames are defined in Table 3-1. The transformation equation of the DH analysis is as follows;

$$T_i^{i-1} = A_i^{i-1} A_i' = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

**Table 3-1** D-H Parameters.

Joint, $i$	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
OA	0	$90^0$	$d_1$	$\theta_1$
B	$a_1$	0	0	$\theta_2$
C	$a_2$	0	0	$\theta_3$
D	0	0	0	Gripper

By substituting these parameters into equation (1), the transformation matrices  $T_1$  to  $T_4$  can be shown as follows:

$$T_1^0 = A_1 = \begin{bmatrix} \cos \theta_1 & 0 & \sin \theta_1 & 0 \\ \sin \theta_1 & 0 & -\cos \theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = A_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & a_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & a_2 \sin \theta_2 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^2 = A_3 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & a_3 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & a_3 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^3 = A_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The robot hand position can be determined based on the forward kinematics equation obtained from the transformation matrix as follows;

$$T_4^0 = A_1 A_2 A_3 A_4 = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & C_1 (a_3 C_{23} + a_2 C_2) \\ S_1 C_{23} & -S_1 S_{23} & -C_1 & S_1 (a_3 C_{23} + a_2 C_2) \\ S_{23} & C_{23} & 0 & a_3 S_{23} + a_2 S_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where  $C_1$  is  $\cos \theta_1$ ,  $C_2$  is  $\cos \theta_2$ ,  $S_1$  is  $\sin \theta_1$ ,  $S_2$  is  $\sin \theta_2$ ,  $C_{23}$  is  $\cos(\theta_2 + \theta_3)$  and  $S_{23}$  is  $\sin(\theta_2 + \theta_3)$ . The first three columns represent the orientation of the end effector whereas the last column represents the position of the robot hand (Fig. 3-1) as in equation (3) below.

$$\begin{aligned} P_x &= d_1 + a_2 \sin \theta_2 + a_3 \sin(\theta_2 + \theta_3) \\ P_y &= \sin \theta_1 [a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3)] \\ P_z &= \cos \theta_1 [a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3)] \end{aligned} \quad (3)$$

### 3.1.2 Inverse Kinematics

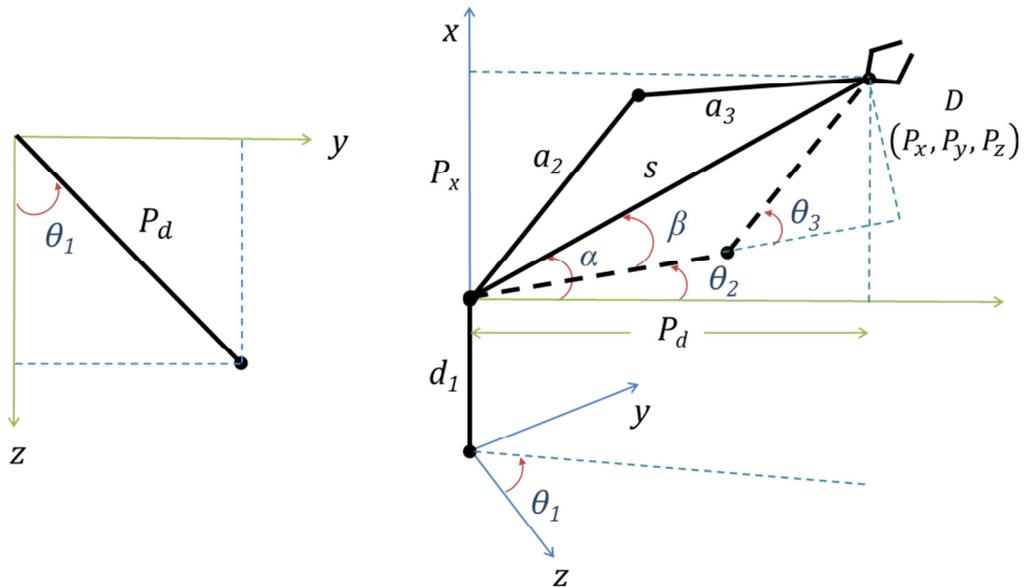
The direct kinematics analysis in the previous section established a functional relationship between the joint angle and the robot hand position and orientation. In in-

verse kinematics analysis, the joint angles are determined based on the given position of the robot arm. Inverse kinematics is very important for determining the current robot hand position and goal position to perform a specific task. In this research, a geometric approach is utilised to determine the inverse kinematics of the robot arms. Referring Fig. 3-2, the three joint angles of the robot arm are determined using equation (4), equation (5) and equation (6) for shoulder, upper arm and lower arm respectively. The detail analysis of the robot kinematics is shown in Appendix A.

$$\theta_1 = \tan^{-1}\left(\frac{P_y}{P_z}\right) \quad (4)$$

$$\theta_3 = \cos^{-1}\left(\frac{(P_d^2 + (P_x - d_1)^2 - (a_2^2 + a_3^2))}{2a_2a_3}\right) \quad (5)$$

$$\theta_2 = \tan^{-1}\left(\frac{(P_x - d_1)}{P_d}\right) \pm \cos^{-1}\left(\frac{a_2 + a_3 \cos \theta_3}{s}\right) \quad (6)$$



**Figure 3- 2** Inverse kinematics analysis of the robot hand.

These equations are utilised in the MATLAB program for the robot arm simulations and on the real robot. The same approach is done for the left hand with a minimum modification of the coordinate system. The detail explanation and derivation of the robot hand's inverse kinematics are discussed in (Mohamed & Capi 2012).

## **3.2 Mechatronics Design & Hardware Architecture**

The development process of the mobile humanoid robot is discussed in this section, in terms of the mechanical design, mechatronics systems and the software architecture of the robot system. The robot system is divided into two main parts, the upper body and the moving platform. The upper body designed is mainly for object manipulation and recognition, while the mobile platform is utilised for the robot navigation and obstacles avoidance.

### **3.2.1 Upper Body**

The upper body of the mobile humanoid robot is consisting of a head, a pair of arms and space for all the electronics parts and components. The initial design of the upper part is done using Solidworks to make sure the kinematics motion of the hands is according to the developed mathematical model and the position of all the electronic components are well place in the system. Each component is drawn separately as components and assembled as a part. Most of the design, fabrication and assembly processes are done in the lab and only the shoulder base part is done in the workshop.

All the DC motors utilized to actuate the joints are aligned in series and kept inside the arm and main body for higher stability while in motion and better appearance. A pair of harmonic drives is attached to the shoulders for smooth motion without gear backlashes. Two grippers are attached to both arms for simple manipulation and

they are actuated by servo motors. Aluminium plates and rods are chosen for most of the robot parts due to its light weight and high strength properties. Initially, plastic material is chosen for the robot arms cover, but in the final design, combinations of two L-shape plate are chosen. These covers not only secure the electronics components inside the arm structures but it is also preventing the arm from twisting while in motion or holding an object. Fig. 3-3 and Fig. 3-4 show the arm design assembly and the real robot arm for right and left hand, respectively.



**Figure 3-3** Right hand design of the mobile humanoid robot.

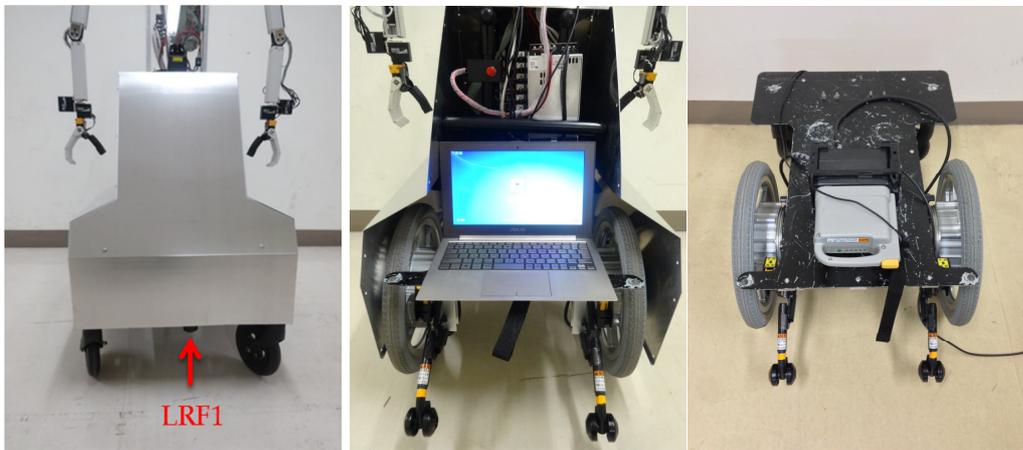


**Figure 3-4** Left hand design of the mobile humanoid robot.

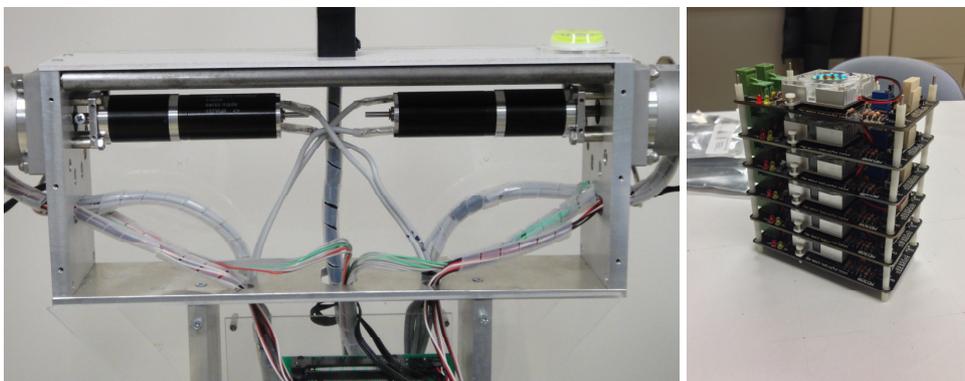
Another important aspect in robot design is the safety features especially when interacting and performing task in human environment. It is a must for a robot to have high safety features in order to co-exist with humans. Currently, an emergency stop button is placed at the back of the upper body for emergency cases and stopping the robot instantly.

### 3.2.2 Mobile Platform

The mobile platform is powered by two AC motors with 24V battery, a controller and base for the upper body. The battery, PC, AC motors, laser range finder (LRF) and the upper body is placed on this platform as in Fig. 3-5. The mobile platform has a maximum speed of  $1 \text{ ms}^{-1}$  and it can be easily controlled using MATLAB (PC). LRF1 is placed at the front lower part of the robot for object detection and obstacles avoidance. The upper body is placed in the middle of the platform to increase its stability.



**Figure 3-5** Mobile platform.



**Figure 3-6** DC Motors and motor drivers.

### 3.2.3 Electronic Components

#### 3.2.3.1 DC Motors

Six Maxon brushless DC motors and six MD30B DC motor drivers are used in this system. These DC motors moves the robot shoulders, upper arms and lower arms as in Fig. 3-6. The motor drivers are very important in order to control the speed of the DC motor. Ability to control the motors at low speed is very important criteria in order to have a stable and smooth motion hand trajectory. Different gear ratios of DC motor are selected based on the requirement of each joints. These motors can be powered by a 9V power supply or a battery.

#### 3.2.3.2 AC Motors

For smooth robot navigation, two Yamaha AC motors are chosen and they are powered by 24V battery as in Fig. 3-7. Smooth motion is very important for the stability of the robot, thus the robot will have low vibration while navigating.



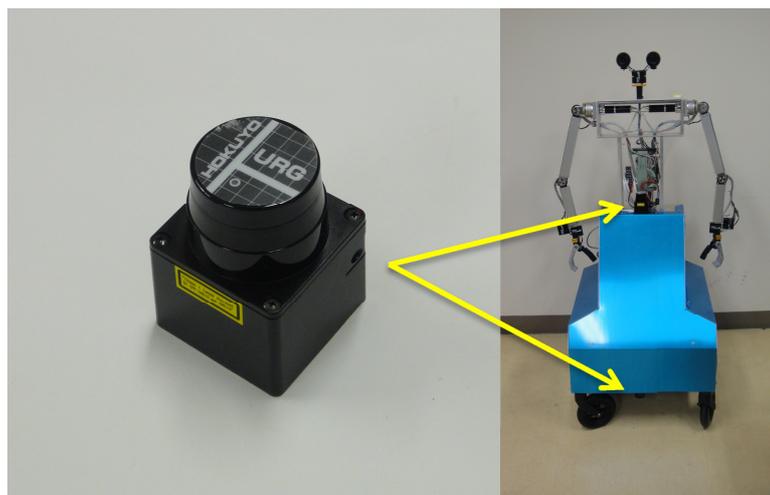
**Figure 3-7** AC Motors.

### 3.2.3.3 Servo Motors

The robot hand motion and the head movement utilised eight servo motors as in Fig. 3-8. Two servos are for simple gripping, four servos for roll and pitch motion of the robot hands and two servos for the robot head pan and tilt. Servos are selected for its simplicity, light weight and easy to control. The grippers are developed in the lab and have the ability to do a simple grasping.



**Figure 3-8** Servo motors.



**Figure 3-9** Laser range finder (LRF).

### 3.2.3.4 Laser Range Finder (LRF)

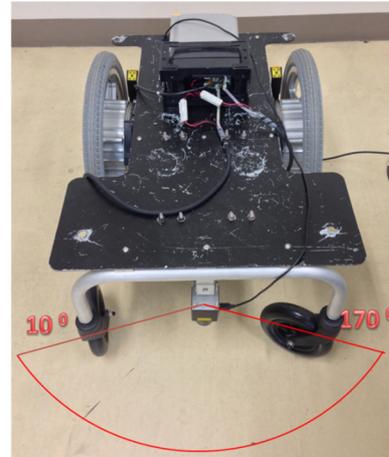
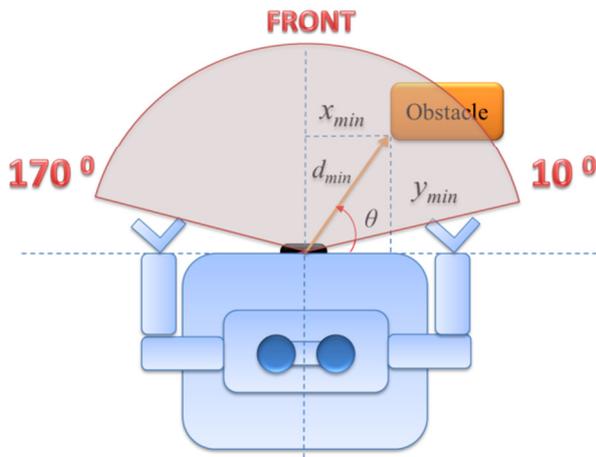
Robot navigation is almost impossible without the laser range finder (LRF). Two set of Hakuyo LRF (URG-04LX-UG01) are used, one for robot navigation around the environment and the other is for object position determination on the standard table height. The LRF can scan from 2 cm to 560 cm in distance with  $240^{\circ}$  range. The accuracy of the measured distance is within 3% at 100 ms/scan scanning time and the scanning resolution is  $0.36^{\circ}$ . The LRF has light weight features weighing only 160 g and powered by 5V voltage supply. The position of the LRFs is shown in Fig. 3-9.

Fig. 3-10(a) shows obstacles detection for mobile platform's LRF. The scanning angle is set to be from  $10^{\circ}$  to  $170^{\circ}$  with obstacle detection within 50cm to 100 cm. If the obstacles are detected on the right area the mobile platform will steer to the left and vice versa. The obstacles detection via LRF can be determined using simple equations as follows:

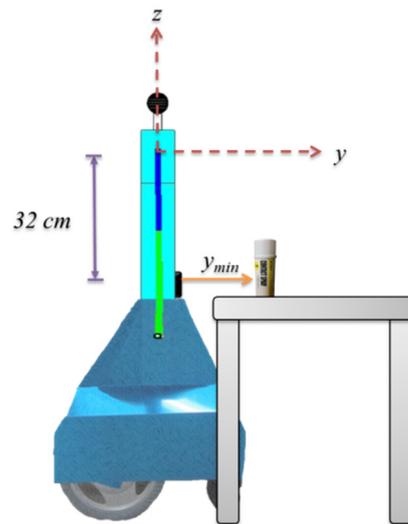
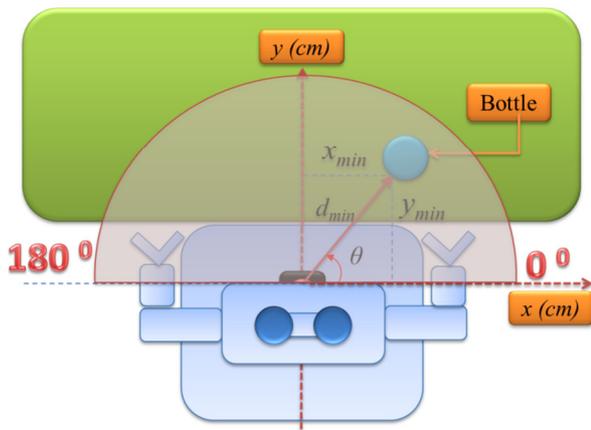
$$\begin{aligned} y_{\min} &= d_{\min} \sin \theta \\ x_{\min} &= d_{\min} \cos \theta \end{aligned} \quad (7)$$

where  $d_{\min}$  is the shortest distance determined using LRF.

Fig. 3-10(b) shows the object position determination on a standard table height by the robot. Utilizing the second LRF, the minimum distance of the object on the table can be detected and by applying equation (7), the position of the object on the table can be determined. The function of this LRF is not only for determining the object to be manipulated on the table but also for detecting the obstacle in the workspace area with the assistance of the camera.



(a)

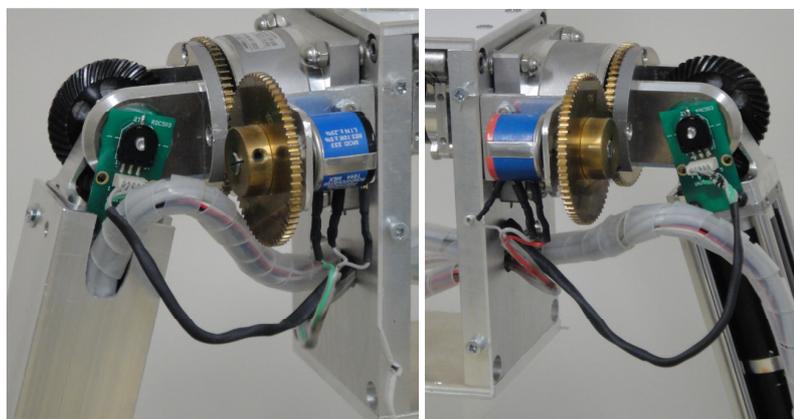


(b)

**Figure 3-10** (a) Obstacle detection via LRF1 (mobile platform) (b) Spray can detection via LRF2 at table height (upper body).

### 3.2.3.5 Potentiometers

In order to determine the robot arm joint angles, six potentiometers are used. These angles are utilized to determine the robot hand positions in Cartesian space. Two type of potentiometers used are shown in Fig. 3-11. The actual joint angles are acquired directly from these potentiometers thus the minimum and maximum angular speed of each joint can be determined. The joint angular speeds are used in the simulations and experiments to generate the kinematics properties of the robot hand.



**Figure 3-11** Potentiometers.

## 3.2.4 Software Architecture

### 3.2.4.1 MATLAB Simulation

Before the optimal neural controllers can be adapted on the simulator and on the real robot, the weight functions of both SOGA and MOGA are generated in MATLAB environment using the developed toolbox. The simulator of the mobile humanoid robot is also developed using MATLAB as in Fig. 3-12(a). The generated neural controller using SOGA and MOGA will be implemented in this environment to

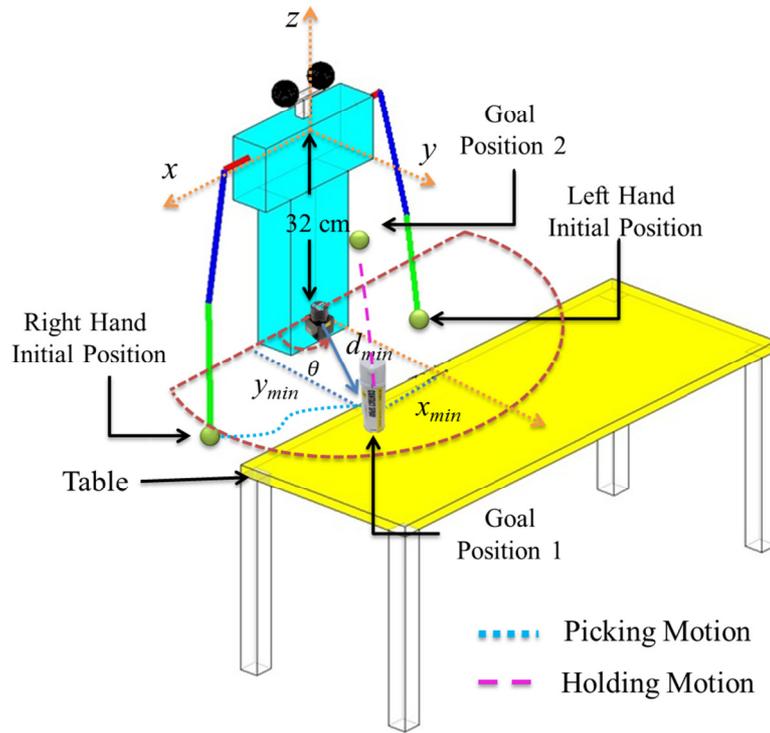
verify the performance before it can be implemented on the real robot. MATLAB is chosen based on its robustness and capability to solve complicated calculation, data acquisition and manipulation. Other advantage of MATLAB is the ability to interact with the cameras, robot controller and mobile platform in real time. A special graphical user interface (GUI) has been designed that allow user to control the robot manually as in Fig. 3-12(b). Each joint can be moved separately and simultaneously via this GUI.

The robot maximum and minimum motion for each joint has been included in the main program to have additional safety features while the robot is in motion. These features are very important in order to make sure that all the mechanical joints and electronics components are not easily damage.

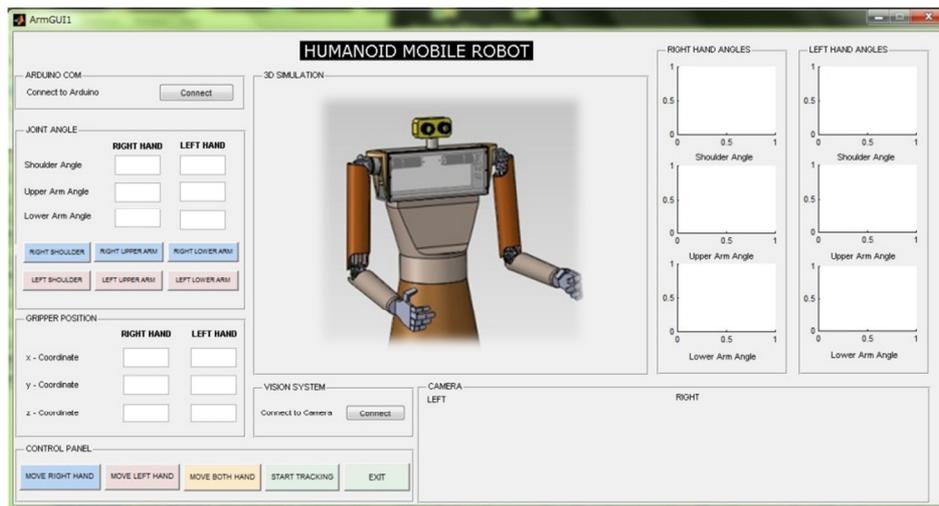
### **3.2.5 System Integration**

Fig. 3-13 shows the complete system and the developed mobile humanoid robot. The key specifications the mobile humanoid robot is shown below:

- Arm length – 54 cm
- Total height – 134 cm
- Total length – 67 cm
- Robot platform width – 52 cm
- Upper body weight – 14 kg
- Mobile platform weight – 16 kg
- Maximum moving speed 1 m/s



(a)



(b)

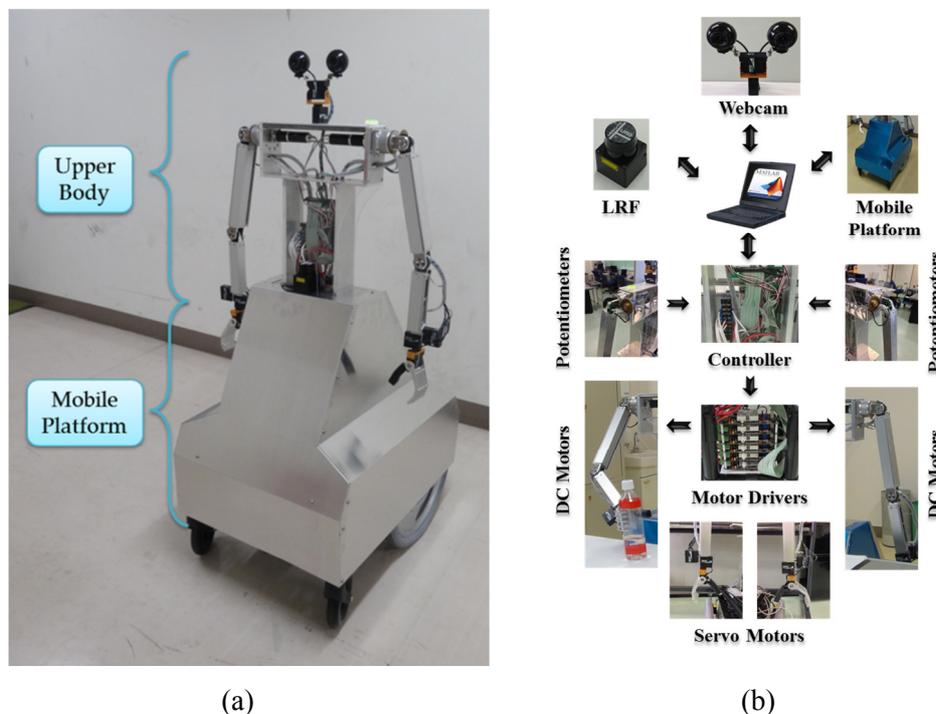
Figure 3-12 (a) Robot arm simulator (b) GUI.



**Figure 3-13** Developed mobile humanoid robot.

The integrated mobile humanoid robot system and the position of each electronics components and mechanical parts are shown in Fig. 3-14. Each component and their functions that have been discussed in the previous section are assembled to have a complete system. The mobile platform, the upper, both cameras and both laser range finders are control directly from the PC via USB connector. This complete system will be utilised to compare the performance of the generated neural controller using single

objective and multi-objective genetic algorithm. Eventhough the developed system is incomparable with the established robot platform; our robot has the ability to adapt the proposed neural controllers and show good performance.



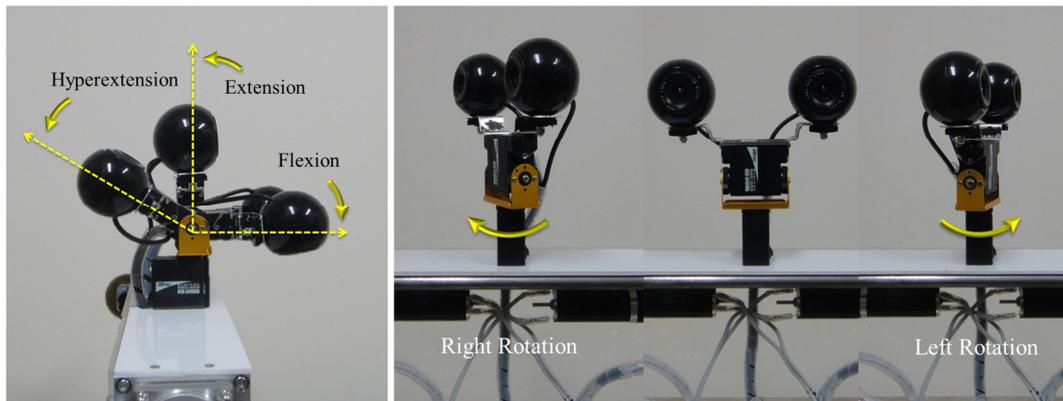
**Figure 3- 14** Integrated system.

### 3.3 Kinesiology of the Robot Motion

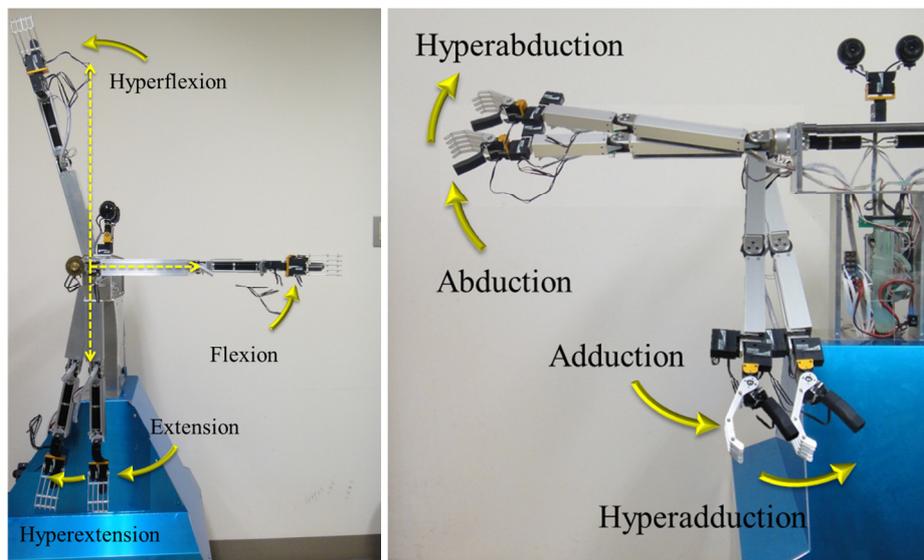
Human motions consist of six basic movements namely, flexion, extension, abduction, adduction, supination and pronation. Flexion is a bending movement relative to reference position and extension is a straightening movement as the joint return to is reference position. Abduction is movement away from the body centreline and the return motion is called adduction. Supination is the movement of the forearm where the palm rotates forward and pronation is when it returns back to the reference position (Hamill & Knutzen 2003; Hamilton et al. 2008). We adapted these movements on the

developed robot with limited available joints. The movements are divided into three section; head, arm and hand.

The two degrees of freedom robot head not only capable of executing flexion and extension movement but also left and right rotations as in Fig. 3-15. These movements are very important for robot navigation and object recognition. Two servo motors are used to actuate both pan and tilting motion.



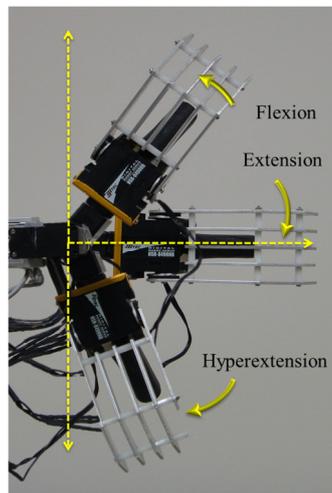
**Figure 3- 15** Robot head movements.



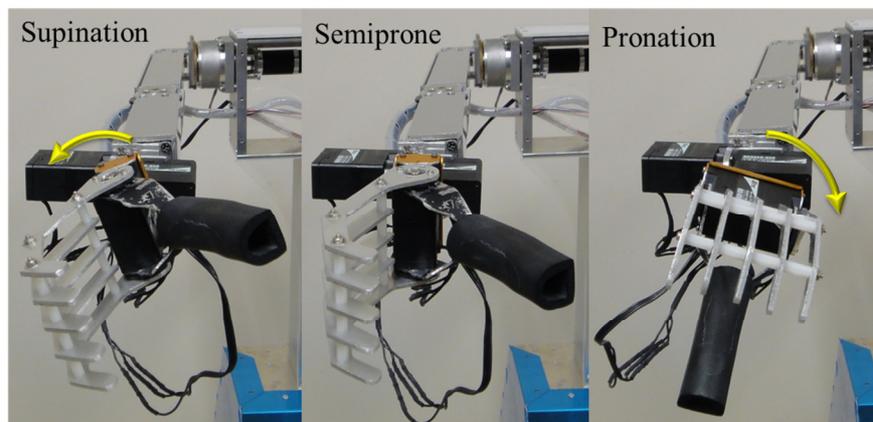
**Figure 3- 16** Robot arm movements.

Fig. 3-16 shows the robot arms movement of flexion, extension, abduction and adduction. This motion is important to determine the maximum reach of the robot hand in the simulated and working environment. For simple manipulation by the robot hand, flexion, extension, pronation and supination movements are important and it is shown in Fig. 3-17 and Fig. 3-18, respectively.

By knowing the kinesiology of the robot arm motions, the safety distance and the effective working area can be determined and implemented in the robot program. The upper and lower limit of all the joints can be set and it is shown in Appendix B.



**Figure 3-17** Robot hand flexion and extension movements.



**Figure 3-18** Robot hand supination, semiprone and pronation movements.

---

# CHAPTER 4

---

## 4 EVOLUTION OF NEURAL CONTROLLERS

In order to find one or more feasible solutions related to one or more objective, optimization is needed to make a system more effective and functional. Evolutionary algorithm utilised both neural network and genetic algorithm is chosen based on the performance and robustness of the technique. Two types of optimizations have been adapted in this thesis, single objective and multi objective optimization. In our implementation, both category of optimization are applied to the simulation and the mobile humanoid robot. Before the neural controller can be generated, the tasks for the robot arm have to perform need to be determine and it will be discussed in the next section.

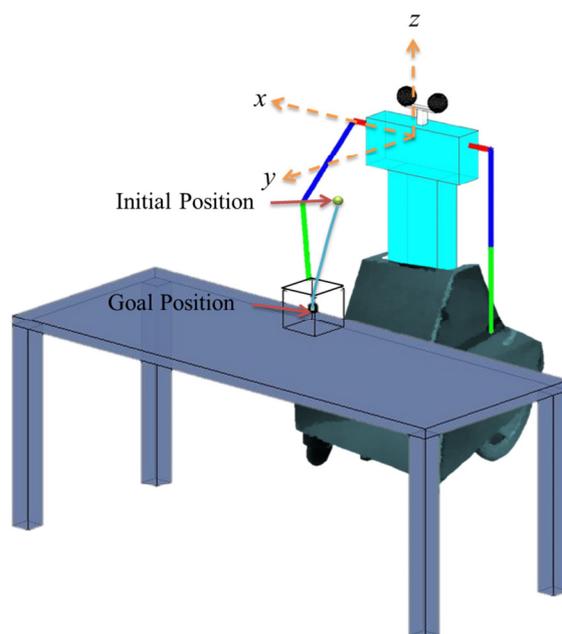
### 4.1 Problem Formulation

In performing a domestic task such as picking and placing a bottle, removing the thrash, pushing a box or avoiding obstacles, the humanoid robot arm trajectory and speed must be carefully selected in order to complete the task successfully. Therefore, in each stage of task performance, the main problem is what trajectory and how the moving speed must change connecting the robot hand and goal positions. The humanoid robot has to move the hand (object) from the initial to the goal position, which is connected with an infinite number of trajectories and motion velocities. The main problem is how to determine the joint trajectories in order to reach the goal position in minimum possible time or distance.

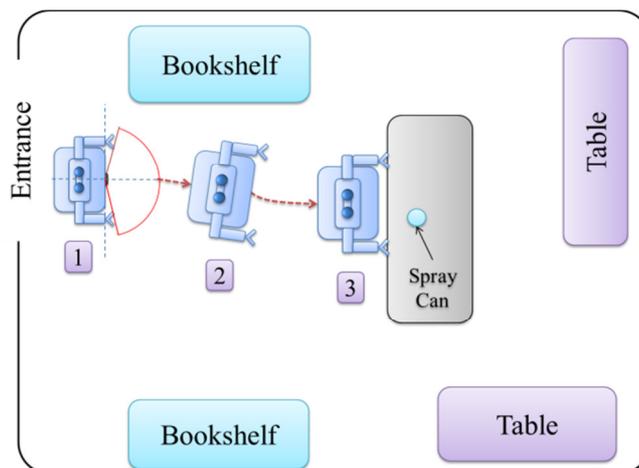
#### 4.1.1 Robot Arm Tasks Selection

In this thesis, the mobile robot humanoid robot is required to perform three different tasks. In the initial stage of this research, a task of placing a bottle on a table from a holding position is chosen as shown in Fig. 4-1. This motion is chosen to verify the performance of our mobile humanoid robot adapting the proposed optimal neural controllers. In this task, the performance of single and multi-objective arm motion generation are tested and compared.

In the next task, the mobile humanoid robot has to move from the lab entrance toward the table with the assistance of webcam and laser range finder for simple navigation and obstacle avoidance as in Fig. 4-2. The robot utilized the camera in order to get in the center of the table (from position 1 to 2). With the assistance of the LRF1 (Fig. 3-10(a)), the robot moves near to the table and stop at the desired distance. For this experiment, the table height is set to be 74 cm, which is the standard range height of a normal table.



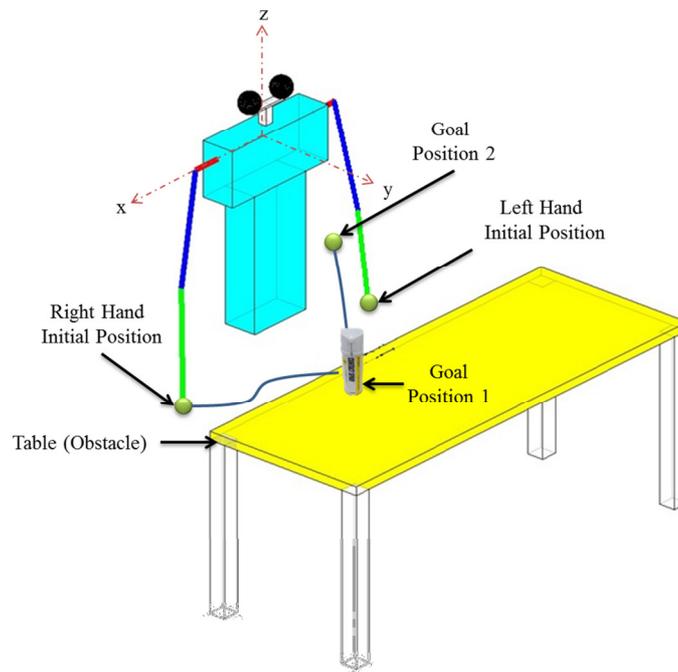
**Figure 4-1** Task 1: Placing a bottle on the table.



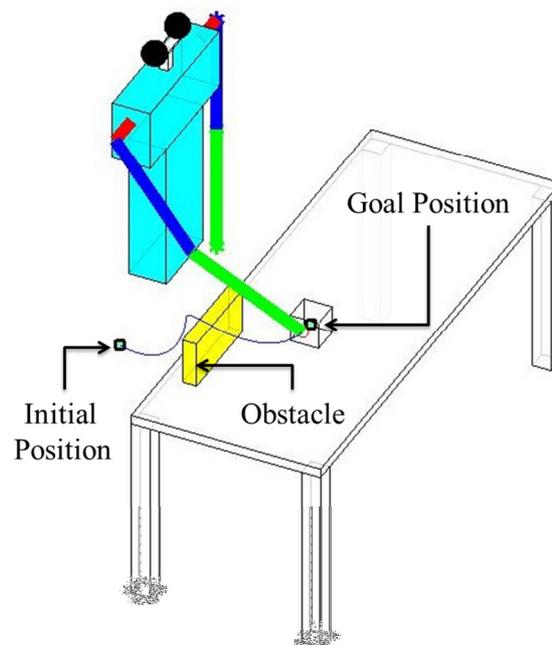
**Figure 4-2** Experimental setup for reaching the table

When the robot reaches the table (position 3) the object is recognized by the camera image and LRF2 (on the upper body) is utilized to determine the position of the spray can in  $x$  and  $y$  directions. The position of LRF2 is 32 cm from the reference position of the robot, as shown in Figure 3-10(b) ( $z$  direction). Once the position of the spray can is determined, the robot hand is required to move to the spray can position while avoiding the table (obstacle) and grasp the spray can (goal position 1), then the spray can is picked up to a holding position (goal position 2) as in Fig. 4-3. The position of the spray can on the table is randomly chosen in the simulation, but experimentally it is determined using LRF2 (Fig. 3-10(b)), as in equation (7). The robot will choose left or right hand to perform the task based on the position of the can on the table.

In the final task, an obstacle is introduced in the pathway of the robot hand with a similar motion of reaching the spray can as in Fig. 4-4. In simulation, the obstacle, initial and goal position are pre-determined but on the real robot all positions are determined via laser range finder.



**Figure 4-3** Task 2: Reaching a spray can on the table and picking motion to a holding position.



**Figure 4- 4** Task 3: Reaching a spray can on the table while avoiding obstacle

### 4.1.2 Arm Motion Generation

In our work, we proposed an optimal arm motion generation for the developed mobile humanoid robot arms via single and multi-objective genetic algorithm (SOGA and MOGA). Each arm will have a set of neural networks consisting of three inputs, six hidden layer and three outputs. The weights of the NNs are trained using GA (Mohamed, Kitani, et al. 2013c). The detail explanations on the evolution of neural controllers are discussed in Section 4.2 and Section 4.3.

Before the optimal neural controllers can be generated, one important aspect is determining the required objective functions. As we had discussed before, in performing a daily life activities, the trajectory and speed of the motion are the main factors need to be considered. In this case we had chosen four different objective functions to be considered throughout this study namely, minimum execution time (MT), minimum distance (MD), minimum acceleration (MA) and minimum angular acceleration (MAA). For each objective function one optimal neural controller is evolved utilising SOGA. In MOGA, Four different combinations of the criteria have been chosen namely, MT-MD, MT-MA, MD-MA and MT-MD-MA. The performance of the these optimal controllers are tested and compared with one extreme solution (SOGA) for performance comparison (Mohamed, Kitani, et al. 2013c; Mohamed, Mano, et al. 2013a).

In the second task, a similar approach is adapted to generate two set of neural controllers for picking and holding motions of left and right hand. The performance of these optimal solutions is tested by choosing three different position of the object on the table for the robot hand to reach. In simulation, the positions are randomly chosen

but in the experiment, the object is detected using laser range sensor (Mohamed, Kitani, et al. 2013a).

In a dynamic environment where an obstacle is placed in the pathway of the robot arm motion, a similar motion generation approach is utilized with additional penalty function introduced in the MOGA. Currently, a set of optimal neural controllers are generated for obstacles avoidance. Once the robot detected the obstacle using LRF, the robot will intelligently choose the best neural controllers for the arm to move to the goal position while avoiding the obstacle. The same neural controllers for each task are adapted on the real robot to compare the performances both in simulated environment and experimentally. The detail explanations on this part are explained in Section 4.4.

Additional task of robot navigation in our lab environment is performed to test the SOGA of the mobile platform. The robot has to navigate its way from the lift into the lab room avoiding any obstacles along the way. The lab environment is divided into three and for each environment one optimal neural controller is generated. Minimum distance between the starting and goal position is chosen as the objective function of the task.

### **4.1.3 Proposed Method**

The main advantage of our method is that we employ a single neural controller for each objective function or their combinations to generate the robot arm motion in a wide range of initial and goal location. These different criteria will make the robot more intelligent when choosing the best objective function for the given task. Another advantage of the proposed algorithm is the ability of MOEA to find multiple Pareto

optimal solution in a single run. In addition, some neural controllers optimize simultaneously multiple objectives.

The actual motor angles are acquired directly from the real robot and from these data; angular velocity, end effector velocity and acceleration are calculated. The joint angular velocities are used in the simulation and experiment to generate the kinematic properties of the robot hand.

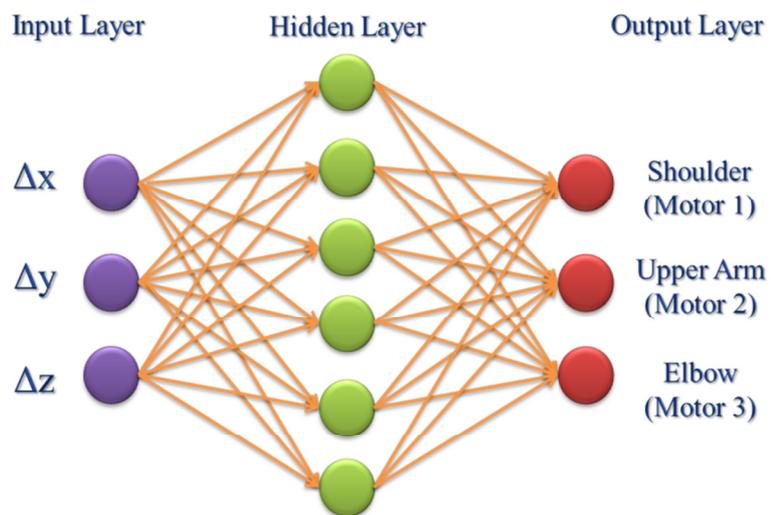
## 4.2 Neural Networks

Neural network (NN) is a mathematical model that is inspired by the biological neural networks. It is a set of algorithms for optimization and consists of links, interconnecting state variables, weight values associated with each links and a transfer function. Mainly there are two types of NN one is feedforward neural network (FFNN) and the other one is backpropagation neural network (BPNN). FFNN are the most widely used in many practical applications and it has been chosen for its simplicity and robustness compared to BPNN which has some drawback if the complexity of the data is increasing (Montana & Davis 1989).

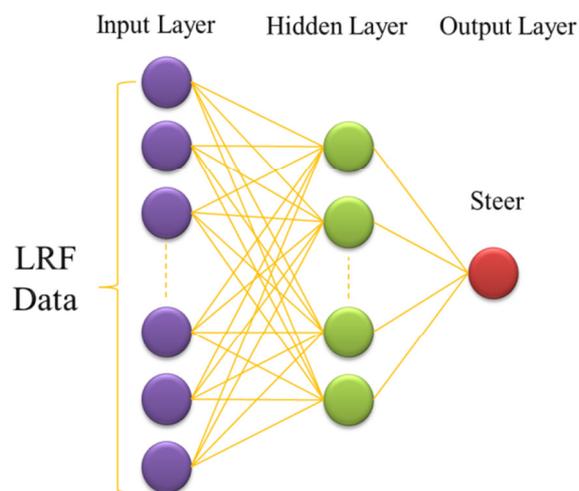
In our implementation FFNN is used both for the robot platform navigation and arm motion generation. For the robot arm motion the FFNN receives three inputs: the difference between the robot hand and goal positions coordinate in x, y and z axis as in Fig. 4-5. The inverse kinematics, based on potentiometer readings, is utilized to determine the current position of the robot hand. In simulated environment, the goal position is pre-determined while in real situations is generated based on the image processing. The output units directly control the three DC motors used to move the shoulder, upper arm and lower arm. The output units use a sigmoid activation function with the range from 0 to 1 where 0 to 0.5 is for one the motor moving direction and

0.5 to 1 for the opposite direction. The weight connections of the neural controller are optimized using genetic algorithm.

For robot navigation, the FFNN received 16 inputs from LR sensor, 10 hidden layers and a single output for steering. The input angles acquired from the laser range sensor are divided into six segments and utilised in the FFNN. The FFNN diagram for the mobile platform is shown in Fig. 4-6.



**Figure 4-5** FFNN for robot arm.



**Figure 4-6** FFNN for mobile platform.

The distance information from LR1 (inputs) is normalized to be the value from 0 to 1 using sigmoid activation function as follows:

$$y = \frac{1}{1 + e^{-x}} \quad (8)$$

### 4.3 Genetic Algorithms (GA)

Genetic algorithms are adaptive heuristics and global searching technique inspired by mimicking some of the process observed in natural evolution (Goldberg 1989). Holland (1975) has introduced a population based GA with crossover, inversion and mutation to solve optimization problems. His attempt to put computational evolution on a firm theoretical became the basis of almost all subsequent theoretical work on GA (Mitchell 1999). Fig. 4-7 shows the basic of GA process.

In our work, we used an extended multi-population genetic algorithm, where the subpopulations apply different evolutionary strategies (Capi & Doya 2005; Belding 1995; Cantu-paz 2000). In addition, the subpopulations compete and cooperate among each other. The summary of SOGA parameters is shown in Table 4-1.

**Table 4-1** Summary of genetic algorithm parameters.

Number of Subpopulations	3
Number of Individuals	450, 450, 300 (1200)
Maximum Generations	80

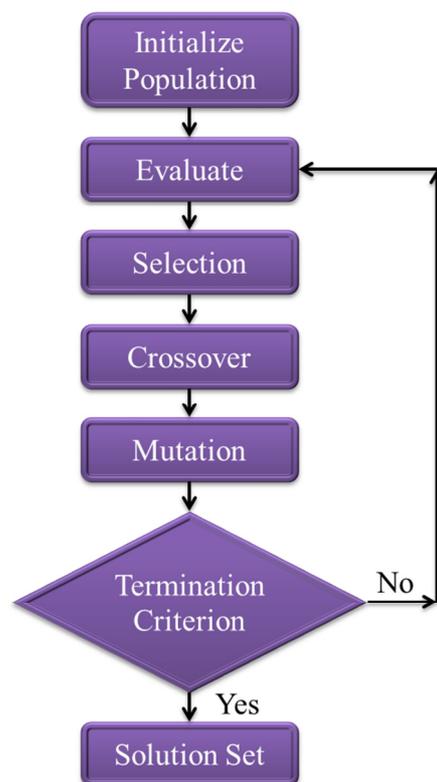
### 4.3.1 Multi-Objective GA

Evolutionary algorithms have proven to be well suited for optimization problem with multiple objectives. It becomes the method of choice for solving optimization problem that are too complex to solve using deterministic techniques such as Jacobian method. The main advantage is they are able to gain a number of solutions in a single run (Belding 1995; Cantu-paz 2000; Dias & de Vasconcelos 2002).

In our work a non-dominated sorting genetic algorithm (NSGA) was employed to evolve the neural controller. NSGA has a better performance than other MOEAs as shown in (Capi 2007) where multiple Pareto optimal solutions can be successfully determined. NSGA was employed to evolve the neural controller where the weight connections are encoded as real numbers. Dias & de Vasconcelos (2002), has compared the NSGA with four other MOEAs using two test problems. The comparison show that NSGA performed better than the others, where the Pareto optimal solutions can be successfully generated. Details explanations on NSGA are discussed in (Capi 2007).

### 4.3.2 Objective Functions

Before the robot arm motion and navigation can be generated, the objective functions need to be carefully selected and it is required to be as close as human motion. These objective functions will guide the GA in order to have the optimal solutions. The robot objective functions are divided into two sections, one for the robot arm motion, both for single and multi-objective optimization and the other one is for the robot navigation, which is considering a single objective function optimization only.



**Figure 4-7** GA Cycle.

#### 4.3.2.1 Arm Motion Objective Functions

Four different objective functions are chosen to generate the arm motion for SOGA and for MOGA only three objective functions are selected to be optimized. The four criteria are minimum execution time, minimum distance, minimum acceleration and minimum angular acceleration. These objective functions are chosen based on the similarity of human motion while performing daily life task.

#### 4.3.2.2 Minimum Execution Time (MT)

Humans move their arms in many different ways depending on the task. For a simple motion of picking a light object such as picking a pen, or moving freely as waving to someone, humans tend to move in the shortest time. Based on this situation,

the first chosen criterion is the minimum execution time. This objective function will be minimizing the robot hand execution time when it moves from the starting point to the goal position. In our system, the sampling time to process the sensors data and send the motor command is 0.03 second. Therefore, the objective function is to minimize the number of step,  $n_{step}$  for the robot to reach the goal position. The first objective function for the arm motion is as follows.

$$f_1 = n_{step} \quad (9)$$

#### 4.3.2.3 Minimum Distance (MD)

Another important characteristic in human arm motion is the shortest distance. Normally human move their arms in this manner especially for a specific task, such as drawing a straight line, arranging books and pushing an object. The trajectory connecting the initial and goal positions must be the shortest one and this is the reason minimum distance is selected to be one of the objective functions. The minimum distance objective function is as follows:

$$f_2 = \left| \sum rt_i - sd \right| \quad (10)$$

where  $\sum rt_i$  is the summation of robot hand moving distance in each time step and  $sd$  is the shortest distance of the robot hand from its initial position to the goal.

#### 4.3.2.4 Minimum Acceleration (MA)

For a more complicated task such as moving a cup of tea or a non-rigid object, human will move the arm in a constant velocity and slower speed. A gradually increasing velocity in the beginning and gradually decreasing velocity toward the goal position are required to have a smooth and stable motion. This motion characteristic is chosen as one of the objective function for the robot hand motion. The total acceleration of the robot hand is minimized to have a constant velocity. Two penalty functions are also introduced, one is for the robot to have a gradually deceleration before reaching the goal position and second, minimizing the number of velocity change for a smooth motion throughout the trajectories. Therefore, the minimum acceleration objective function is as follows:

$$f_3 = \sum a_{hand} + (v_{hand\_end} * w) + (nvc * w) \quad (11)$$

where  $\sum a_{hand}$  is the summation of robot hand acceleration in each time step,  $v_{hand\_end}$  is the robot hand velocity when it approaches the goal position,  $w$  is the weight function and  $nvc$  is number of velocity changes. The number of velocity changes is very important in order to minimize the rapid changes of the robot hand velocity in each time step. The weight function ( $w$ ) is used to adjust the priority between  $\sum a_{hand}, v_{hand\_end}$  and  $nvc$ . In the first motion generation, the value of  $w$  is set to be 1, and once the value of each term is known,  $w$  can be determined. In our implementation the value of  $w$  used is 60.

#### 4.3.2.5 Minimum Angular Acceleration (MAA)

A similar objective function with different approach is chosen by minimizing the joint angular acceleration. The total joint acceleration is minimized to have a slow robot hand motion while performing a task. Therefore, the following fitness function is also considered:

$$f_4 = \sum \alpha_1 + \sum \alpha_2 + \sum \alpha_3 \quad (12)$$

where  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$ , is the robot angular acceleration for shoulder, upper arm and lower arm respectively.

### 4.3.3 Robot Navigation

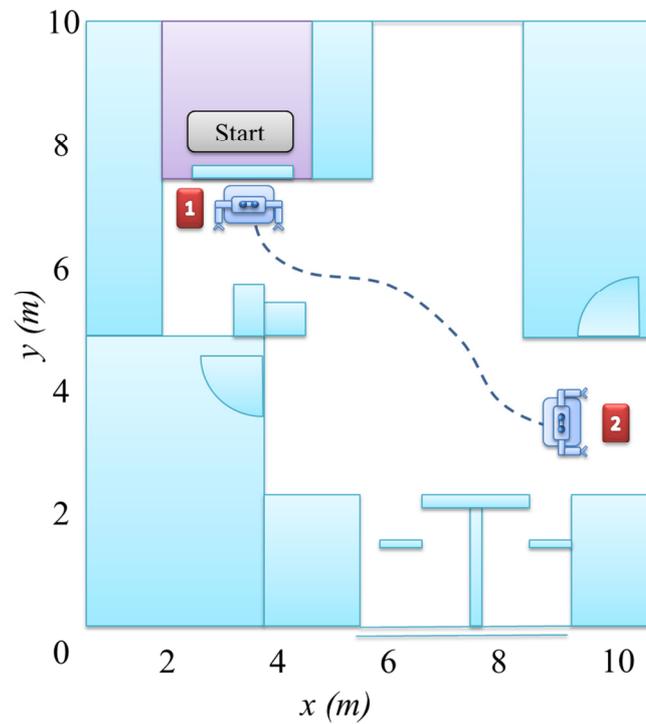
In this thesis, three fitness functions are generated for the robot navigation and have the same objective of minimizing the traveling distance from the starting to the goal position. The overall navigation process is divided into three sections to reduce the generation time.

#### 4.3.3.1 Environment 1

In the first environment, the robot is required to move from the elevator (point 1) to the middle of the hallway (point 2) in a shortest distance while avoiding obstacles along the way as in Fig. 4-8. The fitness function utilised in this environment is as follows:

$$f_5 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} + \frac{sensor_{side}}{n_{step}} \quad (13)$$

where  $x_1$  and  $y_1$  is the position of the robot at point 1,  $x_2$  and  $y_2$  is the position of the robot at point 2,  $sensor_{side}$  is the differences between the left and the right reading of LR1 and  $n_{step}$  is the number of step. This fitness function will minimised the distance between the two points and make sure the robot stay in the middle of the hallway.



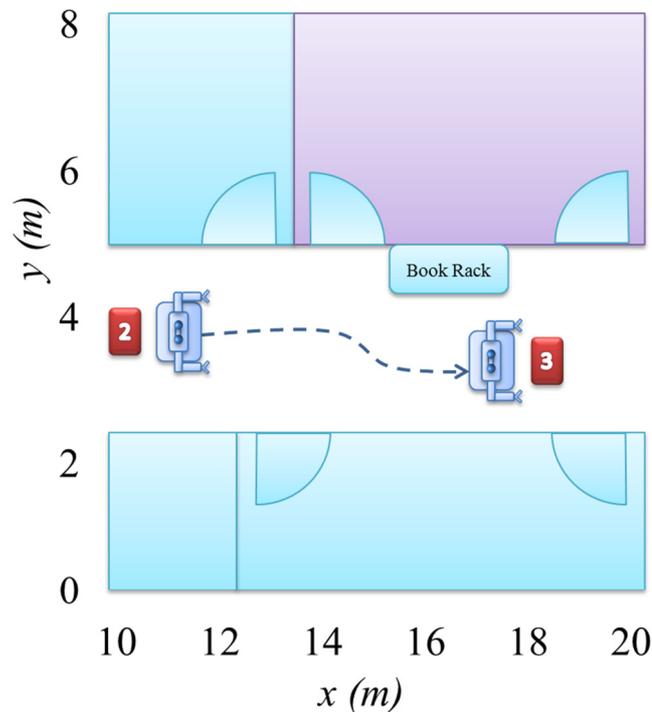
**Figure 4-8** Robot navigation in environment 1.

#### 4.3.3.2 Environment 2

In environment 2, the robot has to manoeuvre from the starting point of the hallway (point 2) to the entrance of the goal position (point 3) as in Fig. 4-9. The shortest distance between two points in the horizontal  $x$  direction is optimized and the sensor data is used for obstacles avoidance and guiding the robot to be in the centre line of the hallway. The fitness function for environment 2 is as follows.

$$f_6 = x_3 - x_2 + \frac{sensor_{side}}{n_{step}} \quad (14)$$

where  $x_2$  and  $y_2$  is the position of the robot at point 2,  $x_3$  and  $y_3$  is the position of the robot at point 3 (the goal position entrance).



**Figure 4-9** Robot navigation in environment 2.

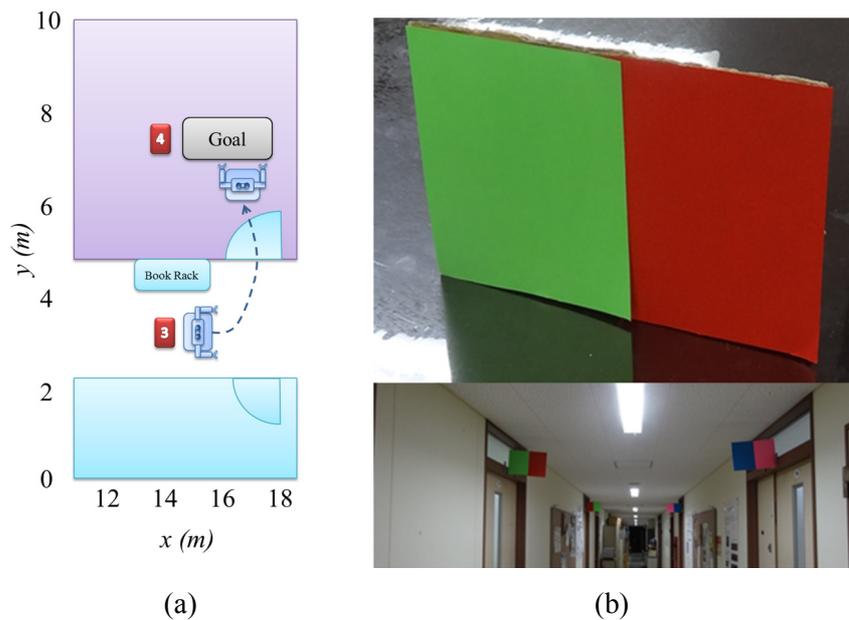
#### 4.3.3.3 Environment 3

The robot will determine the desired entrance door to the goal position based on the landmark placed on top of the door. This is similar as in a hospital or a care centre where each room will have the room number or its own specific landmarks. In our implementation, a colour landmark is utilised as in Fig. 4-10 and it will be detected using the robot camera. The size of the colour landmark is 180 mm x 320 mm and it is

place 2 m from the floor. The fitness function for the neural controllers is shown below:

$$f_7 = \sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2} \quad (15)$$

where  $x_3$  and  $y_3$  is the position of the robot at point 3,  $x_4$  and  $y_4$  is the position of the robot at point 4 (the goal position). This function is optimizing the distance between point 3 and 4 while avoiding the door and table inside the room. The door size is 900 mm and it is slightly narrow for the robot to enter with 550 mm width.



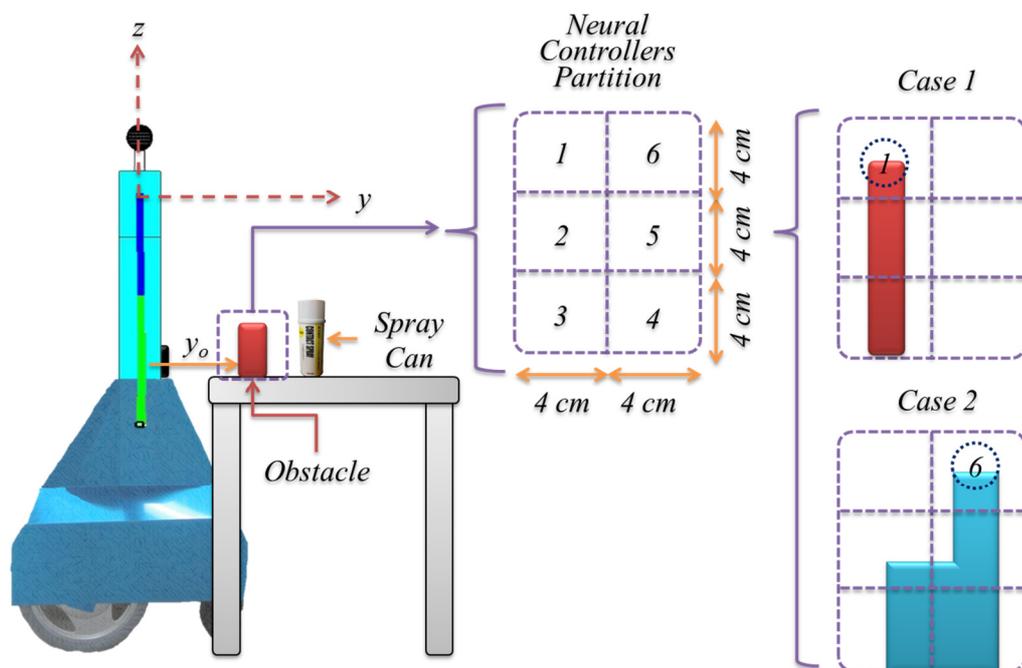
**Figure 4-10** (a) Robot navigation in environment 3 (b) Colour landmark.

#### 4.4 Arm Motion Generation in Dynamic Environment

Another important issue is motion generation in dynamic environments where the robot has to avoid obstacles while reaching the target position. To address this

problem, we divided the area in the lateral plan in 6 parts, and for each part we pre-evolved a neural controller for each objective function (Fig. 4-11). Based on the humanoid robot data, the maximum height of the obstacle detection region is considered 12 cm (3x4cm) and the maximum width 8 cm (2x4 cm).

At this stage of our work, we presume that the obstacle dimensions and position are determined using the camera and laser sensors. Based on the obstacle size and position, the robot determines the partitions that the obstacle covers. The specific neural controller is selected to generate the arm motion that avoids the obstacle while reaching the target position. For example, in case 1 (Fig.4-11) the neural network will control the robot hand to move over the partition 1, therefore avoid hitting the obstacle. In the case 2 obstacle, the partition 6 neural controller would be selected.



**Figure 4-11** Obstacle detection regions for different obstacle shape.

The trajectory generated by the neural controller passes over all the partition. In the case of minimum distance, the shorter trajectory would be to move near to the obstacle. Although the hand trajectory is not the best, this is one way to deal with some errors in determining the obstacle size and position.

---

# CHAPTER 5

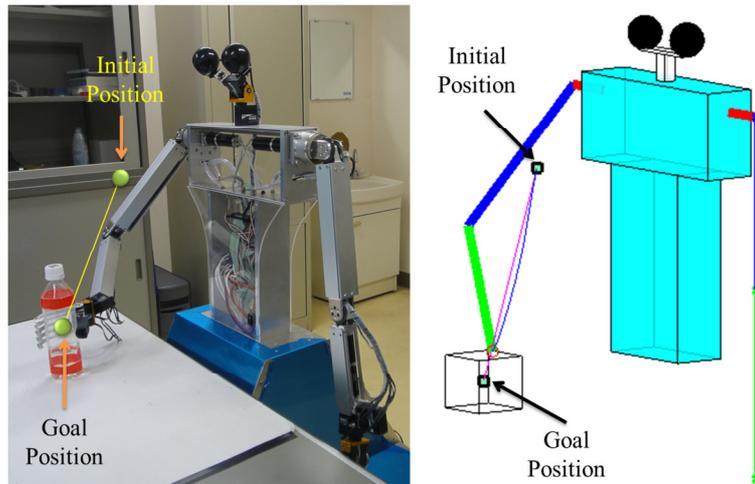
---

## 5 SIMULATION AND EXPERIMENTAL RESULTS

The neural controllers generated based on the selected tasks discussed in the previous chapter are tested in simulated environment before it can be implemented on the real robot for safety reason and comparison. This chapter will discussed further all the findings and results. Single and multi-objective optimization of the robot arm motions are discussed in Section 5.1 and Section 5.2, respectively. Section 5.3 discussed the performance of the generated neural controllers in dynamics environment. In the last section, the robot performance is tested with a simple navigation throughout the laboratory environment.

### 5.1 Single Objective Optimization

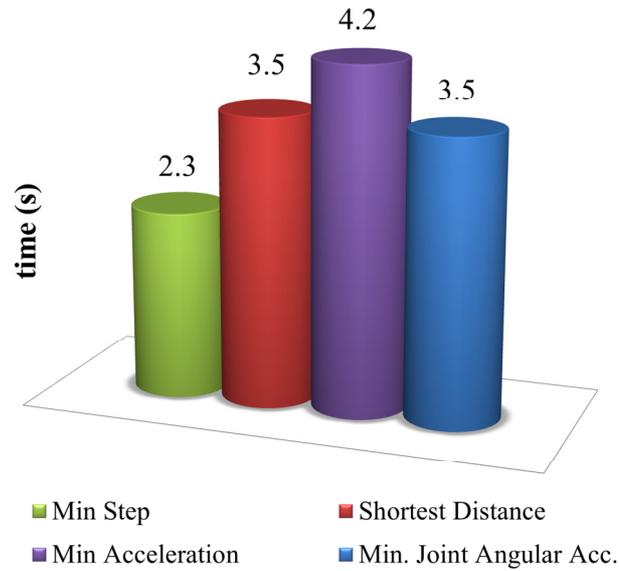
In the first experiment, the performance of the robot arm motion is tested. Four objective functions are selected based on human arm motion characteristics namely; minimum execution time, minimum distance, minimum acceleration and minimum angular acceleration. These objective functions are optimized separately using SOGA. Before the performance of the robot is tested in a complex motion, a simple placing motion is chosen. In this experiment, the robot arm is required to perform a placing motion from a holding position as placing a bottle on the table (Fig. 5-1) (Mohamed, Mano, et al. 2013b).



**Figure 5-1** Experiment 1: Placing a bottle on the table.

### 5.1.1 Simulation Results

The performance of the best neural controller generated for each objective function is shown in Fig. 5-2 in terms of its execution time, Fig. 5-3 for travelling distance and Fig. 5-4 for total velocity. Fig. 5-2 shows that MT neural controller reached the target position the fastest in 2.3 second, while MA neural controller is nearly 2 second slower than MT neural controller. For MAA and MD neural controllers, the robot hand reached the goal position within 3.5 second. Having the fastest time in performing the task will reduce the performance of the neural controller in terms of its travelling distance and total velocity as in Fig. 5-3. If minimum distance and total accelerations are required in the arm motion such as moving a glass of water, MT neural controller is not suitable to be adapted to the robot hand motion. This neural controller is more suitable for simple motion such as moving the arm freely or moving the arm toward the manipulated object.

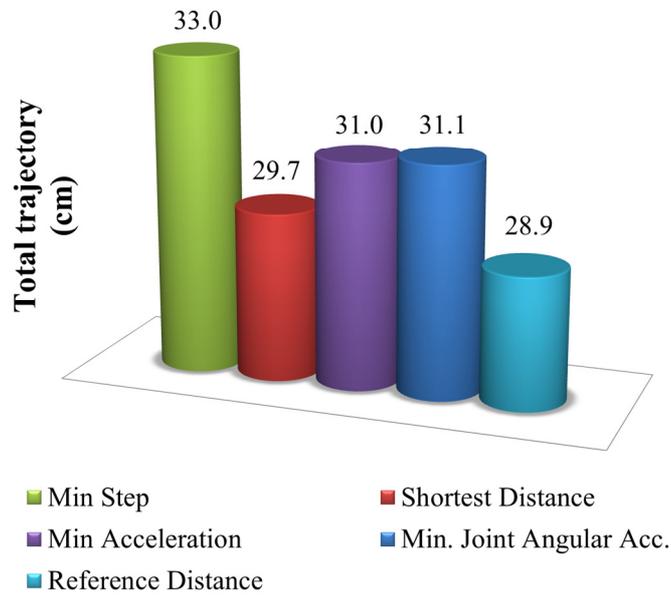


**Figure 5-2** Execution time comparison for each objective function.

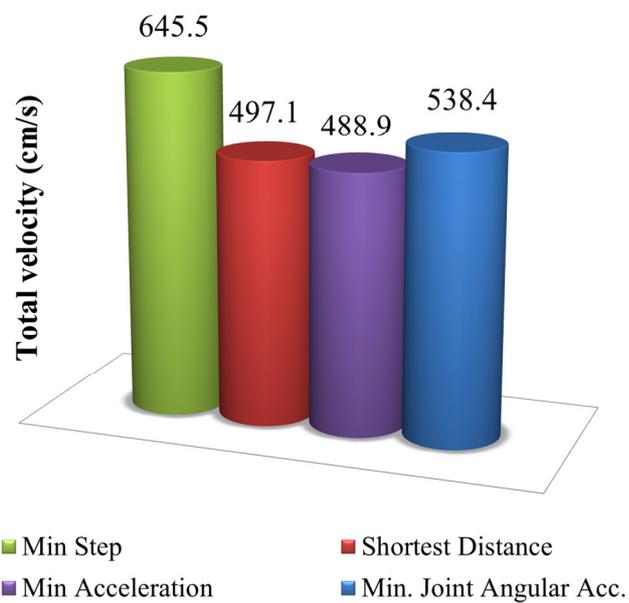
Fig. 5-3 shows an interesting result, the minimum time and minimum distance trajectories show a great difference in the total trajectory. The shortest distance between the initial and goal position is 28.9 cm and it is set to be the reference distance. The minimum time neural controller reached the goal position following the longest trajectory because the change in the hand velocity is included as a penalty function in minimum distance and minimum angular acceleration. MA and MAA objective functions show a similar performance reaching the goal position with a small different of 1 mm.

Fig. 5-4 shows that there is not too much change in the fitness with minimum acceleration of the robot hand. The total velocity for the whole trajectory is 488.91 cm/s. The performance of constant joint angular velocity criterion is slightly lower with 538.45 cm/s. These simulations results show that all four criteria perform accord-

ingly to their objective functions. This is proven that the proposed method and the objective functions perform according to the desired characteristics.



**Figure 5-3** Total trajectory comparison for each objective function.



**Figure 5-4** Total velocity comparison for each objective function.

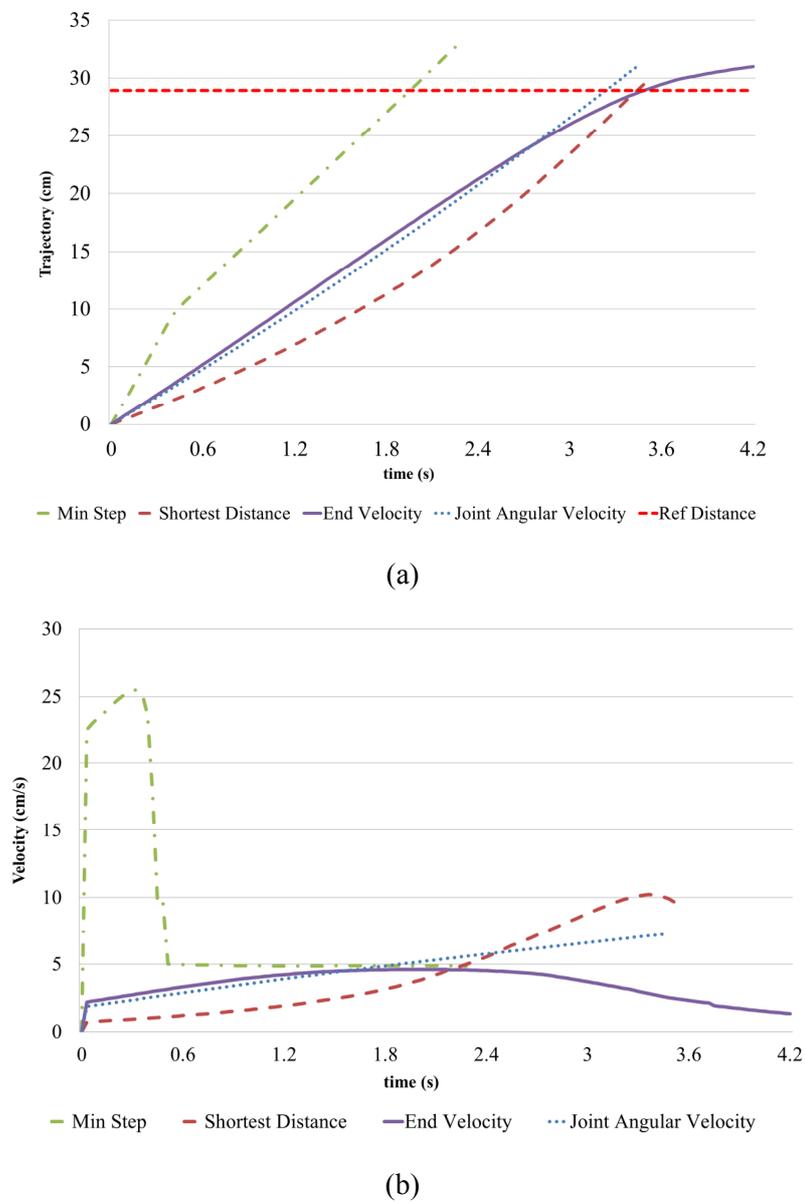
Fig. 5-5(a) and Fig. 5-5(b) show the robot hand trajectories and velocity profile for each objective function. The trajectory distance (Fig. 5-5(a)) show that MD objective functions has the closest result to the reference distance but it takes longer time to complete the task. Even though MA objective function finished the last with 4.2 second, but it has the highest stability in reaching the goal position. From the velocity profile (Fig. 5-5(b)) minimum step criterion has the lowest stability in the motion due to high velocity at the beginning of the motion. The other three criteria have low initial velocity and it increases gradually until it the arm reach the goal position. Only MA objective function reaches the goal in gradually reduced velocity due to the implementation of penalty function to the objective function. The minimum acceleration of the robot hand generated the robot motion which reaches the goal position with a small velocity compared to other objective functions. From these results, the robot can intelligently decide which motion to be adapted based on the task to be performed.

These objective functions have their own advantages and disadvantages over each other. If the robot hand need to move fast, the minimum time neural controller is best solution, for stability, robot hand acceleration and constant joint angular velocity neural controllers can be used. For high accuracy, shortest distance will be the best objective functions. These results will be compared with multi-objective motion generation in the next experiment (Section 5.2) to show the advantages and disadvantages of both solutions.

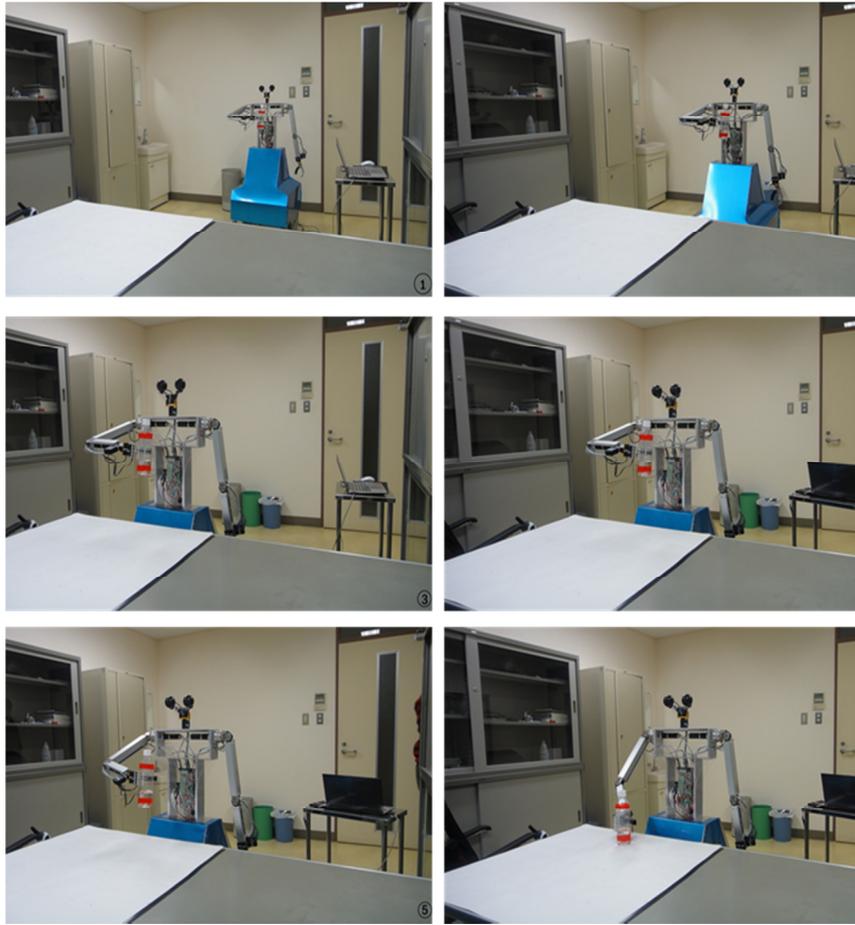
### 5.1.2 Experimental Results

In the implementation of the optimal neural controller on the real robot, first, simple robot navigation is added to the task. The initial position of the robot is slightly

far from the table and the robot utilizes the laser and camera sensors data to reach the table. Once the robot reaches the table, the bottle is required to be placed on the table. The video capture of the experiment with the humanoid mobile robot is shown in Fig. 5-6. The robot arm motion is generated by four optimal neural controllers in order to compare the performance. The same motion is also repeated without holding the bottle to compare the results.



**Figure 5-5** Comparison of robot hand (a) Trajectory (b) Velocity profile.

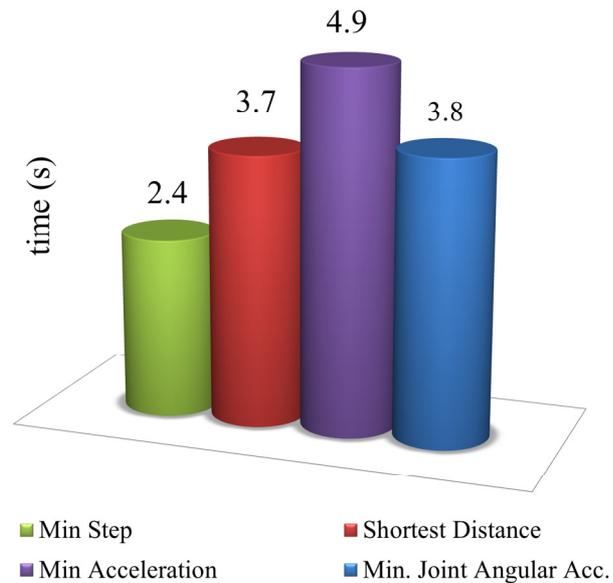


**Figure 5-6** Video capture of the experiment.

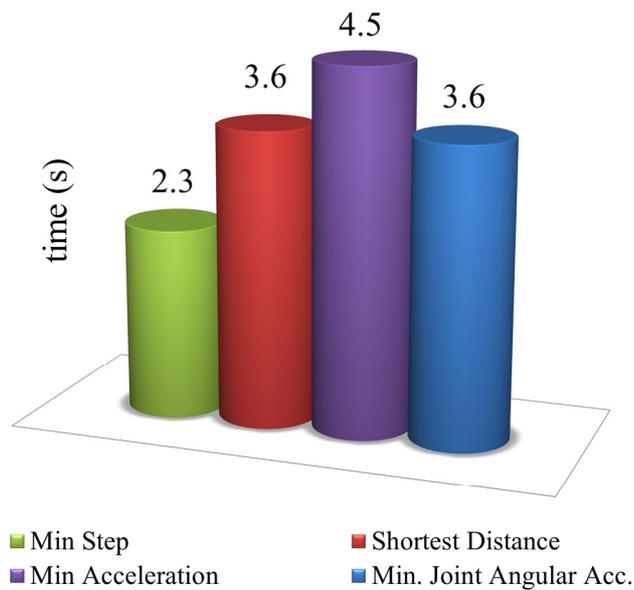
The time needed for the robot hand to reach the table for each objective function for motion without holding the bottle and while holding the bottle is shown in Fig. 5-7(a) and Fig. 5-7(b), respectively. Both results show similar performance with a small fraction of time difference due to the gravitational effect on the motion going downward.

In simulation, for MT objective function, the minimum execution time was 2.28 second, while in the real robot it took 2.43 second for motion without the bottle and slightly different (2.38 second) while holding the bottle. Similar results are shown for the other three objective functions, where the motion is faster while holding the

bottle. The performance of the robot manipulating the bottle is comparable with the simulation results in terms of its execution time. The execution time comparison between the simulation and two experimental setups (with and without bottle) for each objective function are shown in Fig. 5-8.

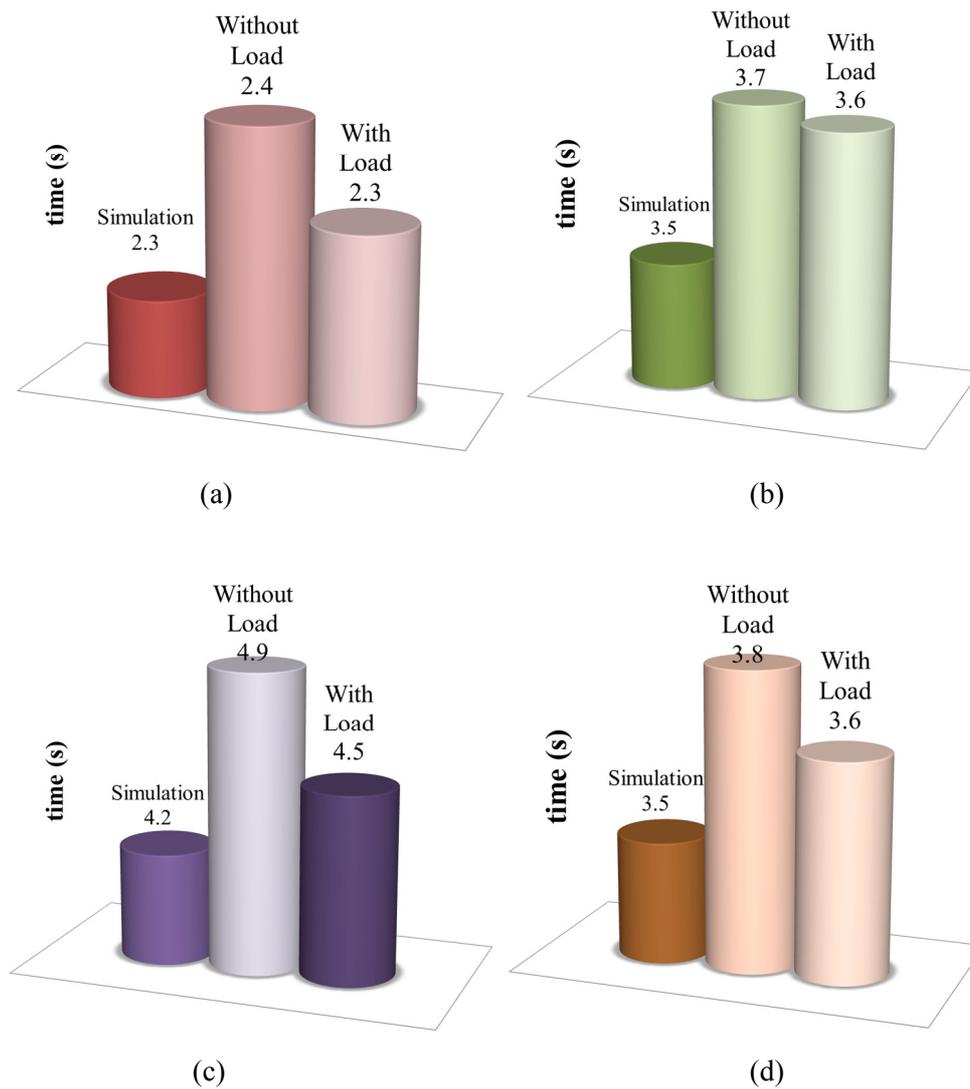


(a)



(b)

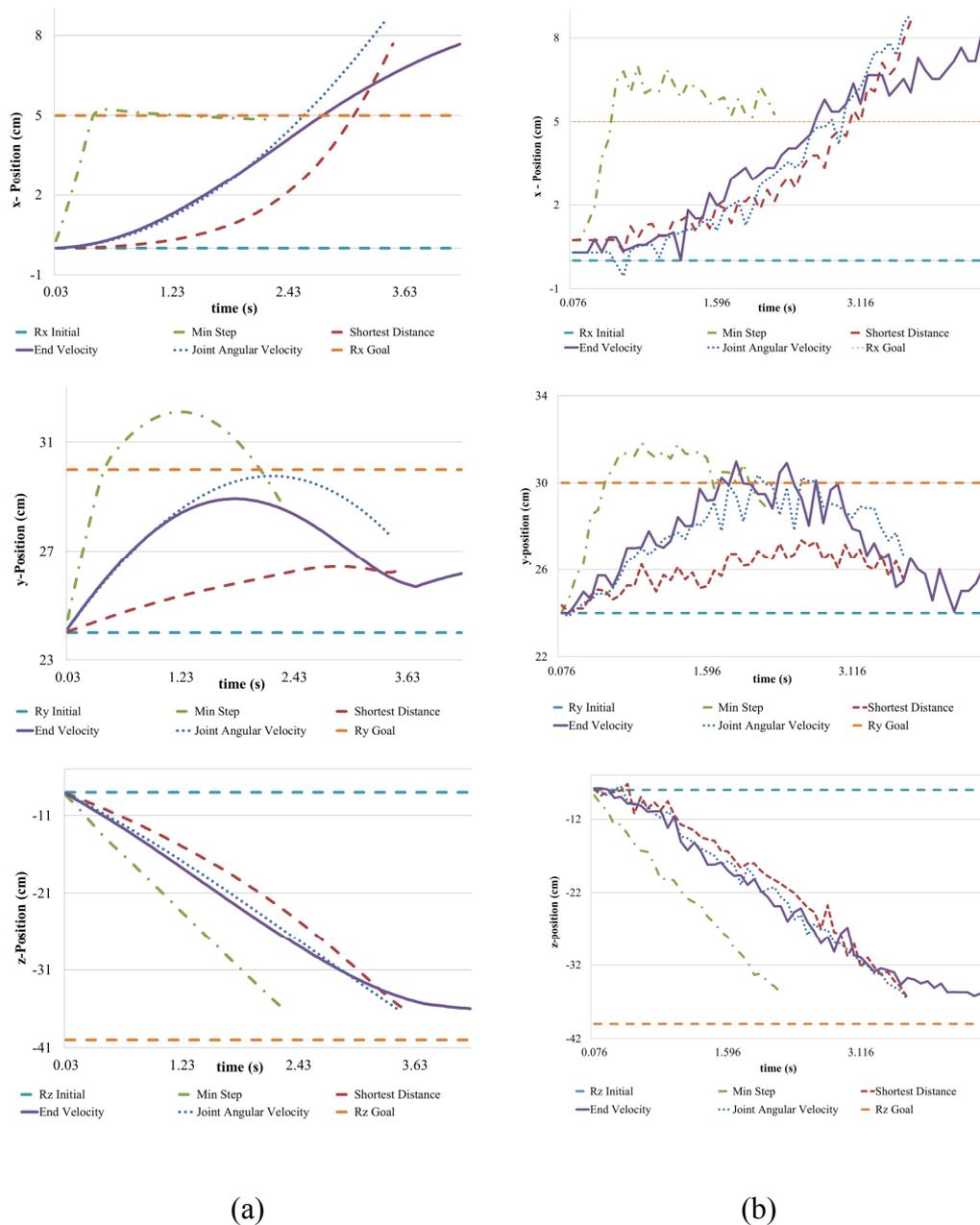
**Figure 5-7** Task execution times for robot hand (a) Without bottle (b) With bottle.



**Figure 5-8** Execution time comparison for simulation, without bottle and with bottle for (a) MT (b) MD (c) MA (d) MAA.

In simulation, the external conditions are not considered such as joint friction, mechanical gear backlashes, motor gearhead backlashes, gravitational effect, mechanical design and the movement of the water inside the bottle, thus having a smooth motion and faster execution time. In the experiment, these factors are affecting the performance of the robot. In terms of time taken, there are small differences for all objective functions. It can be seen that the motion is slightly faster while holding the

bottle due to higher gravitational force. Fig. 5-9(a) and Fig. 5-9(b) show the comparison of the robot hand trajectory in  $x$ ,  $y$  and  $z$  axis for simulation and experiment with load (bottle), respectively. A similar comparison between simulation and experiment results (without load) is shown in Appendix C.



**Figure 5-9** Execution time comparison for each objective functions in  $x$ ,  $y$  and  $z$  -axis  
(a) Simulation (b) With load.

All three axis motion of the right hand are measured and compared. In simulation, the trajectory and the angle movement of each joint are very smooth (Fig. 5-9(a)) for all objective functions, moving from its initial to the goal position. In the real robot, the generated motion follows the same simulation trajectory with some deterioration in the motion smoothness for all objective functions. Although the performance of the robot is not as good as the simulations while holding the bottle, the robot follows nearly the same trajectory generated in the simulated environment and successfully reaches the goal position. The deterioration is due to the movement of the water inside the bottle while the robot hand is in motion. In this experiment the performance of the robot in simulation and on the real robot, show good results, where the SOGA arm motion generation shows the desired output minimizing each objective function according to desired characteristics.

## 5.2 Multi Objective Optimization

### 5.2.1 Two Objective Optimization

In the second experiment, the robot arm motions are generated optimizing two objective functions simultaneously, where three combinations of objective functions have been chosen and the combinations are; MT-MD, MT-MA and MD-MA. A similar motion of placing an object on the table as in the previous experiment is adapted. In this experiment, the performance of multi-objective arm neural controllers is tested in simulated environment and on the real robot (Mohamed, Kitani, et al. 2013b).

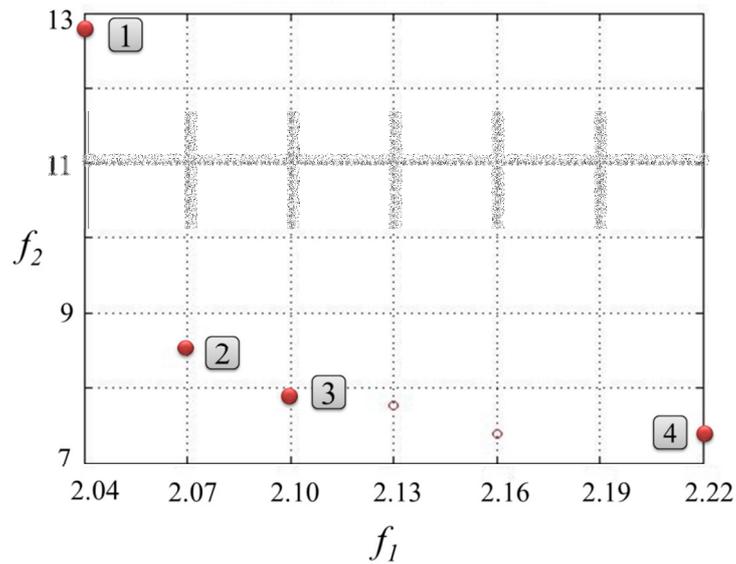
#### 5.2.1.1 Simulation Results: Minimum Time & Minimum Distance: ( $f_1$ - $f_2$ )

The non-dominated optimal front of 80<sup>th</sup> generation has clear trade-off between both objective functions as in Fig. 5-10. Six individuals in Pareto set have been gener-

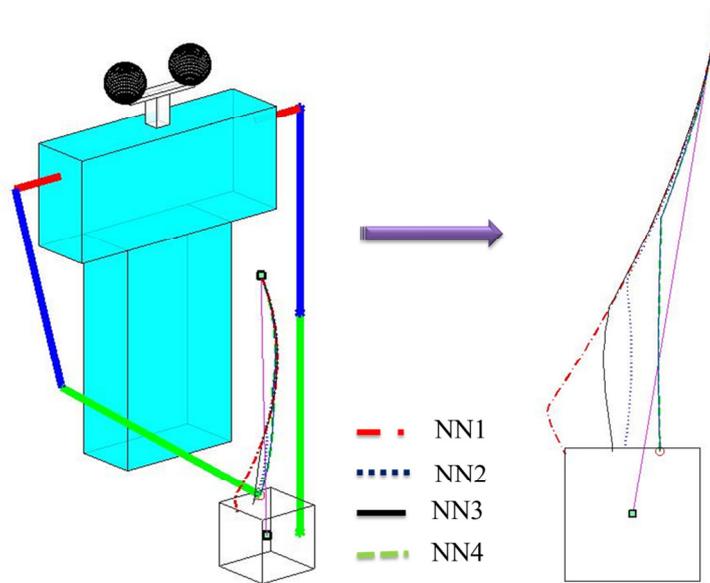
ated and four individuals (NN1, NN2, NN3 and NN4) are chosen to be further discussed. NN1 and NN4 are the extreme solution very similar with the single objective GA for minimum time and minimum distance of the robot motion, respectively. The comparison between these individuals is to show the characteristics and differences between each solution and later can be chosen as one of the arm motion solution based on specific motion characteristics.

NN2 and NN3 are neural controllers that simultaneously optimize both objective functions. NN2 and NN3 show significantly optimal solutions in this experiment by minimizing the time without much deterioration of the distance and vice versa. With an increase in travel time by 0.03 second, the difference between NN1 and NN2 is 4.3 cm. These solutions are suitable if the robot arm motion is required to be minimizing the time and distance simultaneously.

Fig. 5-11 illustrates the motion generated using the neural controllers NN2 and NN3 and the trajectories are improved compared to NN1. The robot hand move closer to the shortest distance for NN2 and NN3 solution compared to NN1 (similar to MT objective function) where the robot hand move away from the shortest trajectory to reach the goal. NN4 solution is similar to shortest distance solution and it shows the shortest path and distance as in Fig. 5-11. The minimum time for MT and MT-MD is similar but for MT-MD, it satisfies both minimum time and minimum distance. With these results, we can choose the motion criteria for the robot arm to perform the task based on our requirement; NN1 for the robot arm moves in minimum execution time, NN4 for minimum distance, NN2 or NN3 for motion satisfying both objective functions.



**Figure 5-10** Pareto fronts of MT-MD objective functions.



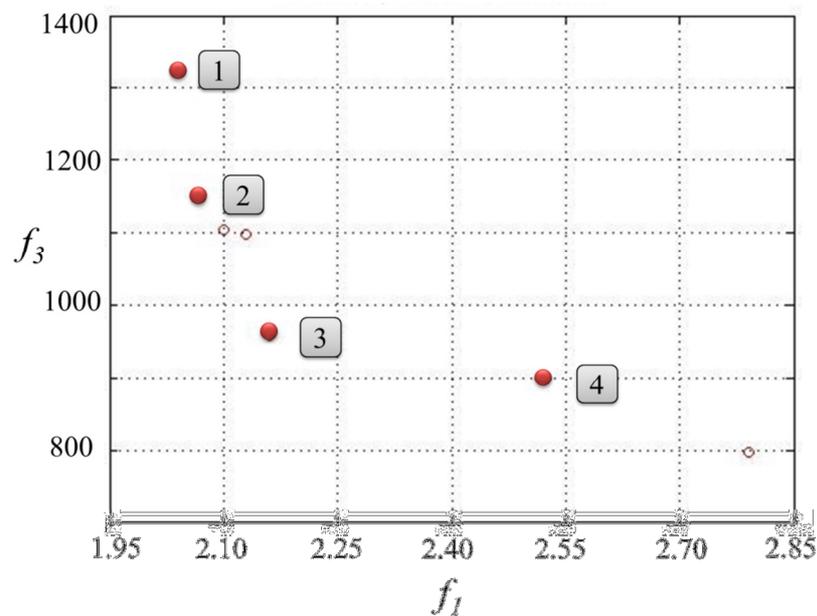
**Figure 5-11** Robot arm motion for NN1, NN2, NN3 and NN4 neural controllers of MT-MD objective function.

#### 5.2.1.2 Simulation Results: Minimum Time & Minimum Acceleration: ( $f_1$ - $f_3$ )

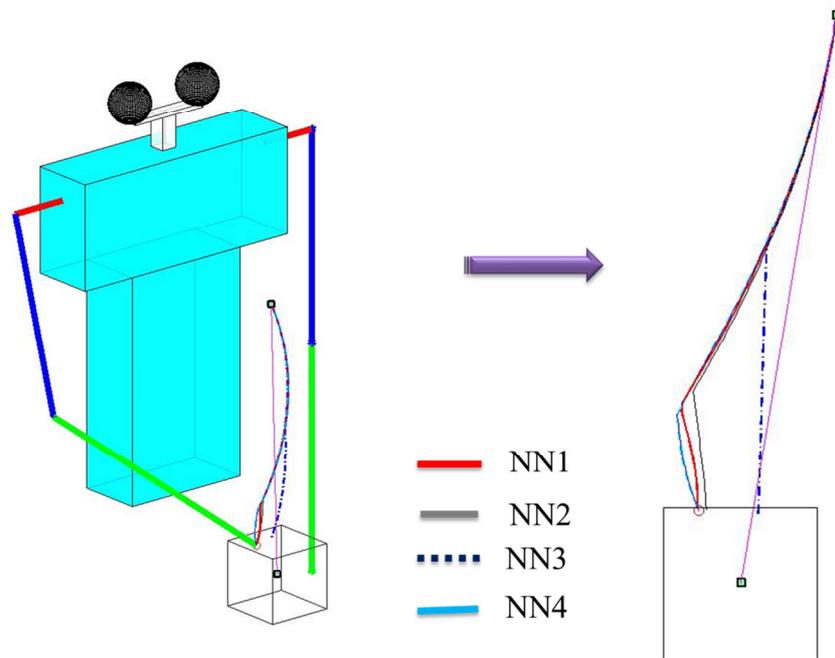
Pareto front of MT-MA objective function is shown in Fig. 5-12. The Pareto front has seven individuals with 80 maximum iteration of the MOGA. Four NNs are

selected from the Pareto front to be compared in terms of its performance and motion characteristics. Based on the simulation, NN3 generated better trajectory and performance compared to the other three NNs. The difference between NN2 and NN3 in terms of  $f_3$  is significantly large for a small difference in  $f_1$ . A similar comparison can be done to NN3 and NN4, with small difference in  $f_3$ , the difference in  $f_1$  is large.

Fig. 5-13 presents the robot hand motion for both for all four solutions. Clearly shown that the motion generated by NN3 solution perform the best. The motion of the robot hand is closed to the shortest distance trajectory. The other three neural controllers show a different solution where the motion of the robot hand is moving away from the shortest path trajectory. By selecting different solutions, the motion criteria can be chosen depending on the task that the robot arm needs to perform; NN1 for minimum execution time, NN4 for minimum acceleration and NN3 for both objective functions.



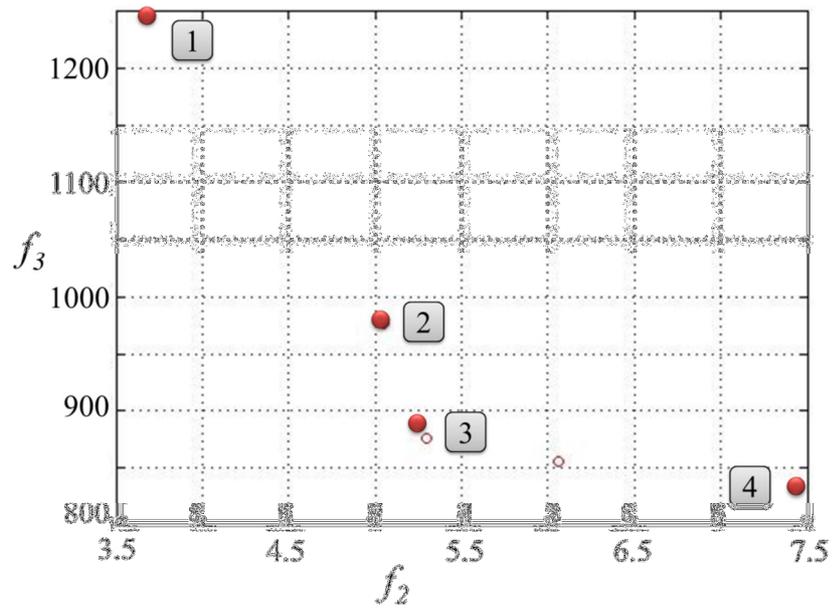
**Figure 5-12** Pareto fronts of MT-MA objective functions.



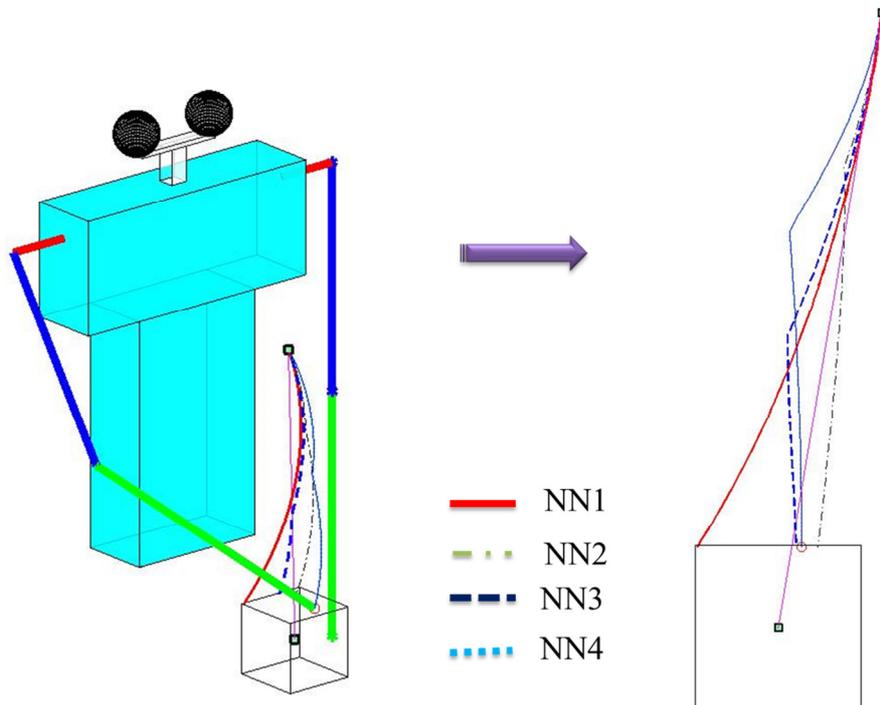
**Figure 5-13** Robot arm motion for NN1, NN2, NN3 and NN4 neural controllers of MT-MA objective function.

### 5.2.1.3 Simulation Results: Minimum Distance & Minimum Acceleration: ( $f_2$ - $f_3$ )

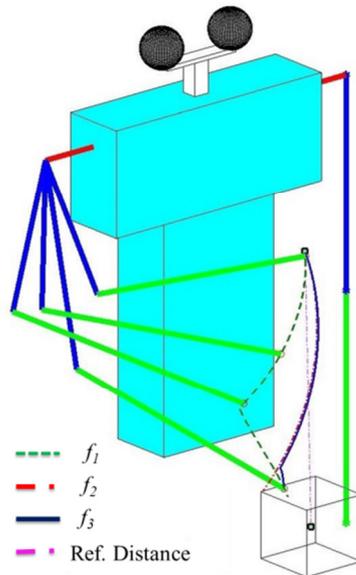
The third generated Pareto front for MD-MA objective functions are presented in Fig. 5-14 and six individuals in Pareto set are generated with 80 maximum iteration of the MOGA. It shows clear trade-off between both objective functions where NN1 show that GA gives higher priority to  $f_3$  and NN4 to  $f_2$ . Having a better trajectory and performance, NN2 and NN3 solutions is applied to the simulated humanoid robot Fig. 5-15. These results prove the advantage of MOGA compared with SOGA. With a very small deterioration in one objective function, we can achieve a significant improvement of the other objective function.



**Figure 5-14** Pareto fronts of MD-MA objective functions.



**Figure 5-15** Robot arm motion for NN1, NN2, NN3 and NN4 neural controllers of MD-MA objective function.



**Figure 5-16** Robot arm motion for  $f_1$ ,  $f_2$  and  $f_3$  neural controllers of single objective function neural controllers.

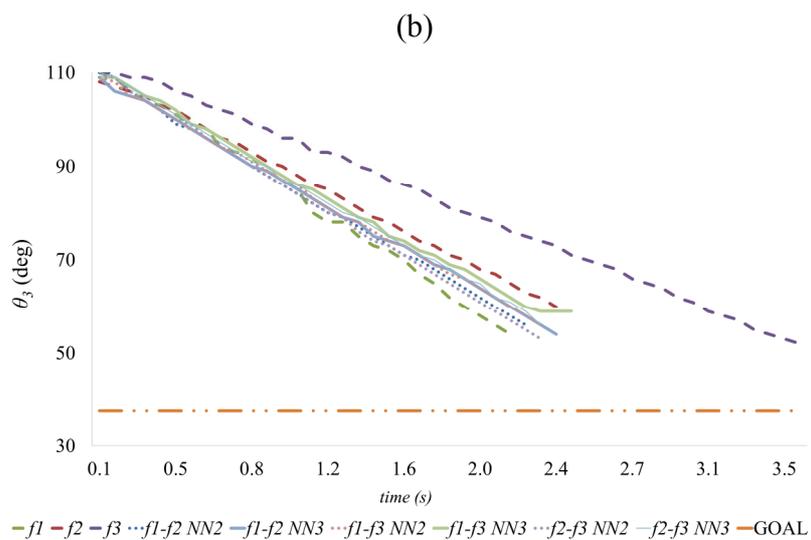
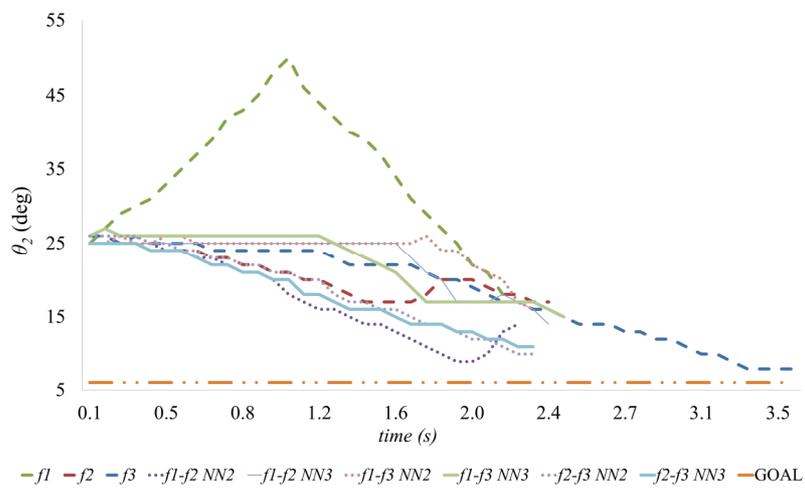
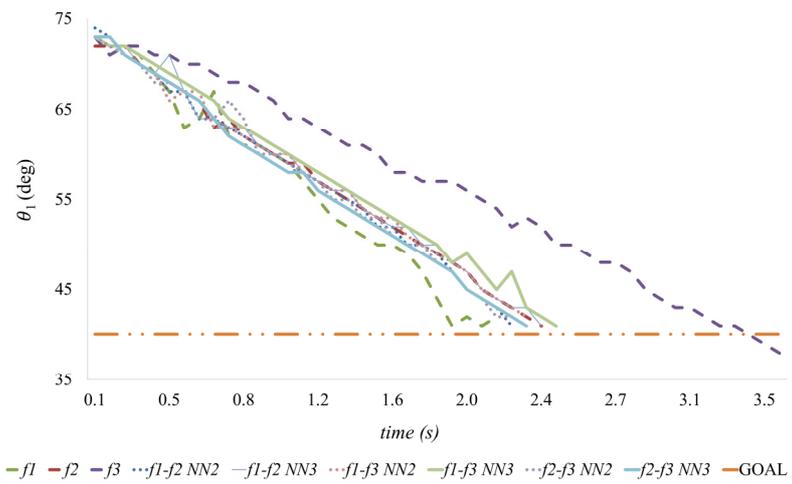
Single objective robot arm motion generation adapted from the previous experiment is shown in Fig. 5-16. The MT objective function ( $f_1$ ), shows the lowest performance in trajectory distance, where the robot arm move away from the shortest path. MD ( $f_3$ ) neural controller has the closest trajectory to the shortest path. These arm SOGA arm motions has similar performance as the three combinations discussed before, but SOGA solution only minimizing a single objective function in the motion, while the three combinations has two objective optimization in a single motion. In section 5.2.2, three objective optimization problems will be discussed.

#### 5.2.1.4 Experimental Results: Two Objective Optimization

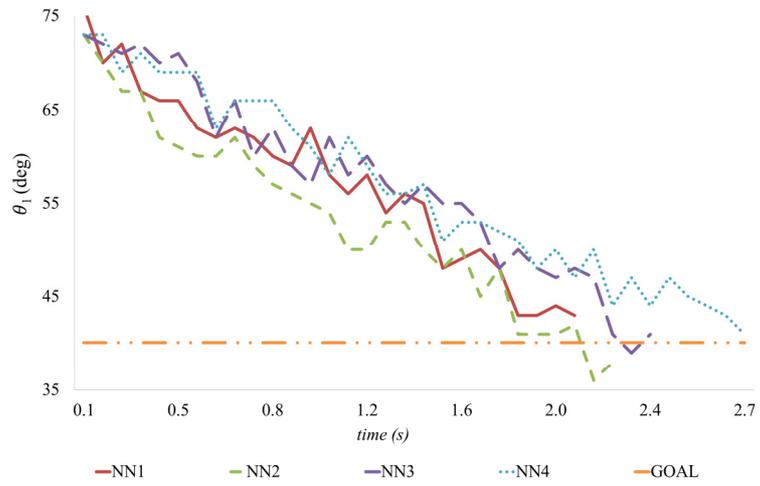
The robot hand performance is evaluated by adapting the three set of two objectives optimal neural controllers to the real robot. The results obtained for solution NN2 and NN3 of the Pareto optimal front in Fig. 5-10, Fig. 5-12 and Fig. 5-14 are pre-

sented in Fig. 5-17(a), Fig. 5-17(b) and Fig. 5-17(c). The three joint displacements of the robot arm ( $\theta_1$ ,  $\theta_2$  and  $\theta_3$ ) is compared between SOGA and MOGA as in Fig. 5-17. The experimental results show that robot hand adapting two objectives GA perform better than SOGA. All three combinations have significantly similar performance regarding the minimum time, minimum distance and minimum acceleration compare to single objective GA. For  $\theta_1$ , the joint 1 angular position of NN2 and NN3 solution are in between the single objective function as in Fig. 5-17(a). Instead of having a single objective optimization, the robot hand could have two objective functions to be optimized in a single generation without compromising the overall performance. Joint 2 and joint 3 angular positions as in Fig. 5-17(b) and Fig. 5-17(c) show a similar results. MOGA neural controllers show the best solution optimizing the two objective functions without compromising the execution time.

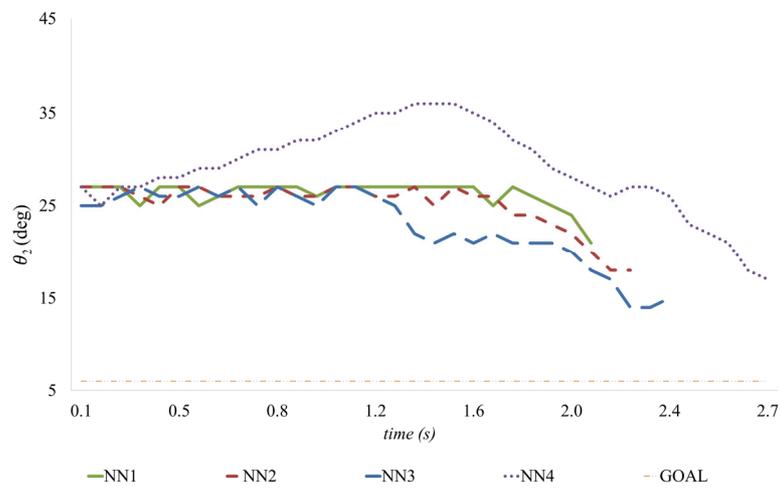
The performance of multi-objective GA is further tested by adapting all four NN1, NN2, NN3 and NN4 of MT-MA (Fig. 5-12) objective functions to the mobile robot. In this experiment, the robot hand move with the generated neural controllers while holding a bottle of water. The experimental results show that differences in characteristic of robot motion for each generated NN. Fig. 5-18(a), Fig. 5-18(b) and Fig. 5-18(c) are the joint 1, joint 2 and joint 3 angular position comparisons between all four selected NN. Performance of NN2 and NN3 are better than NN1 and NN4 not only on MT and MA objective function but also reaching to the goal position. NN4 solution takes the longest time to complete the task and NN1 the fastest. Although NN2 and NN3 neural controller are not the fastest or the slowest solutions, with a small differences in execution time, both NN optimizing two objectives function simultaneously.



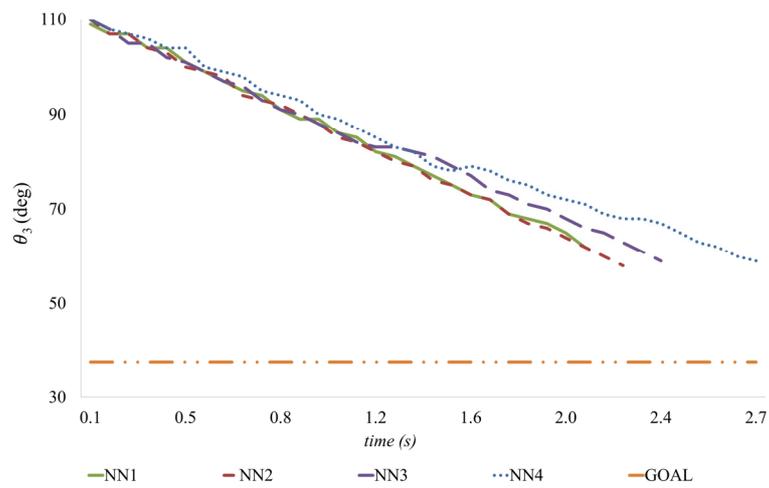
**Figure 5-17** Robot arm joint trajectories for (a)  $\theta_1$  (b)  $\theta_2$  (c)  $\theta_3$ .



(a)

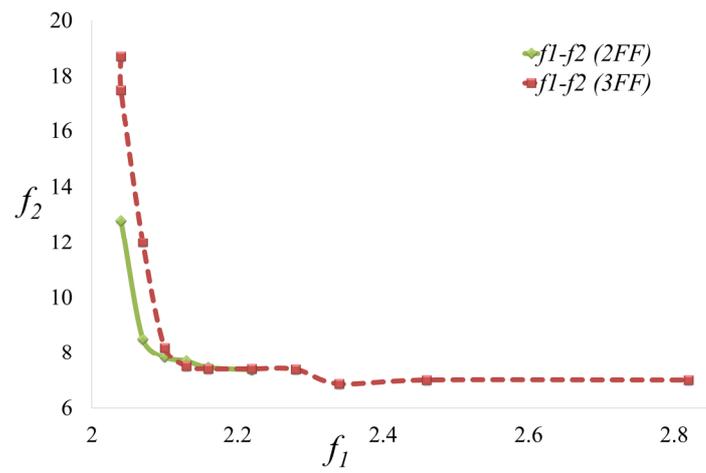


(b)

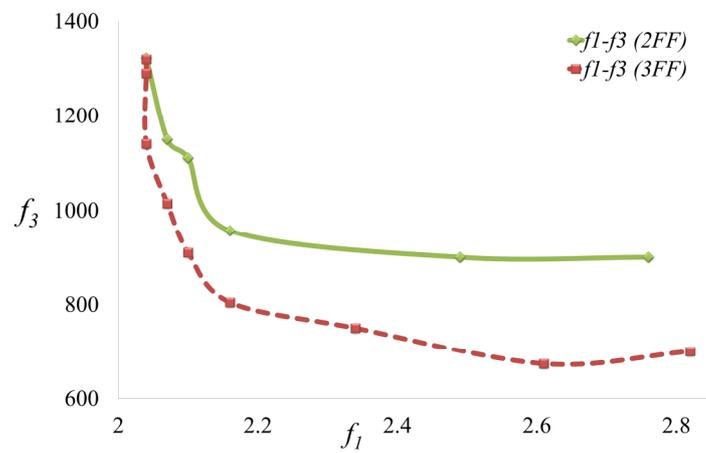


(c)

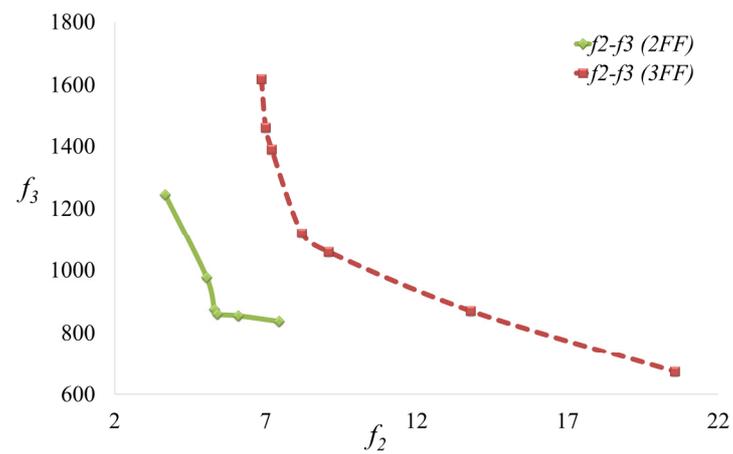
**Figure 5-18** MT-MA joint trajectories for (a)  $\theta_1$  (b)  $\theta_2$  (c)  $\theta_3$ .



(a)



(b)



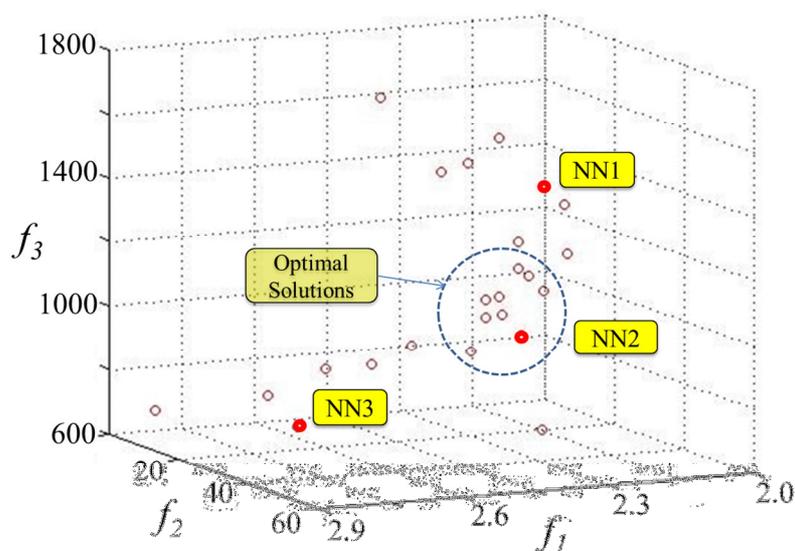
(c)

**Figure 5-19** Pareto comparison of two and three objective functions optimization for  
(a) MT-MD (b) MT-MA (c) MD-MA.

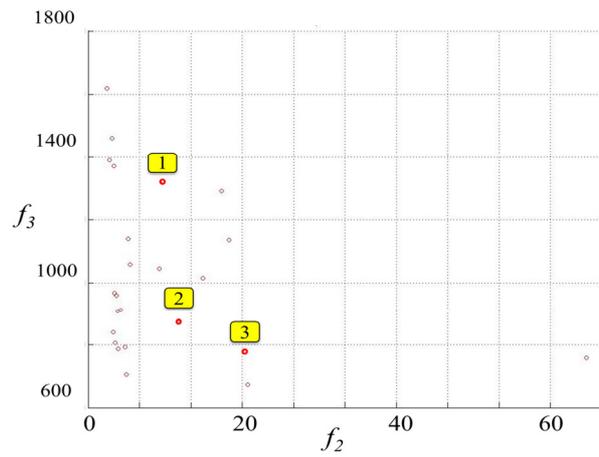
The comparisons of Pareto front obtained by adapting two and three objective optimization are presented in Fig. 5-19(a), Fig. 5-19 (b) and Fig. 5-19 (c). Simulation results show that, generating two objective functions using MOGA perform better in these two cases, MT-MD and MD-MA as in Fig. 5-19(a) and Fig. 5-19(c) respectively. For MT-MA, three objective functions generation shows a better result with lower total acceleration. Comparison of the simulation results illustrates the performance of the robot hand is significantly reduced for three objective functions optimization. The performance of the evolved neural controllers considering all three objective functions simultaneously will be discussed in the next section.

### 5.2.2 Three Objective Optimization

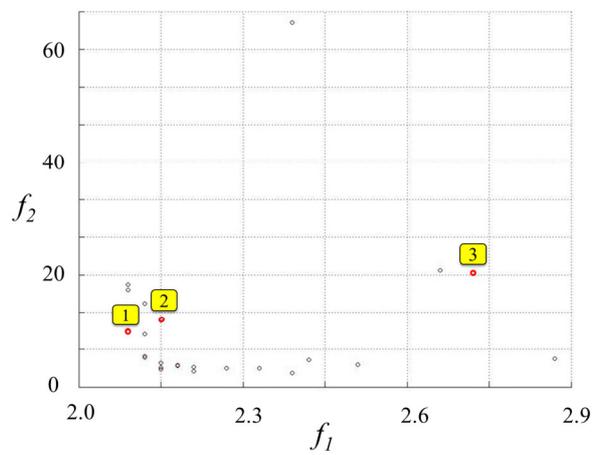
In this experiment, a similar task of placing a bottle from a holding position on the table is adapted. The performances of generated optimal neural controllers are tested in simulated environment and on the real robot. The Pareto front optimizing the three criteria as in ( $f_1$ ), ( $f_2$ ) and ( $f_3$ ), are shown in Fig. 5-20 (Mohamed, Kitani, et al. 2013d).



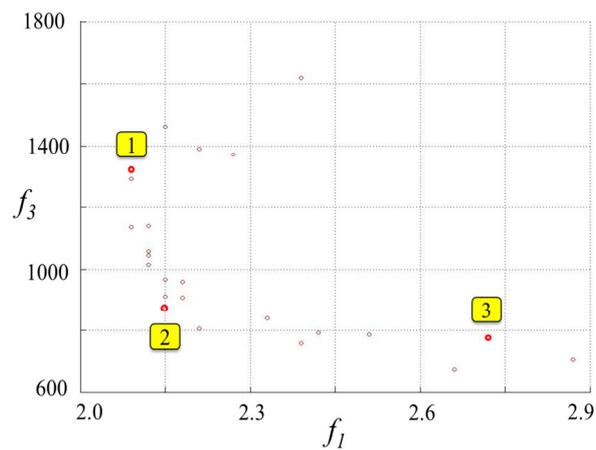
**Figure 5-20** Pareto front of MT-MD-MA objective functions optimization.



(a)



(b)



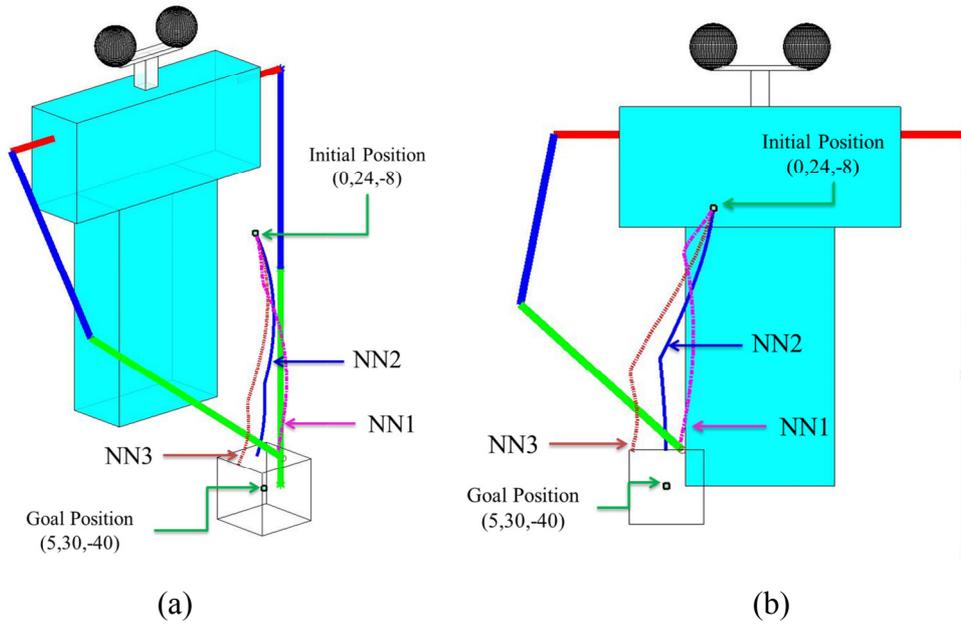
(c)

**Figure 5-21** Pareto front of MT-MD-MA objective functions optimization (a)  $f_2$ - $f_1$  view (b)  $f_3$ - $f_1$  view (c)  $f_3$ - $f_2$  view.

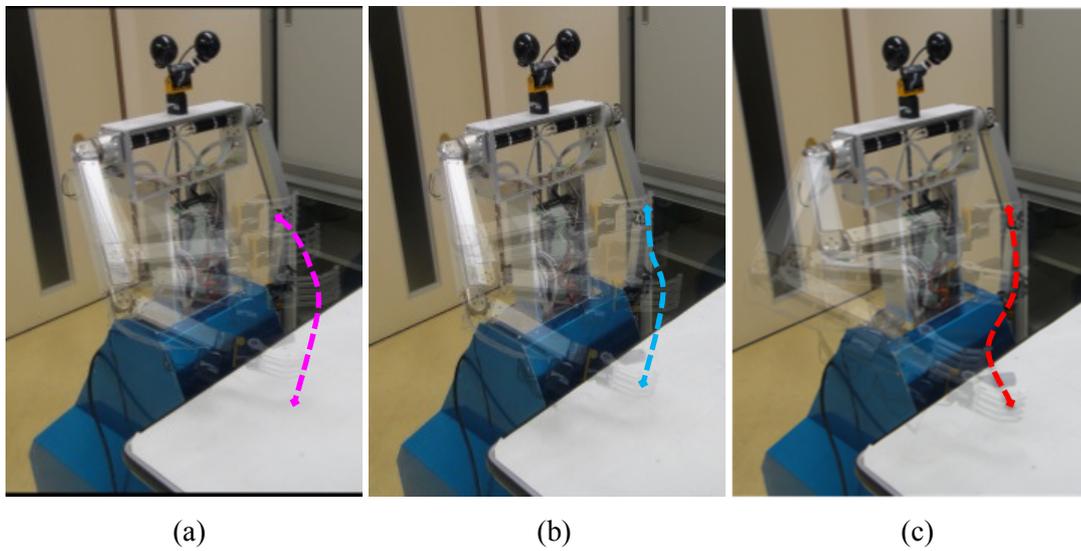
Twenty four individuals in Pareto set have been generated with maximum 80 iteration of the MOGA. Three individuals (NN1, NN2 and NN3) are chosen to be further discussed. The selection of these individuals is for comparing the performance of the optimal neural controller NN2 (one of the optimal solutions) with NN1 and NN3, which are prioritizing on two and single objective optimization respectively. NN3 is the extreme solution and very similar to single objective optimization which give priority to minimum acceleration objective function only (Fig. 5-20).

Fig. 5-21(a) and Fig. 5-21(b) clearly show the performance of NN1 solution optimizing two objective functions, minimum time ( $f_1$ ) and minimum distance ( $f_2$ ). However, moving with a minimum time will make the robot hand moves faster and have a higher total acceleration ( $f_3$ ) which is 30% more than NN2 and NN3 solutions. Fig. 5-21(a) and Fig. 5-21(c) show NN3 solution which is similar to a single objective optimization and only minimizing the total acceleration ( $f_3$ ) of the robot hand motion. The robot hand completes the task 20% slower than NN2 and NN3. If all three criteria is a priority in generating the robot hand motion, NN2 perform the best. With a small difference in distance ( $f_2$ ) and time ( $f_1$ ) compare to NN1, NN2 has lower total acceleration ( $f_3$ ) thus having a smoother motion and suitable for a task that required high stability and accuracy.

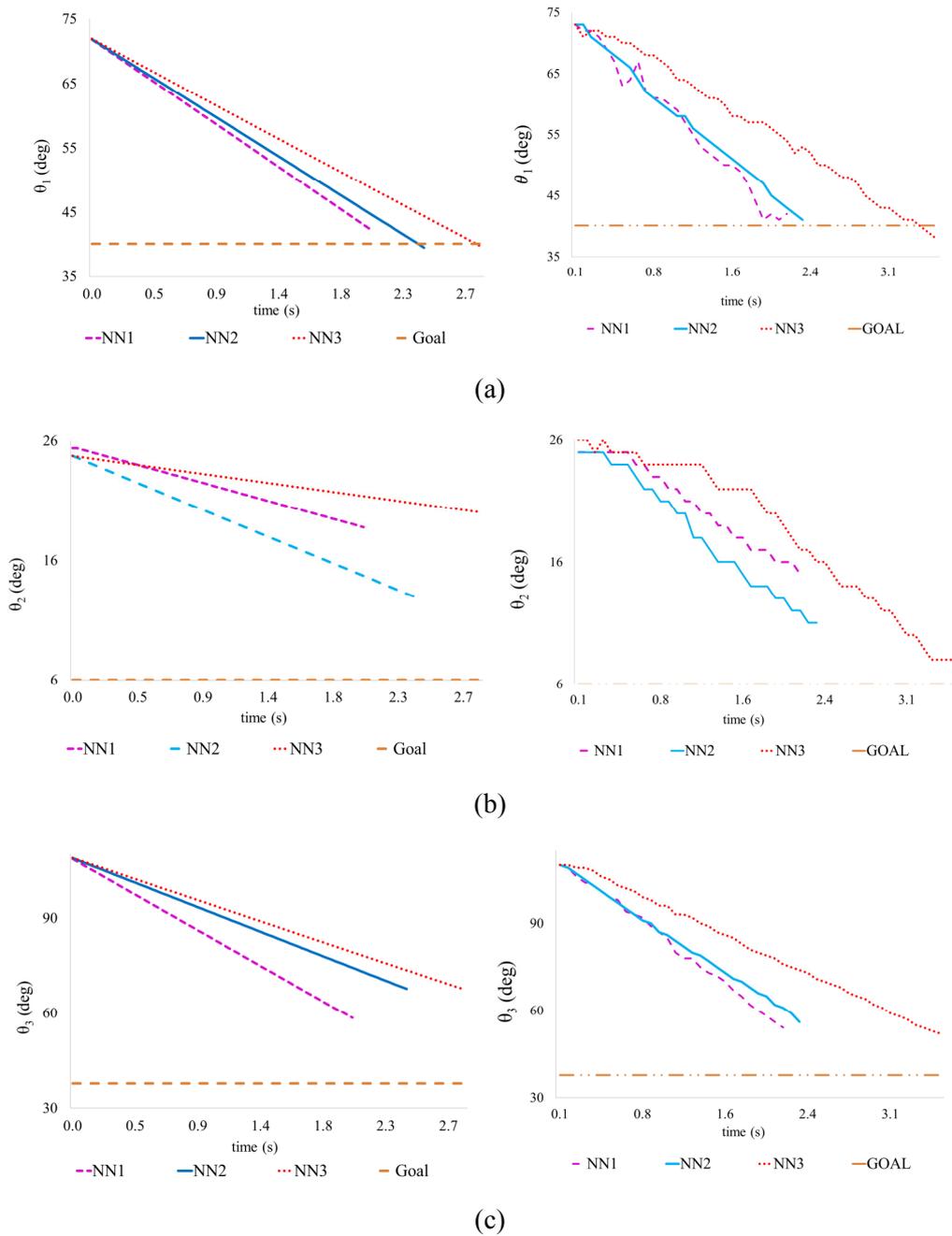
Fig. 5-22 shows the simulated motion generated using NN1, NN2 and NN3 solutions. The trajectory for NN1 and NN2 are improved compared to NN3. The robot hand move closer to the shortest distance for NN2 and NN1 solution compared to NN3 objective function where the robot hand move slightly away from the shortest trajectory to reach the goal.



**Figure 5-22** Robot hand motion adapting NN1, NN2 and NN3 solutions.



**Figure 5-23** Video capture of robot hand motion adapting NN1, NN2 and NN3 solutions.



**Figure 5-24** Joint angular displacement comparison between simulation and experiment of the robot arm (a)  $\theta_1$  (b)  $\theta_2$  (c)  $\theta_3$ .

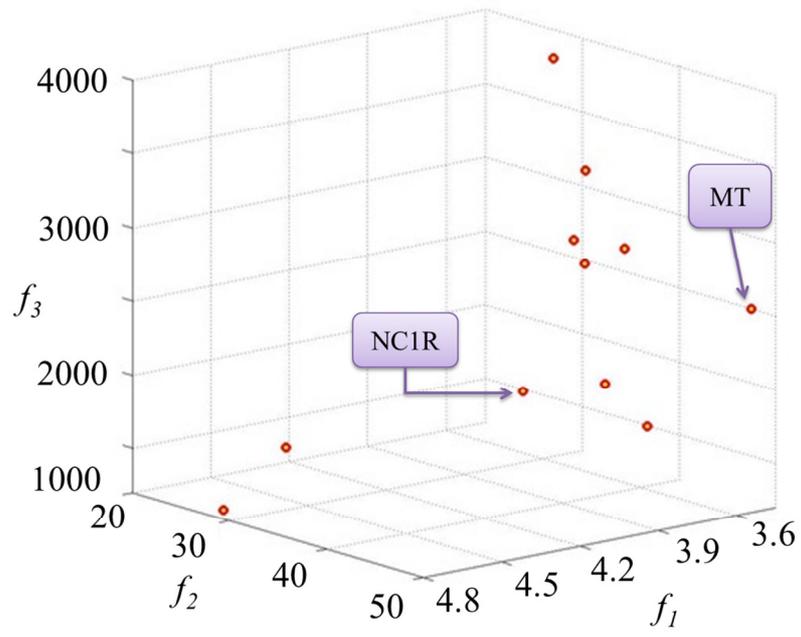
The performance of the generated neural controllers is further tested on the real robot. The video capture of the experiment with the humanoid mobile robot for NN1, NN2 and NN3 are shown in Fig. 5-23(a), Fig. 5-23(b) and Fig. 5-23(c) respectively.

The generated motion satisfying all three objective functions can be clearly differentiate and visualized. Fig. 5-24(a), Fig. 5-24(b) and Fig. 5-24(c) show simulation results of the joint angular displacement for shoulder ( $\theta_1$ ), upper arm ( $\theta_2$ ) and lower arm ( $\theta_3$ ) respectively. The performance is compared with the joint angular displacement acquired from the real robot. The neural controllers show good performance in both environments but show some deterioration in maintaining a straight line trajectory and it can be visualize when the robot is in motion.

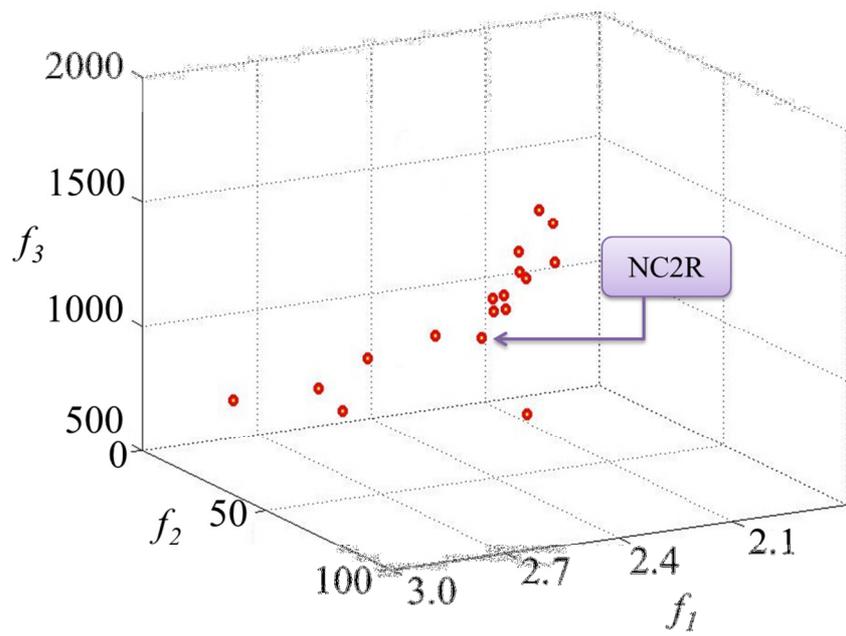
### 5.3 Neural Controllers Performance and Obstacle Avoidance

The Pareto front a single run MOGA of 80<sup>th</sup> generation optimizing all three criteria for the right (Fig. 5-25) and left hand (Fig. 5-26), picking and holding motions show a clear trade-off among objective functions. The Pareto front has 11 and 17 neural controllers for the right hand picking and holding motions, respectively. While the Pareto front of the left hand has 12 and 18 neural controllers. The MOGA did not converge in the same number of the Pareto front solutions. The motion to reach the target object is different from the motion to the holding position because the robot has to avoid the table (Mohamed, Mano, et al. 2013a).

Based on the generated Pareto front, we have selected two sets of the best optimized robot arm motions. One neural controller is for picking the spray can, NC1R (Fig. 5-25(a)) and another one for holding it up, NC2R (Fig. 5-25(b)) using the right hand. Another set of neural controllers is for the left hand motion (NC1L and NC2L) are shown in Fig. 5-26(a) and Fig. 5-26(b), respectively.

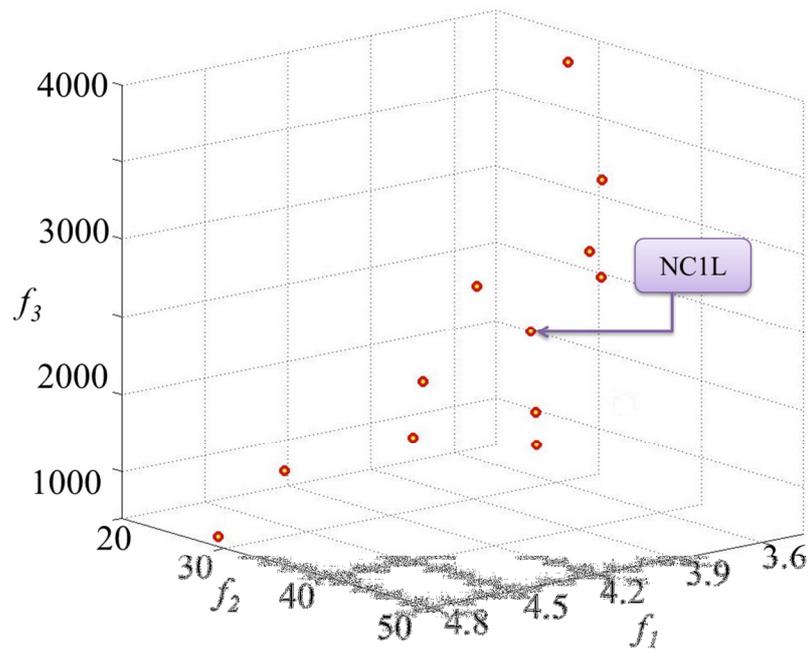


(a)

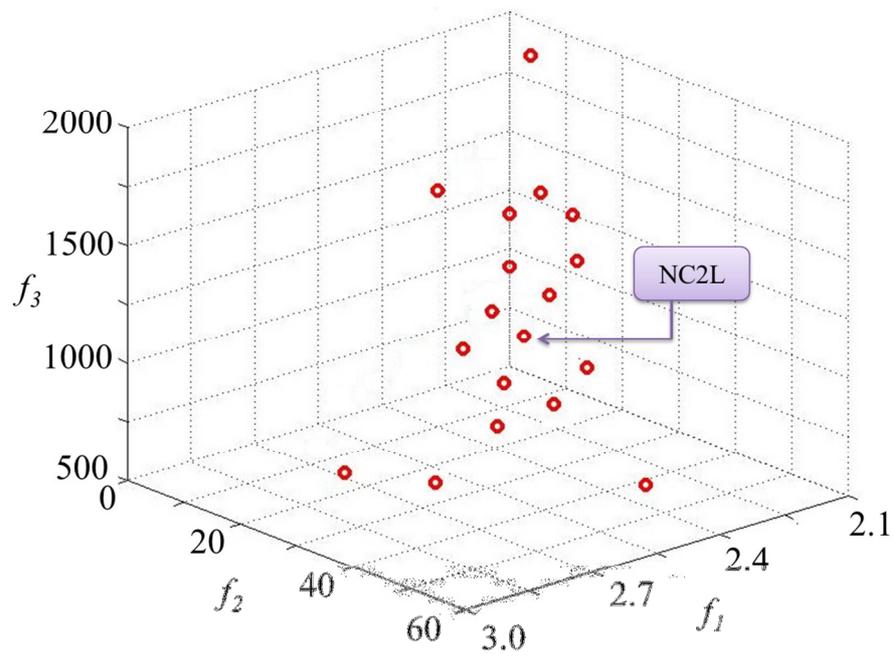


(b)

**Figure 5-25** Selected neural controllers for right hand (a) Picking motion (b) Holding motion.



(a)



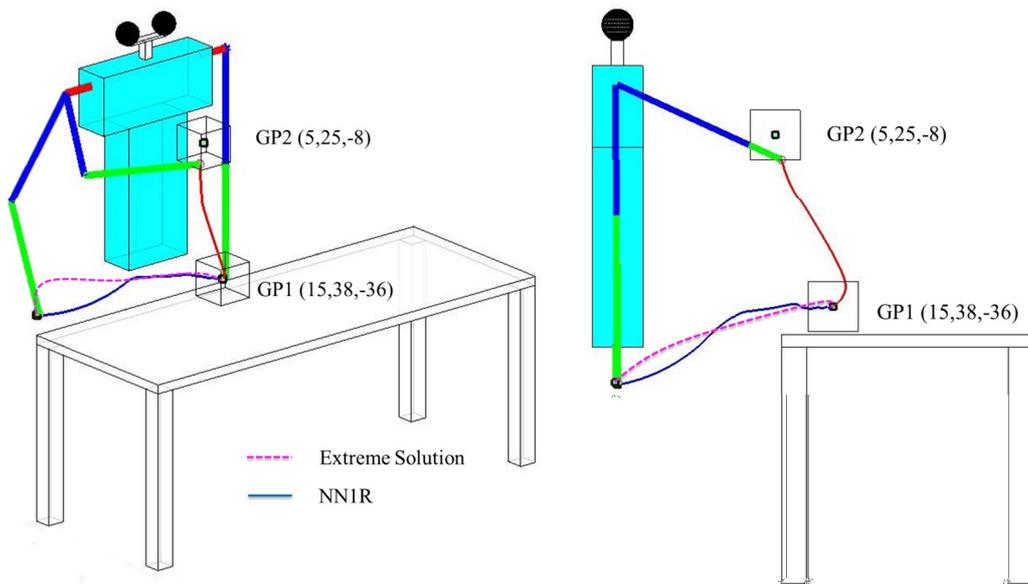
(b)

**Figure 5-26** Selected neural controllers for left hand (a) Picking motion (b) Holding motion.

Random goal positions 1 and 2 are chosen to test the performance of the evolved neural controllers. The robot has to move the hand from its initial position to grasp the spray can place on the table (goal position 1). Then move the spray can to goal position 2. The robot will use its right or left arm based on the spray can location relative to his body. For example, if the spray is on the right hand side, the robot will choose its right hand. Table 5-1 shows the summary of simulation environment set-ups.

**Table 5-1** Simulation parameters.

Environment Setup	Initial Position (Fixed)			Goal Position 1 (Random)			Goal Position 2 (Random)		
	X <sub>init</sub>	Y <sub>init</sub>	Z <sub>init</sub>	X <sub>g1</sub>	Y <sub>g1</sub>	Z <sub>g1</sub>	X <sub>g2</sub>	Y <sub>g2</sub>	Z <sub>g2</sub>
Environment 1	30	0	-48	15	38	-36	5	25	-8
Environment 2	-30	0	-48	-15	38	-36	-20	30	-15
Environment 3	30	0	-48	25	34	-35	5	22	-18



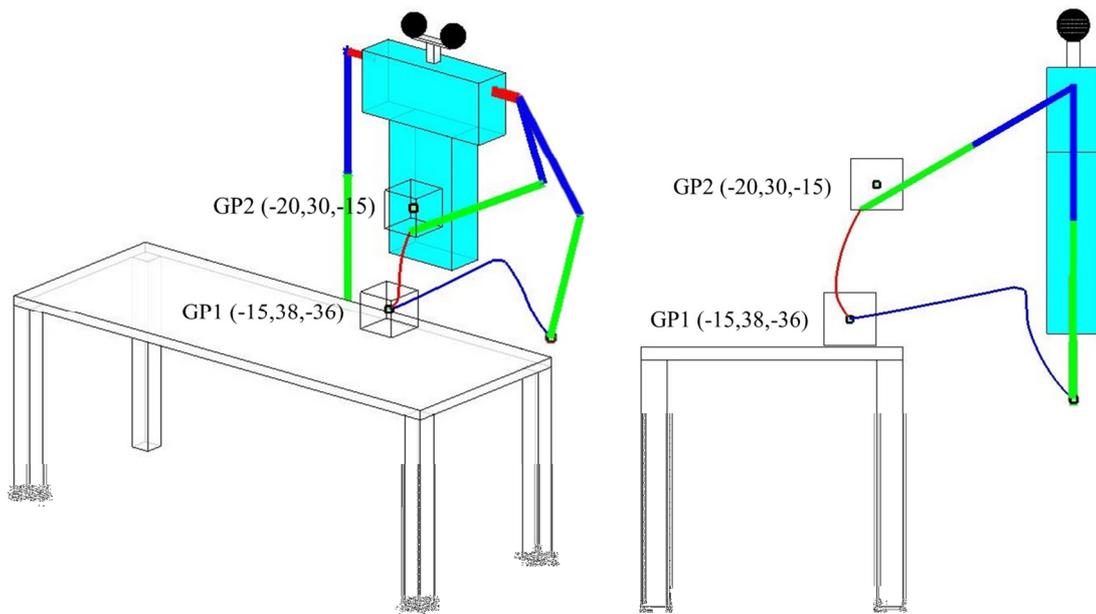
**Figure 5-27** Simulation results for random goal position 1 and 2 for environment 1.

Three different positions of the spray can on the table have been selected. In the first simulation, the spray can is randomly placed near to the center of the table, as shown in Fig. 5-27. The robot hand moves in a constant velocity toward the goal position 1. The velocity is reduced when it gets near to the goal position 1. Once the robot hand reaches the goal position 1, the current position of the robot hand is determined and it is set to be the initial position of the next motion to goal position 2.

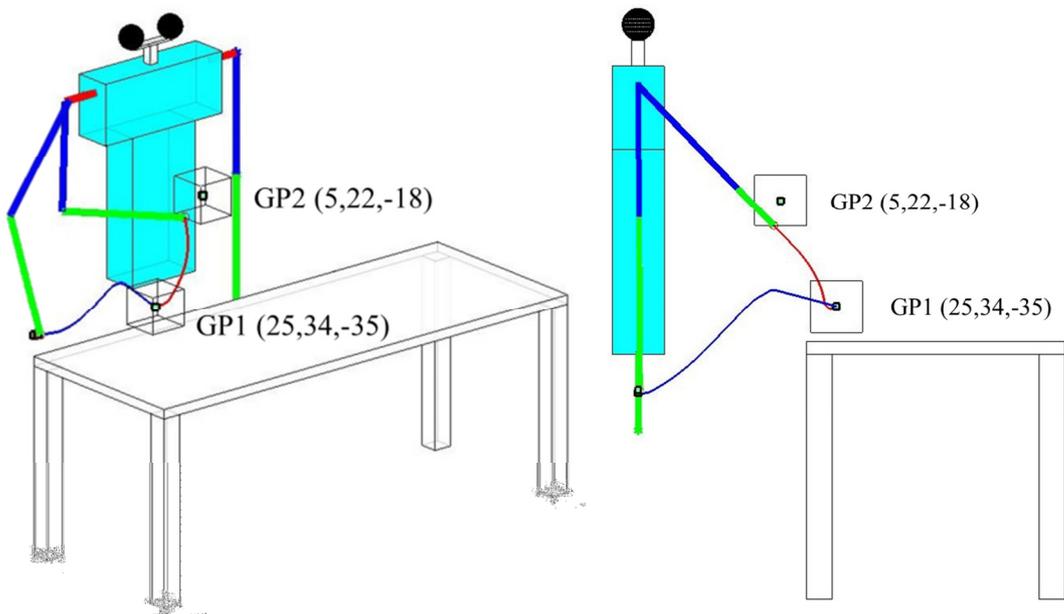
In order to compare the performance of NC1R, the neural controller that minimizes the time (MT) to goal position is selected (Fig. 5-25(a)). The motion trajectories are shown in Fig. 5-27. The result shows that the hand reached the goal location in 3.8 second, which 0.2 second longer than the minimum time (MT) trajectory. On the other hand, for such a small deterioration in the moving time, the acceleration and distance are reduced by 40% and 50% respectively.

In the next set-up, the position of spray can is on the left side of the robot, hence the left hand is chosen by the robot to perform the task (Fig. 5-28). Although the motion characteristic of the left hand compare to the right hand is not similar, the task is completed successfully. This is due to the differences in the motor data obtained from both arms.

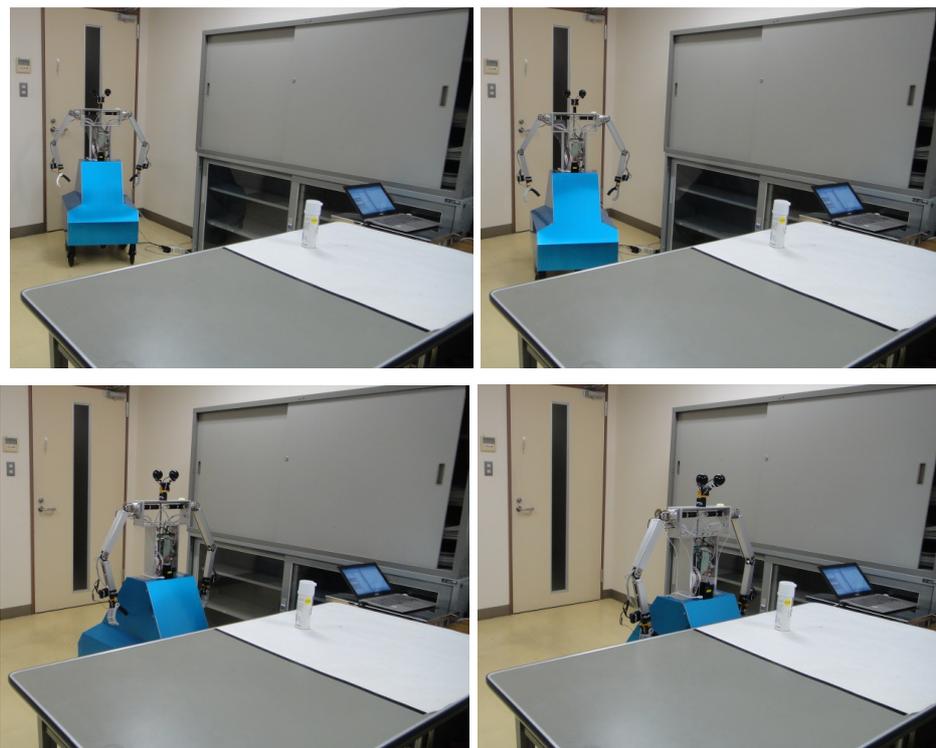
In the third simulation, the position of the spray can is slightly further to the right, as shown in Fig. 5-29. The kinematics constraints for the robot hand to reach the spray can in this motion is reduced compare to the first motion, thus, by utilizing the same neural controller, the robot hand manages to perform the task successfully.



**Figure 5-28** Simulation results for random goal position 1 and 2 for environment 2.



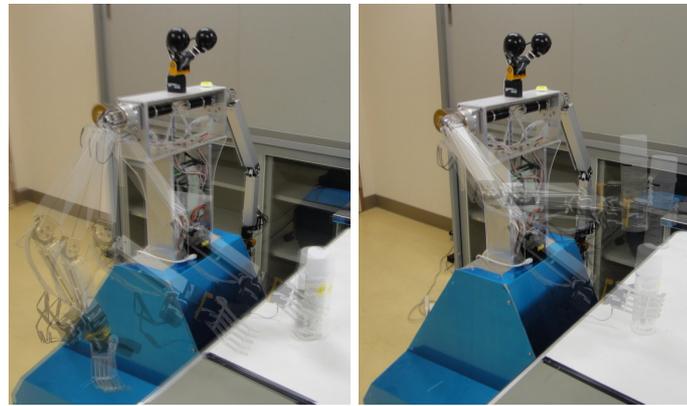
**Figure 5-29** Simulation results for random goal position 1 and 2 for environment 3.



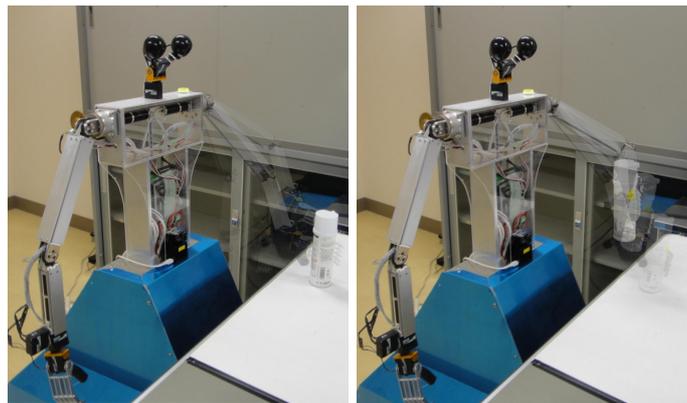
**Figure 5-30** Mobile humanoid robot moving toward the table.

The performance of optimized motion generated is tested on the mobile humanoid robot. In this experiment, the mobile humanoid robot is required to move from the entrance toward the table as in Fig. 5-30. The mobile humanoid robot utilized the webcam and laser range finder to navigate toward the table and stop in the desired distance relative to the object. In the first experiment, the spray can is placed on the same position used in the simulation and the right hand is chosen for this task (Fig. 5-31(a)). The robot performance is similar with the simulated one, where the speed is reduced before the robot reaches the goal position 1 and 2. In difference from simulation, in the real hardware it is hard for the robot to follow a straight line. The reason is that, kinematically the robot arms have some difficulties if the spray can is placed in the middle of the workspace and the communication between the robot and the PC while the robot

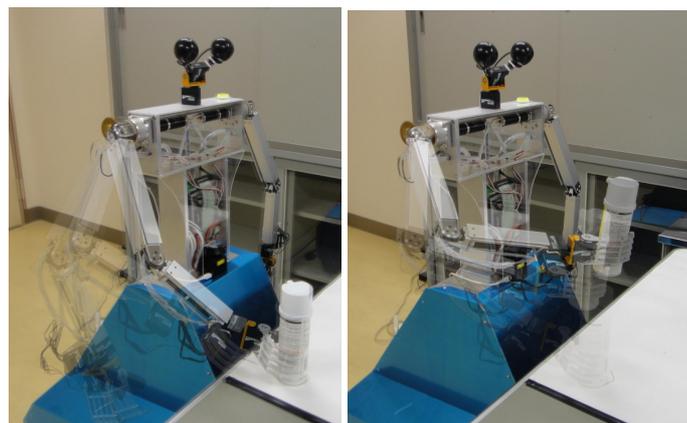
moves need to be further improved. The experiment is repeated 20 times to verify the performance of the proposed method. Results show that out of 20 times, the robot hand reached the goal position in 90% of the trials.



(a)



(b)



(c)

**Figure 5-31** Experimental results for random goal position 1 and 2 (a) Environment 1 (b) Environment 2 (c) Environment 3.

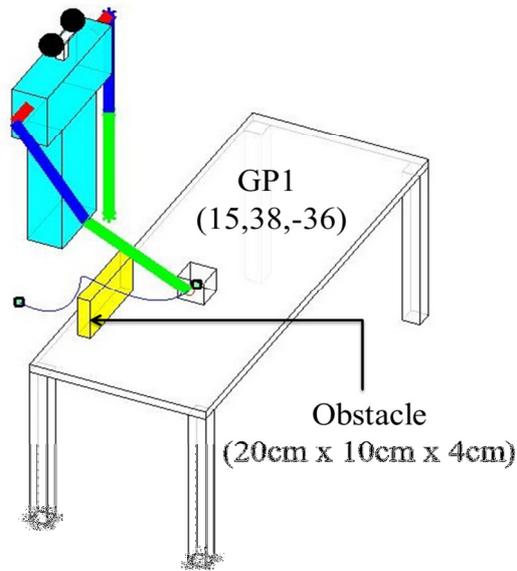
Fig. 5-31(b) shows the robot left hand motion for the second experiment. The spray can is placed on the left side of the table. Similar problem occur in this experiment, the robot hand has some difficulties maintaining the straight line motion. Out of 20 trials, 90% the robot left hand successfully reaches the goal position.

In the third experiment, the spray can is placed on the further right of the table as shown in Fig. 5-31(c). The performance of the robot hand is better in this experiment, where the robot motion is smoother following a straight line. The kinematics of the robot arm for this motion is less restricted compared to the other two cases. The successful rate of the robot hand reaching the goal position is 95%. The reason that the robot did not reach the goal in 5% of the trials is related with some error in the sensory data.

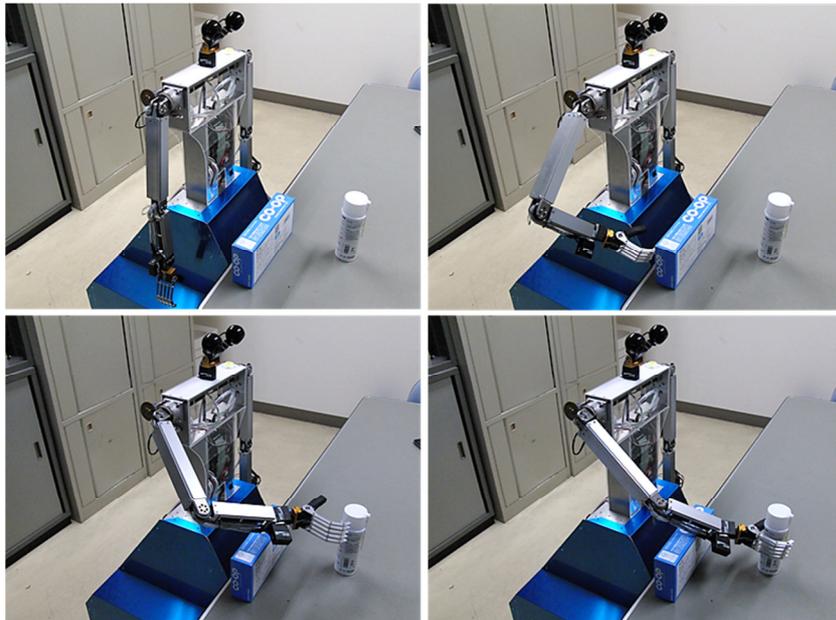
In order to verify the performance of evolved neural controllers, dynamic environments where the trajectory is partly blocked by obstacles of different sizes and positions is also considered. In these experiments the right arm motion is generated. Based on the obstacle size and position, which can be determined by camera and laser sensor, the specific pre-evolved neural controller is selected. In this experiment, the length, height and width of the obstacle are 20 cm, 10 cm, and 4 cm, respectively. The height and width of the obstacle is within the partition 1.

The motion generated by the minimum distance neural controller is shown in Fig. 5-32. Both simulation and experimental results show good performance. The robot avoids the obstacle and reaches the goal successfully. The successful rate of the robot hand reaching the goal position is 90%. Fig. 5-33(a), Fig. 5-33(b) and Fig. 5-33(c) show the comparison between simulated motions and the real robot hand trajectory for  $x$ ,  $y$  and  $z$  axis respectively. These results show that the simulation and real

robot angle trajectories are very similar. Fig. 5-33(a) and Fig. 5-33(c) show some differences between simulation and the real robot implementation in the  $x$  and  $z$  directions. Some irregularities are also shown when the robot arm nearly reaching the goal position. This is due to some difference in the sampling rate between the simulator and real robot experiments.

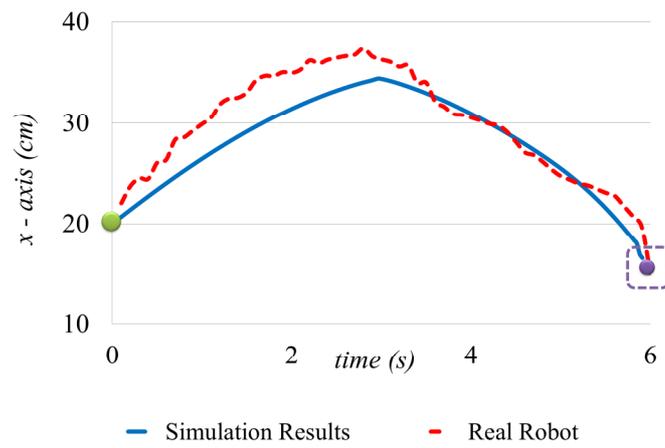


(a)

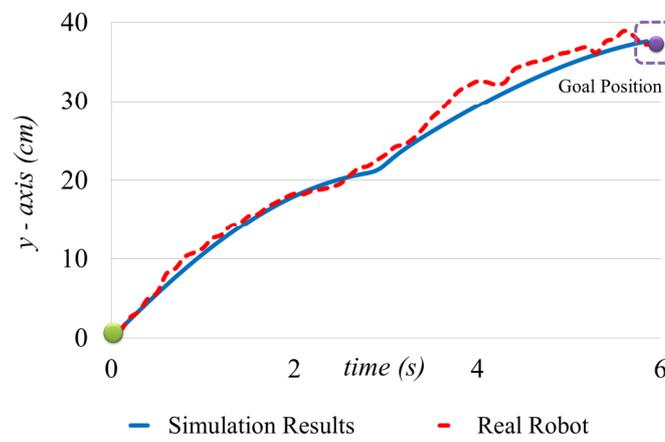


(b)

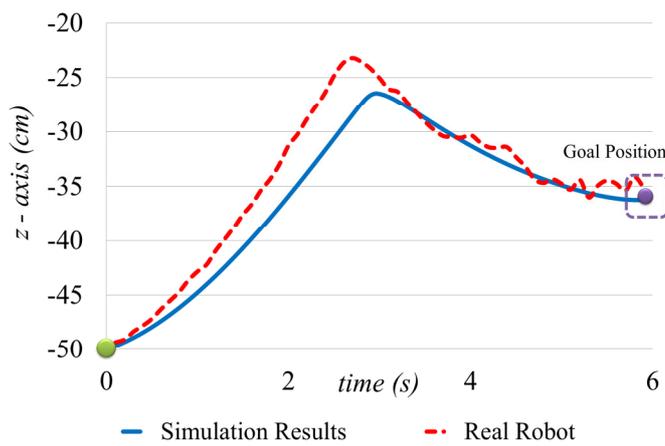
**Figure 5-32** Obstacle avoidance results for (a) Simulation (b) Real robot.



(a)



(b)

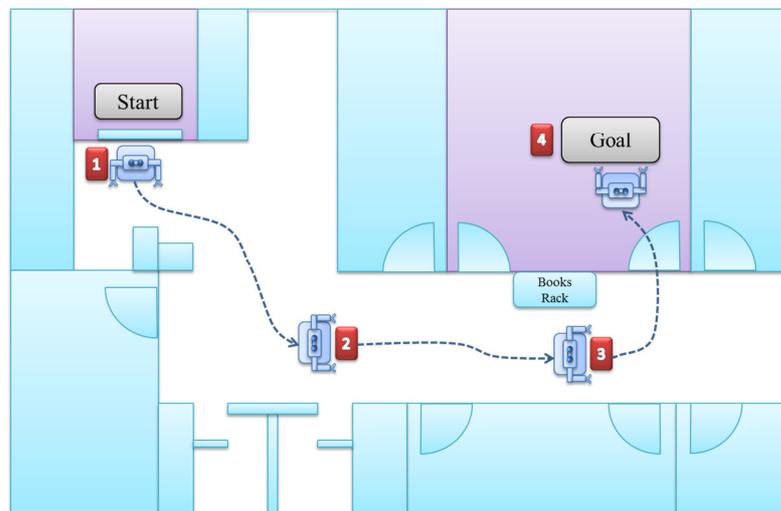


(c)

**Figure 5-33** Trajectory comparison between simulation results and the real robot for (a) x-axis (b) y-axis and (c) z-axis.

## 5.4 Robot Navigation

This experiment is divided into two sections, first is the robot navigation from the lift (point1) to the goal position at point 4 and second the robot arm motion generation from the robot hand initial position to the spray can position on the table. The robot navigation task from the starting point to the goal is divided into three environments as in Fig. 5-34. Environment 1 is from point 1 to point 2, environment 2 is point 2 to point 3 and the third environment is from point 3 to point 4.



(a)



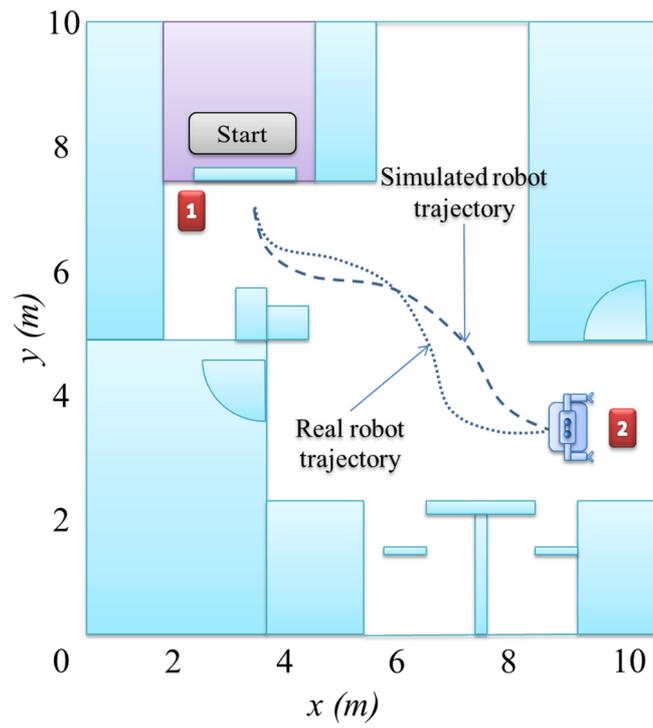
(b)

**Figure 5-34** Robot navigation (a) Simulation (b) Real environments.

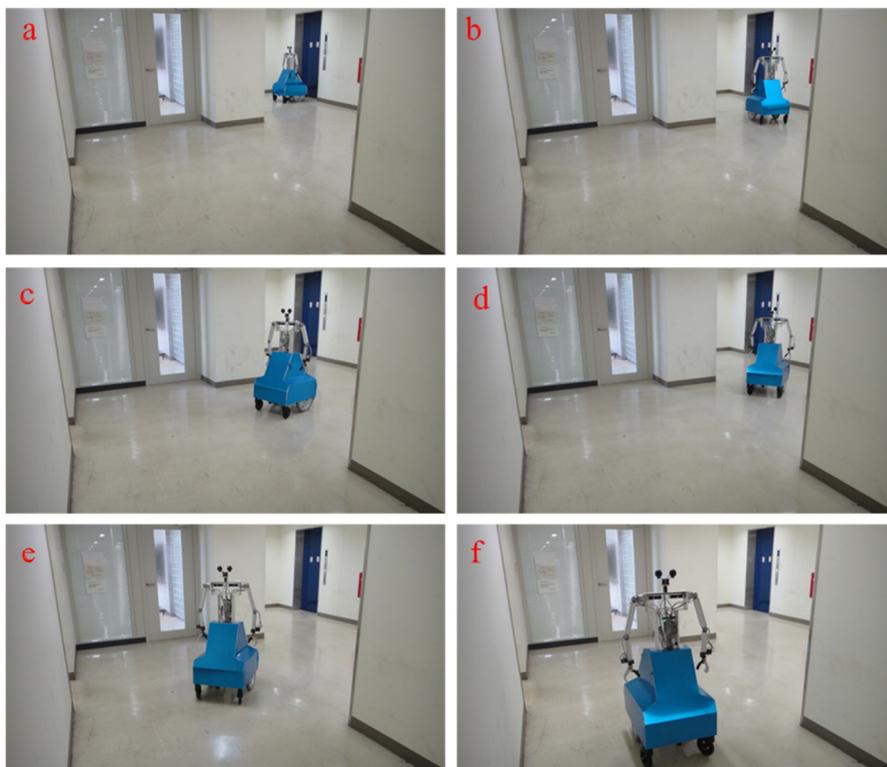
The performance of neural controllers generated in the simulation is tested on the real robot. Although it is in three sections, the robot moves continuously without stopping between each environment. The comparison show the robot has some difficulties in environment 1 as in Fig. 5-35(a). Simulated result shows shorter and smoother trajectory from the starting point 1 to point 2 comparing to the real robot motion. The total distance of the real robot is slightly higher but manages to reach the target location successfully. Out of 20 trials that have been carried out, 95% the robot manage to reach at the centre of the hallway. Fig. 5-35(b) shows the video capture of the real robot motion in environment 1.

As the robot reach environment 2, neural controller 2 will be employed and result show good performance in both simulation and on the real robot as in Fig. 5-36(a). The robot has to move is a straight hallway avoiding a book rack before it reach the third environment. The real robot has the ability to move in a similar distance and trajectory as in simulated environment. Fig. 5-36(b) shows the robot motion along the hallway while avoiding the book rack. Out of 20 trials that have been carried out, 95% the robot manage to reach point 3 successfully.

In the third environment the robot has some difficulties to execute the task. The robot needs to enter the room through a 95cm door and stop near the table for manipulation task. Fig. 5-37(a) shows the best result and the real robot move in a slightly in the inner section compares to the simulation result. In this environment, the successful rate of the robot reaching the goal position is reduced. Fig. 5-37(b) shows the video capture of the real robot entering the room. The successful rate of the robot entering the room is 80% out of 20 trials.

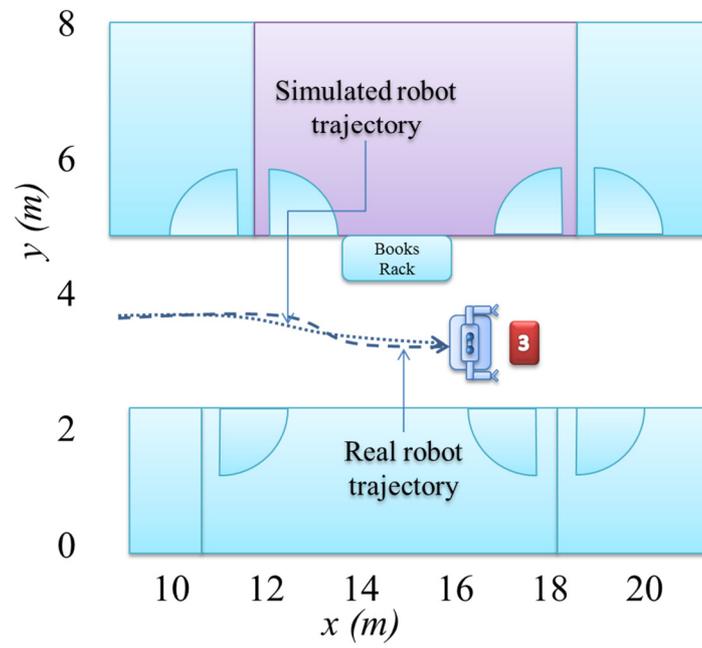


(a)



(b)

**Figure 5-35** Robot navigation in environment 1 (a) Simulation (b) Experiment.

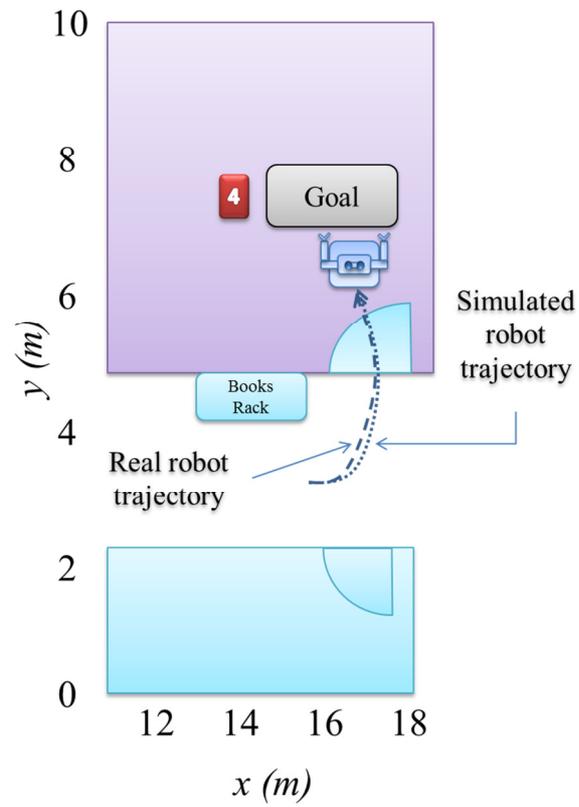


(a)



(b)

**Figure 5-36** Robot navigation in environment 2 (a) Simulation (b) Experiment.



(a)



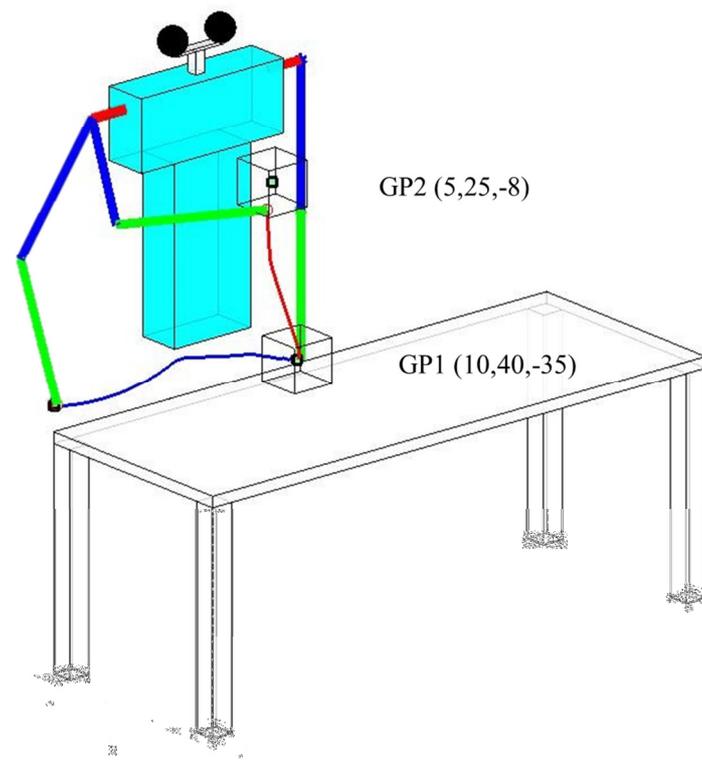
(b)

**Figure 5-37** Robot navigation in environment 3 (a) Simulation (b) Experiment.

In the next task, the robot hand is required to perform a similar task as in the previous experiment to pick a spray can on the table. The neural controllers generated in Section 5.3 are utilized in this experiment. LRF2 is used to determine the position of the spray can on the table and once the spray can position is determined, the robot will decide the best neural controller to for executing the task. The determination of which arm to manipulate the task is depending on the position of the spray can on the table as in the previous experiment.

Fig. 5-38 shows the simulation results of the optimized arm motion. The robot arm moves with the shortest distance and minimum acceleration (MD-MA) while avoiding the table. The simulation goal position 1 (GP1) is determine based on the actual position of the spray can as in the experiment. A different neural controller is utilized to pick up the spray can from the table to goal position 2 (GP2). The two neural controllers adapted in this task have the best performance in simulation environment. The same neural controllers manage to execute the motion in a wide range of initial and goal position and it is proven in Section 5.3

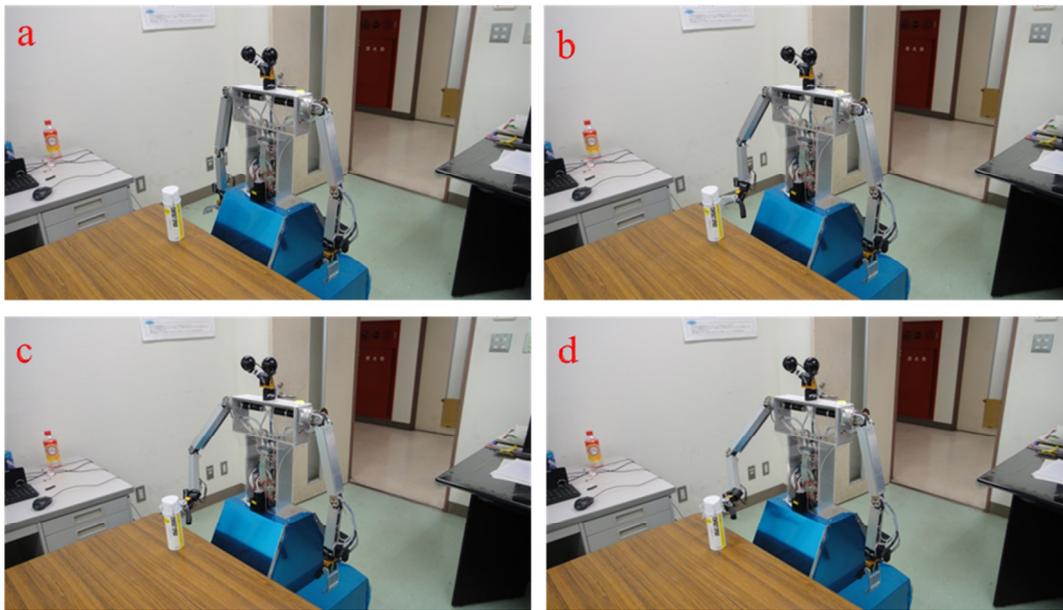
The video capture of the experiment with the humanoid mobile robot is shown in Fig. 5-39, Fig. 5-40 and Fig. 5-41. Fig. 5-39 shows the robot motion for environment 3 and simultaneously the robot utilizes the laser and camera sensors data to reach the table. The robot arm motion is generated by an optimal neural controller optimizing distance and acceleration in order to compare the performance. The position of the spray can is determined via LRF2 as in Fig. 5-40 and the spray can is detected on the right side of the robot, thus executing the task using right hand.



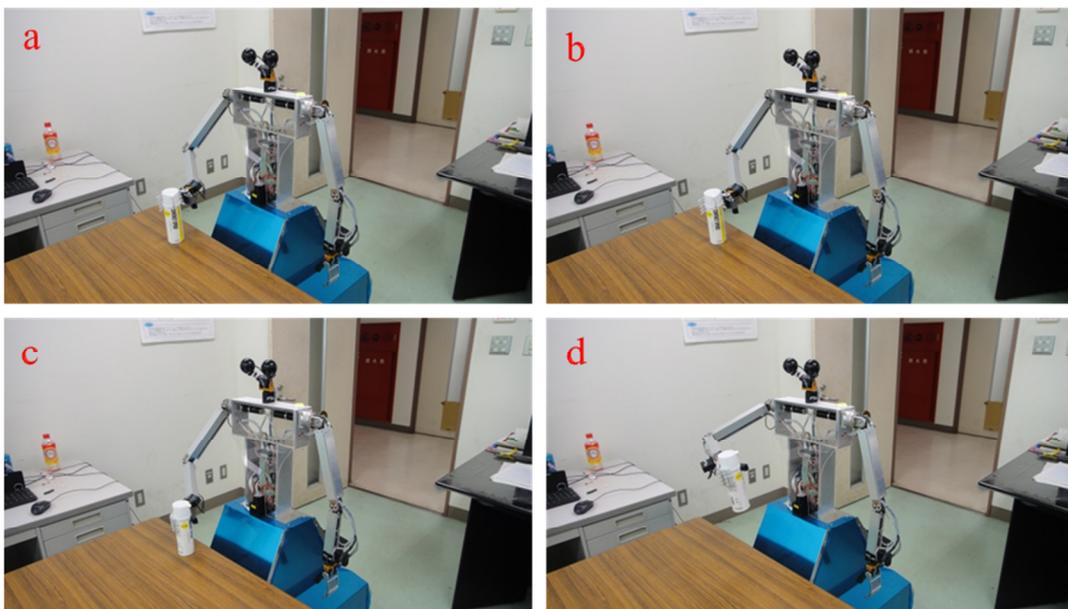
**Figure 5-38** Robot hand simulation.



**Figure 5-39** Mobile robot approaching the table.



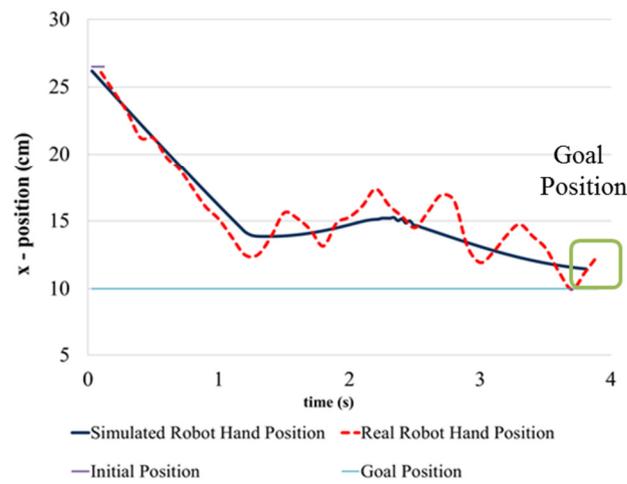
**Figure 5-40** Spray can detection via laser range sensor and reaching motion of the robot hand.



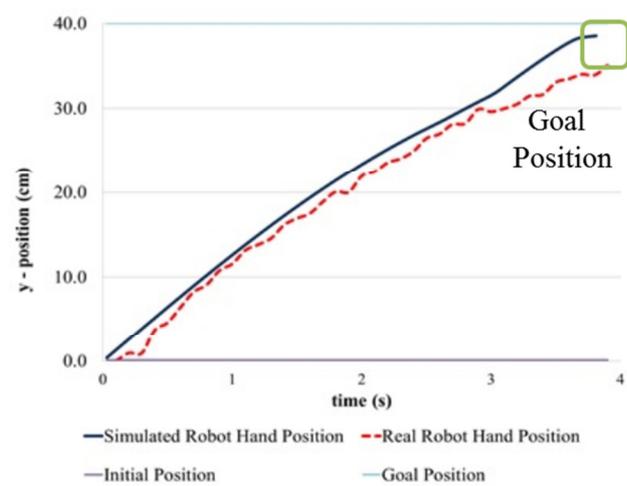
**Figure 5-41** Reaching and picking the spray can motion of the robot hand.

Fig. 5-42(a), Fig. 5-42(b) and 5-42(c) show the comparison results between simulation and experiment for  $x$ ,  $y$  and  $z$  axis respectively. These comparisons verified the performance of the neural controllers applied to the real robot and show good performance. Some deterioration in motion can be seen in  $x$  and  $z$  directions (Fig. 5-42(a) and 5-42(c)) as the robot hand try to maintain the straight line trajectory toward the goal position when avoiding the table and the performance show similar result with the previous experiment.

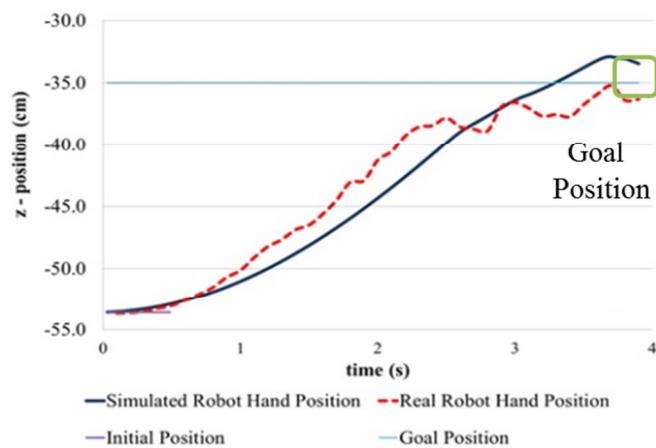
A similar comparison is done for the second arm motion from GP1 to GP2 as in Fig. 5-43(a), Fig. 5-43(b) and Fig. 5-43(c). The robot successfully adapted the optimal neural controllers to complete the second task of picking the spray can. In this second task, GP1 becomes the initial position for the robot hand to move to GP2. This shows that the generated neural controllers can adapt changes in initial and goal position without compromising the performance. Out of 20 trials, 90% the robot hand manage to reach the spray can successfully. Eventhough both navigation and arm motion experiment are not perform continuously, in the future, the mobile humanoid robot will be further improved by increasing its mobility and fully integrating the mobile platform and upper body.



(a)

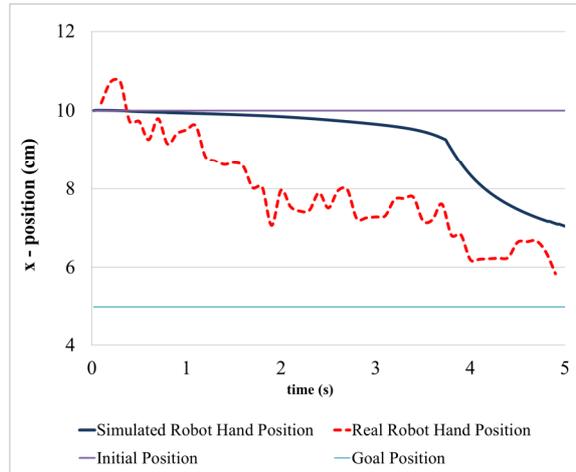


(b)

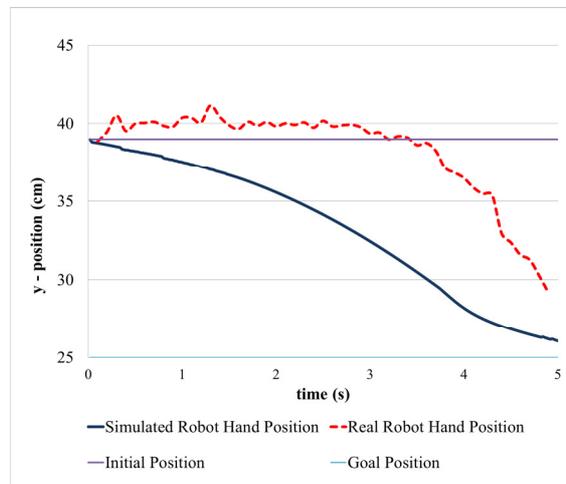


(c)

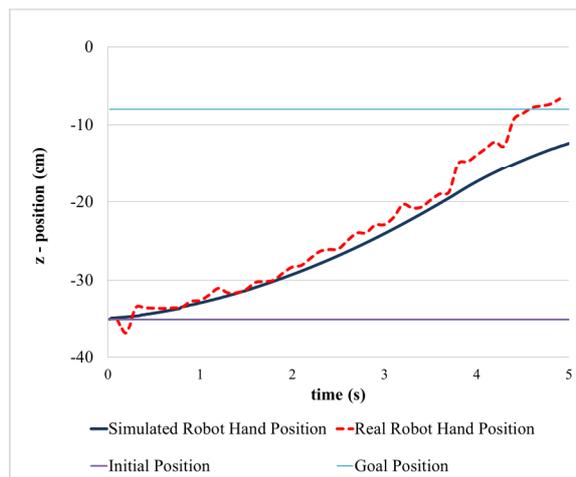
**Figure 5-42** Comparison of the robot hand motion between simulation and experiment (GP1) for (a)  $x$ -axis (b)  $y$ -axis (c)  $z$ -axis



(a)



(b)



(c)

**Figure 5-43** Comparison of the robot hand motion between simulation and experiment (GP2) for (a)  $x$ -axis (b)  $y$ -axis (c)  $z$ -axis

---

# CHAPTER 6

---

## 6 CONCLUSION AND DISCUSSION

In this thesis we have proposed a new method for humanoid robot arm generation satisfying several objective functions. Four different objective functions are chosen for the robot arm motion generations which are minimum execution time, minimum distance, minimum acceleration and minimum angular acceleration. The advantage of the proposed algorithm is that the robot can generate the hand motion by the best neural controller based on the task it has to complete. In addition, the same neural controller can be employed to generate robot hand motion for different initial and goal positions. The results obtained indicate that by increasing the number of objective functions, the overall performance of the MOGA slightly decreases but the evolved neural controllers generated a much better motion trajectory, satisfying multiple criteria, simultaneously. In order to compare the performance, SOGA results are also presented.

We also proposed an evolution based approach for humanoid robot arm motion generation in dynamic environments where obstacle avoidance motion generation of the robot arm was considered. Our mobile humanoid robot shows good performance both in simulation and experiments. The robot was able to reach the table, select the neural controller for the robot arm motion generation to pick up the object both, with or without obstacle in the pathway. The robot motion is generated based on three different objective functions which are simultaneously optimized. Therefore, the

humanoid robot can perform a wide range of tasks in real life environments, by selecting the appropriate motion.

A simple navigation of the robot maneuver in our lab environment is also presented in this thesis. SOGA is utilized in generating the robot motion in three environments. The robot is required to navigate from the lift to the goal position inside one of our lab room. The robot navigation systems show good performance in this task with high percentage of successful rate. A similar object detection experiment on the table is also done once the robot reaches the table.

The results are tested in the real hardware of mobile humanoid robot. The robot can navigate in the environment avoid obstacles and reaching the table where the object is placed. Based on the task the specific neural controller is selected to generate the optimal motion. Results show that although there are some small differences between simulations and experiments the robot completed the task.

## **6.1 Future Work**

In order to be fully autonomous assistive humanoid robot the following will be considered:

- a) Utilizing the sensors to determine the specific neural controller

In our system, we predetermine the robot task. However, in real applications the robot must utilize the sensors, such as camera, laser in order to determine the task and select the appropriate neural controller. The vision system needs to be fully utilized in assisting the robot arms to recognize and differentiate the obstacles and objects to be manipulated. The vision system is not only for the robot arm manipulations but also assisting the mobile platform when navigating throughout the environment in order for

the robot to be more robust in dynamic environments. The object and obstacle detection via LRF can be more flexible when scanning the environment both vertically and horizontally. These features are very important in determining object or obstacle size, height and distance from the robot.

b) Improving the robot hardware

A lot of improvements can be done to the existing mobile humanoid robot system in order to perform more complicated and difficult tasks. The overall design can be further improved especially the robot arms, to improve the accuracy and stability of the motion when adapting the generated arm motion. The size of the mobile platform can be reduced for better mobility while navigating and reaching the target location. The instability of the robot arm motion can be reduced by having better grippers, which can hold firmly the manipulated object without slippage. The safety features of the robot can also be increase by utilizing sensors on the main body.

c) Human robot communication

Human robot communication such as natural language understanding must also be considered. The robot has to understand commands from humans and execute them.

## REFERENCES

- Albers, A. et al., 2006. Body of a new Humanoid Robot - the Design of ARMAR III. In *IEEE-RAS International Conference on Humanoid Robot*. pp. 308–313.
- Ang, M., Pham, D.T. & Ng, K., 2009. Minimum-Time Motion Planning for a Robot using Bee Algorithm. In *7th IEEE International Conference on Industrial Informatics*. pp. 487–492.
- Asada, H. & Slotline, J.-J., 1986. *Robot Analysis and Control*, USA: John Wiley & Son.
- Asfour, T. et al., 2006. ARMAR-III : An Integrated Humanoid Platform for Sensory-Motor Control. *Humanoids 2006 - 6th IEEE-RAS International Conference on Humanoid Robots*, pp.169–175.
- Asfour, T., Berns, K. & Dillman, R., 2000. The Humanoid Robot ARMAR : Design and Control. In *1st IEEE-RAS International Conference on Humanoid Robot*. pp. 7–8.
- Belding, T., 1995. The Distributed Genetic Algorithm Revisited. In *Proceedings of the 6th International Conference on Genetic Algorithms*. pp. 114–121.
- Bergener, T. et al., 1999. Complex behavior by means of dynamical systems for an anthropomorphic robot. *Neural networks : the official journal of the International Neural Network Society*, 12(7-8), pp.1087–1099.

- Cantu-paz, E., 2000. Topologies , Migration Rates , and Multi-Population Parallel Genetic Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*. pp. 91–98.
- Capi, G., 2007. Multiobjective Evolution of Neural Controllers and Task Complexity. *IEEE Transactions on Robotics*, 23(6), pp.1225–1234.
- Capi, G. & Doya, K., 2005. Evolution of recurrent neural controllers using an extended parallel genetic algorithm. *Robotics and Autonomous Systems*, 52(2-3), pp.148–159.
- Chen, M. & Zalzal, A.M.S., 1997. A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration. *Journal of Robotic Systems*, 14(7), pp.529–544.
- Chen, T.L. et al., 2013. Robots for Humanity : A Case Study in Assistive Mobile Manipulation. *IEEE Robotics & Automation Magazine, Special issue on Assistive Robotics*.
- Chen, T.L., Jain, a & Kemp, C.C., 2010. Towards an assistive robot that autonomously performs bed baths for patient hygiene. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (1), pp.319–324.
- Chen, T.L. & Kemp, C.C., 2011. A Direct Physical Interface for Navigation and Positioning of a Robotic Nursing Assistant. *Advanced Robotics*, 25(5), pp.605–627.

- Chen, T.L. & Kemp, C.C., 2010. Lead me by the hand: Evaluation of a direct physical interface for nursing assistant robots. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Ieee, pp. 367–374.
- Coelho, J., Piater, J. & Grupen, R., 2000. Developing Haptic and Visual Perceptual Categories for Reaching and Grasping with a Humanoid Robot. , (September), pp.7–8.
- Dario, P. et al., 1996. Robot assistants : Applications and Evolution. , 18, pp.225–234.
- Dias, A.H.F. & de Vasconcelos, J. a., 2002. Multiobjective genetic algorithms applied to solve optimization problems. *IEEE Transactions on Magnetics*, 38(2), pp.1133–1136.
- Flash, T. & Hogans, N., 1985. The Coordination of Arm Movements: Mathematical Model'. *The Journal of Neuroscience*, 5(7), pp.1688–1703.
- Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company Inc.
- Hamill, J. & Knutzen, K., 2003. *Biomechanical Basis of Human Movement* Second. P. Darcy, ed., Lippincott William & Wilkins.
- Hamilton, N., Weimar, W. & Luttgens, K., 2008. *Kinesiology Scientific Basis of Human Motion* Eleventh., McGrawHill.
- Holland, J., 1975. *Adaptation in Natural and Artificial System*, The University Michigan Pres.

- Holz, D., Nieuwenhuisen, M. & Droeschel, D., 2013. Active Recognition and Manipulation for Mobile Robot Bin Picking.
- Iwata, H. & Sugano, S., 2009. Design of human symbiotic robot TWENDY-ONE. *2009 IEEE International Conference on Robotics and Automation*, pp.580–586.
- Jazar, R.N., 2007. *Theory of Applied Robotics : Kinematics, Dynamics and Control*, New York: Springer.
- Kawato, M. et al., 1990. Trajectory Formation of Arm Movement by Cascade Neural Network Model Based on Minimum Torque Change. *Biological Cybernetics*, 62, pp.275–288.
- Konno, A. et al., 1997. Development of A Humanoid Robot Saika. In *IEEE International Conference on Intelligent Robotic System*. pp. 805–810.
- Kose, S., 1997. Housing Elderly People in Japan. *Ageing International*, 1966(Table 1), pp.148–164.
- Kubota, N. et al., 1997. Trajectory Generation for Redundant Manipulator Using Virus Evolutionary Genetic Algorithm. , (April), pp.205–210.
- Lee, M.K. et al., 2009. The Snackbot : Documenting the Design of a Robot for Long-term Human-Robot Interaction.
- Lee, M.K. & Forlizzi, J., 2009. Designing Adaptive Robotic Services. In *International Association of Societies Design Research 2009*. pp. 1–10.

- Liu, Y., Jiang, Y. & Li, L., 2011. Multi-objective performance optimization of redundant robots using differential evolution. *Proceedings of 2011 6th International Forum on Strategic Technology*, (2), pp.410–414.
- Lopes, M., Beira, R. & Prac, M., 2004. An anthropomorphic robot torso for imitation : design and experiments. , pp.661–667.
- Mitchell, M., 1999. *An Introduction to Genetic Algorithms*, The MIT Press.
- Mohamed, Z., Mano, M., et al., 2013a. Adaptive Humanoid Robot Arm Motion Generation by Evolved Neural Controllers. *International Journal of Control Automation and System*, pp.1–6.
- Mohamed, Z., Mano, M., et al., 2013b. Adaptive Humanoid Robot Arm Motion Generation by Evolved Neural Controllers. In *International Conference on Advances in Computer and Electronics Technology*. pp. 978–981.
- Mohamed, Z. & Capi, G., 2012. Development of a New Mobile Humanoid Robot for Assisting Elderly People. *Procedia Engineering*, 41(Iris), pp.345–351.
- Mohamed, Z., Kitani, M. & Capi, G., 2013a. Adaptive Arm Motion Generation of Humanoid Robot Operating in Dynamic Environments. *Industrial Robot: An International Journal*.
- Mohamed, Z., Kitani, M. & Capi, G., 2013b. Humanoid Robot Arm Performance Optimization using Multi Objective Evolutionary Algorithm. *International Journal of Control Automation and System*, pp.1–8.

- Mohamed, Z., Kitani, M. & Capi, G., 2013c. Optimization of Robot Arm Motion in Human Environment. *International Journal of Enhanced Research Publication*, 2(10), pp.1–7.
- Montana, D.J. & Davis, L., 1989. Training Feedforward Neural Networks Using Genetic Algorithms. In *Eleventh International Joint Conferences on Artificial Intelligence*. pp. 762–767.
- Mukai, T. et al., 2010. Development of a nursing-care assistant robot RIBA that can lift a human in its arms. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.5996–6001.
- Nakano, E. et al., 1999. Quantitative Examinations of Internal Representations for Arm Trajectory Planning : Minimum Commanded Torque Change Model  
Quantitative Examinations of Internal Representations for Arm Trajectory Planning : Minimum Commanded Torque Change Model. *Journal of Neurophysiol*, 81, pp.2140–2155.
- Natale, L. et al., 2007. Learning precise 3D reaching in a humanoid robot. *2007 IEEE 6th International Conference on Development and Learning*, pp.324–329.  
Available at:  
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4354059>.
- Nieuwenhuisen, M. et al., 2013. Mobile bin picking with an anthropomorphic service robot. *2013 IEEE International Conference on Robotics and Automation*, (May), pp.2327–2334.

- Odashima, T. et al., 2006. A Soft Human-Interactive Robot RI-MAN. In *International Conference on Intelligent Robot and Systems*.
- Pires, E.J.S. & Machado, J.A.T., 2000. Trajectory Optimization for Redundant Robots Using Genetic Algorithms. In *Proceeding of the Genetic and Evolutionary Computation Conference*. p. 967.
- Pires, E.J.S., de Moura Oliveira, P.B. & Machado, J. a. T., 2007. Manipulator trajectory planning using a MOEA. *Applied Soft Computing*, 7(3), pp.659–667.
- Pollack, M.E. et al., 2002. Pearl : A Mobile Robotic Assistant for the Elderly.
- Ramabalan, S., Saravanan, R. & Balamurugan, C., 2008. Multi-objective dynamic optimal trajectory planning of robot manipulators in the presence of obstacles. *The International Journal of Advanced Manufacturing Technology*, 41(5-6), pp.580–594.
- Rana, A.S. & Zalzal, A.M.S., 1996. An Evolutionary Planner for Near Time-Optimal Collision-Free Motion of Multi-Arm Robotic Manipulators. In *International Conference on Control*. pp. 29–35.
- Rosenbaum, D.A., Loukopoulos, L.D. & Meulenbroek, R.G.J., 1995. Planning reaches by evaluation sorted posture.pdf. *Psychological Review*, 102(1), pp.28–67.
- Rosheim, M.E., 2006. *Leonardo's Lost Robot*, Minneapolis, USA: Springer.
- Sahar, G. & Hollerbach, J.M., 1986. Planning of Minimum - Time Trajectories for Robot Arms. *The International Journal of Robotics Research*, pp.90–100.

- Schraft, R.D., Schaeffer, C. & May, T., 1998. Care-O-bot: the concept of a System for Assisting Elderly or Disabled Persons in Home Environments. *IECON '98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society*, 4, pp.2476–2481.
- Sciavicco, L. & Siciliano, B., 2001. *Modelling and Control of Robot Manipulators*, London: Springer.
- Spong, M., Lewis, F. & Abdallah, C., 1993. *Robot Control : Dynamics, Motion Planning, and Analysis*, New York: IEEE Press.
- Stuckler, J., Grave, K., et al., 2009. Dynamaid : Towards a Personal Robot that Helps with Household Chores. In *Proceeding of RSS Workshop on Mobile Manipulation in Human Environments*.
- Stuckler, J., Schreiber, M. & Behnke, S., 2009. Dynamaid , an Anthropomorphic Robot for Research on Domestic Service Applications. In *4th European Conference on Mobile Robot (ECMR 09)*. pp. 87–92.
- Uno, Y. Kawato, M. Suzuki, R., 1989. Formation and Control of Optimal Trajectory in Human Multijoint Arm Movement. *Biol. Cybernetic*, 101, pp.89–101.
- Ur-Rehman, R. et al., 2010. Multi-objective path placement optimization of parallel kinematics machines based on energy consumption, shaking forces and maximum actuator torques: Application to the Orthoglide. *Mechanism and Machine Theory*, 45(8), pp.1125–1141.

- Vahrenkamp, N. et al., 2008. Adaptive Motion Planning for Humanoid Robots. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 22–26.
- Vahrenkamp, N. et al., 2011. RDT<sup>+</sup>: A parameter-free algorithm for exact motion planning. *2011 IEEE International Conference on Robotics and Automation*, pp.715–722.
- Wada, Y. et al., 2001. Quantitative examinations for multi joint arm trajectory planning--using a robust calculation algorithm of the minimum commanded torque change trajectory. *Neural networks : the official journal of the International Neural Network Society*, 14(4-5), pp.381–93.
- Wang, Q. & Zalzal, A.M.S., 1996. Genetic Control of Near Time-optimal Motion for an Industrial Robot Arm. In *IEEE International Conference on Robotics and Automation*. pp. 2592–2597.
- Yamazaki, K. et al., 2012. Home-Assistant Robot for an Aging Society. *Proceeding of IEEE*, pp.2429–2441.

---

# APPENDICES

---

## APPENDIX A:

### Inverse Kinematics of the Robot System

Referring to Fig. 3-1 and Fig. 3-2, the arm length is as follows;

$$OB = d_1 = 220 \text{ mm} , BC = a_2 = 260 \text{ mm}, CD = a_3 = 350 \text{ mm}$$

Solving for shoulder angle,  $\theta_1$ ;

$$P_d = \sqrt{P_z^2 + P_y^2} \quad \text{where, } P_z = P_d \cos \theta_1 \text{ and } P_y = P_d \sin \theta_1$$

$$\theta_1 = \tan^{-1} \frac{P_y}{P_z}$$

Solving for elbow angle,  $\theta_3$ ;

$$P_d = a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3) \quad P_x - d_1 = a_2 \sin \theta_2 + a_3 \sin(\theta_2 + \theta_3)$$

Summing both side yields;

$$P_d^2 + (P_x - d_1)^2 = (a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3))^2 + (a_2 \sin \theta_2 + a_3 \sin(\theta_2 + \theta_3))^2$$

$$P_d^2 + (P_x - d_1)^2 = a_2^2(c_2^2 + s_2^2) + a_3^2(c_{23}^2 + s_{23}^2) + 2a_2a_3c_3(c_2^2 + s_2^2c_3)$$

$$P_d^2 + (P_x - d_1)^2 = a_2^2 + a_3^2 + 2a_2a_3c_3$$

$$\cos \theta_3 = \frac{(P_d^2 + (P_x - d_1)^2 - a_2^2 - a_3^2)}{2a_2a_3}$$

$$\theta_3 = \cos^{-1} \left( \frac{(P_d^2 + (P_x - d_1)^2 - a_2^2 - a_3^2)}{2a_2a_3} \right)$$

Solving for elbow angle,  $\theta_3$ ;

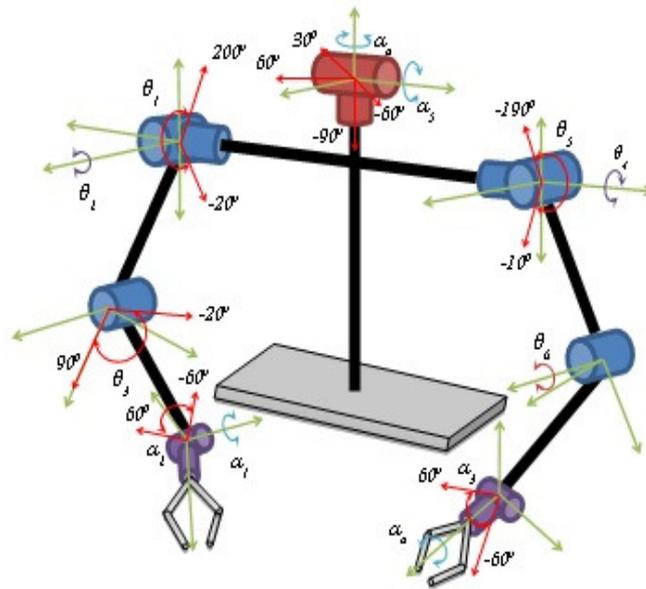
$$\tan \alpha = \frac{(P_x - d_1)}{P_d} = \frac{(P_x - d_1)}{\sqrt{P_z^2 + P_y^2}} \quad \alpha = \tan^{-1} \left( \frac{(P_x - d_1)}{\sqrt{P_z^2 + P_y^2}} \right) \quad s = \sqrt{(P_x - d_1)^2 + P_d^2}$$

$$\cos \beta = \frac{a_2 + a_3 \cos \theta_3}{s} = \frac{a_2 + a_3 \cos \theta_3}{s} \quad \beta = \cos^{-1} \left( \frac{a_2 + a_3 \cos \theta_3}{s} \right)$$

$$\theta_2 = \alpha \pm \beta = \tan^{-1} \left( \frac{(P_x - d_1)}{P_d} \right) \pm \cos^{-1} \left( \frac{a_2 + a_3 \cos \theta_3}{s} \right)$$

## APPENDIX B:

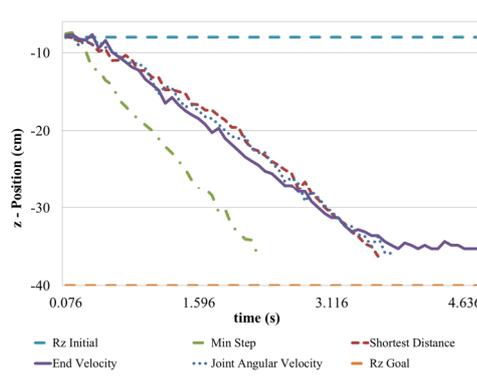
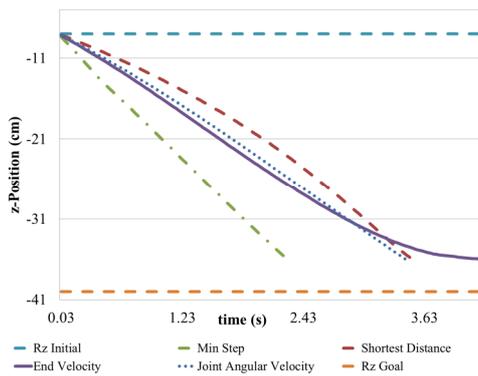
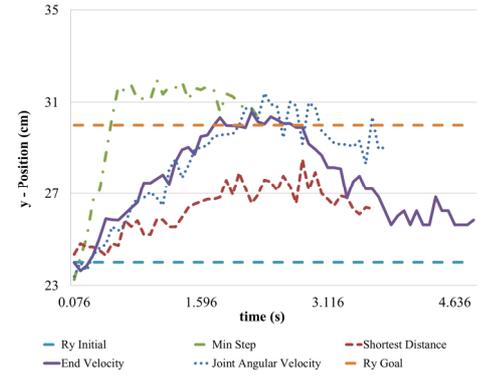
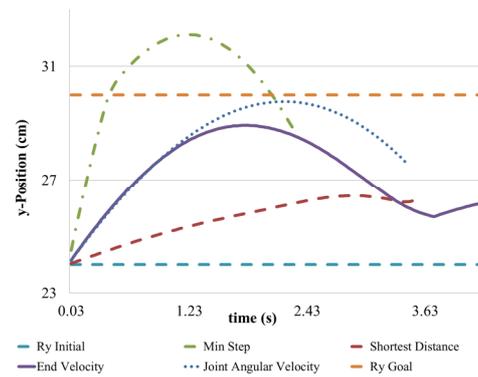
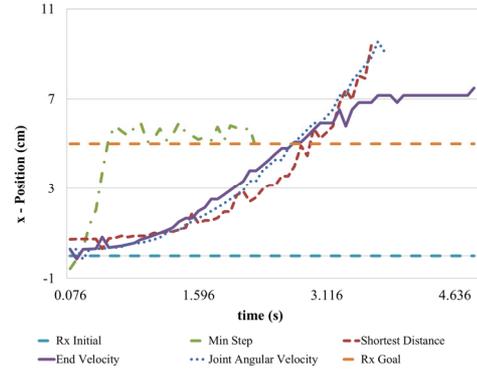
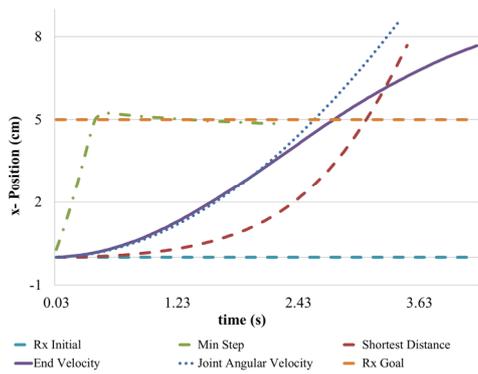
### Upper and Lower Limit of the Robot Joints



JOINTS	MIN. ANGLE	MAX. ANGLE
Left & Right Shoulder	$-20^{\circ}$	$200^{\circ}$
Left & Right Upper Arm	$-10^{\circ}$	$190^{\circ}$
Left & Right Lower Arm	$-20^{\circ}$	$90^{\circ}$
Head Tilt	$-90^{\circ}$	$30^{\circ}$
Head Pan	$-60^{\circ}$	$60^{\circ}$
Left & Right Hand Pitch & Roll	$-60^{\circ}$	$60^{\circ}$

**APPENDIX C:**

Execution time comparison for each objective functions in  $x$ ,  $y$  and  $z$  -axis (a) Simulation (b) Without load.



(a)

(b)