بسم الله الرحمن الرحيم

Palestine Polytechnic University

Electrical, Mechanical and Computer Engineering Departments

Graduation Project

Virtual Rehabilitation Robot (VRR)

Project Team

Khaled A. Azzam, Mechatronics Engineering

Mohannad H. Takrouri, Mechatronics Engineering

Feras E. Abu-Alfeilat, Communication Engineering

Mamoun Sh. Ideis, Computer Engineering

Supervisor:

Prof. Dr. Karim Tahboub

Hebron-Palestine

Feb, 2013

List of Contents:

CHAPTER7: Software System 75

List of Figures:

List of Tables:

Abstract

Robotic rehabilitation is a promising approach to rehabilitation of post stroke impairments. For that reason, a robotic arm is used for the upper-limb rehabilitation of stroke patients. This project is to study, analyze, design and implement a wirelessly controlled manipulator with three degrees of freedom. The manipulator has the possibility to operate in different planes. It allows patients to perform rehabilitation exercises while playing video games. These games are designed based on the rehabilitation protocol taking into account the patient's age, strength, mental condition and the affected regions of the brain. The device is built, tested, and evaluated with a stroke patient under direct supervision of a physiotherapy specialist. The results demonstrate the importance of the added features such as the inclined plane that is found to correspond to movements needed for daily activities.

Acknowledgment

We would like to express our gratitude to our parents, our families and our friends, whom we admire very much and have brought love, warmth, patience, and understanding to our lives, without whom we would be unable to complete our project. We want to thank Prof. Dr. Karim Tahboub, our supervisor, for his patience, help and guidance through the entire project who appreciated us for our work and motivated us and before all of them to God who made all the things possible. Special thanks to Dr. Akram Amro, Dr. Ali Abu-Gazalli, therapists Samih Dweik and Munther Oweiwi, also Al-Ihsan Society and Al-Ahli Hospital for their support and help. This graduation project has been supported by the Deanship of Graduate Studies and Scientific Research through "Distinguished Graduation Projects Fund".

Palestine Polytechnic University

Collage of Engineering and Technology &
College of Computer Engineering and Information Technology

Departments of Mechanical, Electrical and Computer Engineering

Virtual Rehabilitation Robot (VRR)

Project Team

Khalid A. Azzam	Feras I. Abu Abu-Alfielat
Mamoun Sh. Ideis	Mohannad H. Takrouri

According to the directions of the project supervisor and by the agreement of all examination committee members, this project is presented to the Departments of Mechanical, Electrical and Computer Engineering, for partial fulfillment of Bachelor of engineering degree requirements.

Supervisor Signature

Prof. Dr. Karim Tahboub

……………………………..

Committee Member Signatures

Dr. Murad Abo Sbeh ……………………………..

Dr. Yousef Alsweiti ……………………………..

Departments Head Signatures

Dr. Ramzi Qawasma ……………………..

Eng. Mohammad Awad ……………………..

Dr. Radwan Tahboub ……………………..

Feb, 2013

APPENDIX A
Virtual Rehabilitation Robot-Preliminary Thought

Virtual Rehabilitation Robot-Preliminary Thoughts

Khaled A. Azzam[1], Mohannad H. Takrouri[1], Feras I. Abu-Alfielat[2], Mamoun Sh. Ideis[2] and Karim A. Tahboub[1]
[1]Mechanical Engineering Department, [2]Electrical & Computer Engineering Department
Palestine Polytechnic University
Hebron, Palestine

Abstract—**Robotic rehabilitation is a promising approach to rehabilitation of post stroke impairments. For that reason, a robotic arm will be used for the upper-limb rehabilitation of stroke patients. This project is to study, analyze, design and implement a wirelessly communication manipulator with three degrees of freedom. The manipulator has a possibility to operate in different planes, allowing patients to perform rehabilitation exercises while playing video games. These games are based upon the rehabilitation protocol. They must be designed carefully for the exercises taking into account the patient's age, strength, mental condition and the affected locations of the brain. The final step will be testing the device at different planes (horizontal, vertical and inclined). The test must be done with real cases under a direct supervision of a specialist in physiotherapy. The results are essential to study the planes suitability for rehabilitation (especially the inclined plane) and the overall enhancement in the patient's condition.**

Keywords- Robotic; Rehabilitation; Post stroke; Upper limb; Wirelessly-operated mechanism; Video games.

I. INTRODUCTION

Impairments could happen due to trauma, stroke or brain injury, cerebral palsy, incomplete spinal cord injuries, multiple sclerosis. Other impairments could affect the musculoskeletal system such as bone fractures, muscular dystrophies, limb's burns or even a cut in the muscles or blood vessels [1]. According to World Health Organization (WHO) cerebrovascular accident (CVA) is the second most common cause of death in high and middle income countries and the sixth in low income countries. In 2008 for example about 6.15 million CVA patients died [2]. In Hebron, the number of CVA patients increased to 540 cases with a 139 new cases in just a year (04/08/2009 - 04/08/2010) [3].

In general, rehabilitation aims to help patients to restore, develop or maintain physical, psychosocial, cognitive and/or communication skills according to the type of impairments. While cognitive and communication skills are not important for some types of impairments that have a direct or physical effect on the affected limb such as bone fractures, burns…etc. They must be taken into account as in the case of CVA when the affected limb is the brain. Impairments need rehabilitations immediately after receiving the appropriate clinical treatment. The rehabilitation techniques and exercises differ and they are not the same for all impairments, they depend on the type of impairment, location and age of the patient.

Virtual rehabilitation is the use of virtual reality (VR) and virtual environments (VE) within rehabilitation. VR and VE can be described as a simulation of real world environments through a computer and experienced through a "human-machine interface" [4]. The importance of virtual rehabilitation arises due to the lack of motivation in the traditional rehabilitation with physiotherapy. The patient feels bored as a direct result of monotony and repetition in traditional therapeutic exercises. However, in order to get the motivation and enhance the rehabilitation process, an interesting computer game should be placed in front of the patient [5] and [6].

This paper presents preliminary thoughts about a virtual rehabilitation robot. It is consists of seven sections. Section II presents the need for such a robot in the rehabilitation processes. Section III describes an overview for the robot parts, and there functional specifications. Section IV listed the required sensors and actuators. Section V presents an overview for the communication wireless modules which will be used for connecting the robot side with the computer side. Section VI describes the computer and software part including (games design, levels description, and mapping between the workspace and the screen). Section VII presents the conclusion and the future work.

II. RECOGNITION OF THE NEED

From preliminary discussions with several occupational and physical therapists like Dr. Akram Amro [3], Dr. Ali Abu-Ghazala [7], Samih Dweik [1] and Munther Oweiwi [8], it has been concluded that there is a need for active-repetitive exercises under the therapists' supervision without the necessity of his direct intervention during exercises. However, therapists face a problem in motivating patients who get bored quickly.

Patients' motivation is achieved by linking rehabilitation exercises to well-designed computer games. This could be done by letting the interaction with the game to be through an active manipulation of a specially-designed mechanism. In rehabilitation exercises, there are other things which are crucial to complete the exercise correctly rather than moving a weak limb without any resistance in the space. For example, strengthening the affected muscles to restore the normal strength or just below the normal of a healthy person with the same age. After that, the patient should be able to withstand with normal resistance or loads that the person could face in daily life activity. The specially-designed mechanism is needed for strengthening the muscles rather than motivating the patient. Otherwise the patient could use a computer mouse or a small piece without the need for such a mechanism.

Usually the patient is asked to move small objects with different masses from one place to another. These objects are not always available with a large range that is suitable for all patients. In addition, the occupied volume of these objects in the clinic is large which could be annoying for both the therapist and the patient.

As a result, the required device must allow the interaction between the patient and the virtual environment (game), with a wide range of virtual weights. In addition, a record for the patient’s progress during the session is preferred.

As the device introduces a new technology for rehabilitation in Palestine, it should satisfy some features and requirements. According to physiotherapy specialist Dr. Akram Amro [3], a stroke rehabilitation device must allow the following features: 1) Eye-hand coordination, 2) Combined movements (functional movements), 3) Certain amount of resistance to increase the muscle strength, 4) Completion of visual tasks, 5) Passive movements for some cases, 6) Plane motion (horizontal plane for example to draw circles or squares) is suitable and useful. However, it is preferable to allow for three dimensional motions.

There are other requirements such as: 1) The device must be safe for both patient and therapist, 2) The device including the game must be suitable for patients of different ages, 3) The device must be convenient for patients with different sizes, 4) It must be suitable for both body sides, 5) Flexibility in the device is required. For example, if the patient is sitting rather than lying in the bed or sitting on a chair, he/she should be able to use the device [7] [8].

III. Conceptual Design and Functional Specification

Before building any device, a set of parameters must be considered, they are divided into two groups. The first one is related to device itself such as: 1) safety, 2) portability, 3) cost, 4) friction, 5) design simplicity, 6) workspace availability, 7) volume occupied by the device, 8) ability to afford heavyweights (also known as mechanical stiffness) 9) needing special components such as an adjustable table or chair. The second one is related to the patient such as: 1) size, 2) suitability for both hands if the patient suffers from left or right hand impairment.

For that, it is desired to design and produce an electro-mechanical device with measurement sensors that use computer game as an interaction tool. The device will be designed on the basis of the mechanical movement of the arm. The process starts when the patient holds the end-effector of the mechanical device with his/her arm. As the patient moves it in the workspace, sensors detect this motion and translate the change in position into electrical signals. These sensors are connected to a microcontroller called Arduino which in turn will send the signals to a ZigBee module. ZigBee a special module that allows signals to be transmitted and received wirelessly. The computer can now deal with signals as they represent the position coordination in the game that is displayed on the screen. The game could be moving things for example in which the patient is asked to move an object from one place to another. This movement should happened with the feeling of the actual weight of the object. To achieve this, the device through rotary dampers will resist the patient in his/her arm movement. The patient will perform an extra effort to move his/her arm while moving the object only. With different changing in resistance, the patient will feel that the objects are of different weights or materials. The block diagram of the whole system is shown in Figure 1.

The device is divided into systems, these are: 1) Mechanical device system, 2) Load system, 3) Sensors and interfacing system, 4) Computer and software system.

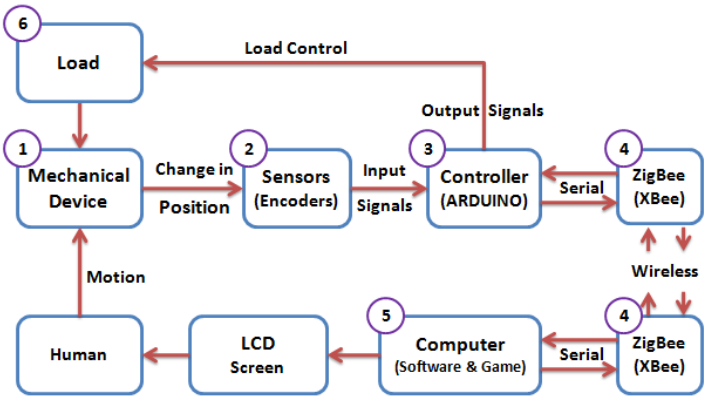


Figure 1: Block diagram for the whole system

A. Mechanical device (block 1)

The design is based on a parallelogram linkage. The main features of this arrangement are: good rigidity of the structure, direct drive of the manipulandum, which eliminates any backlash in the force/motion transmission and minimization the overall inertia, because most of the mass is either fixed, or close to the rotation axes [9]. The parallelogram manipulator choice is based on an open source design called Braccio di Ferro [9]. This design has two degrees of freedom either in vertical or horizontal plane. The project team will add a new degree of freedom called the configuration angle, and study its enhancement on the patient’s condition. However, the device with its three degrees of freedom will allow working in inclined planes in addition to vertical and horizontal planes.

This system consists of the mechanical arm with the attached components and the frame of the device as shown in Figure 2.

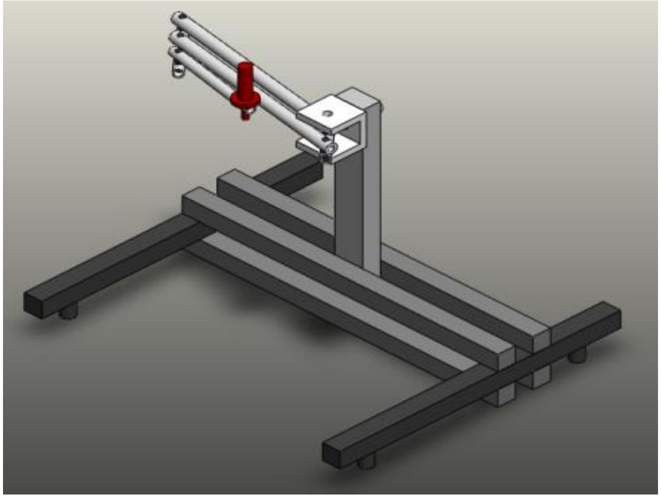


Figure 2: The mechanical arm with the frame only at the initial position

Figure 3 shows the top view of the manipulator. The manipulator must satisfy some requirements like mechanical rigidity (which is the ability to withstand with the load that result in this case from the patient). The design should compromise between the mechanical rigidity and the device weight. The suitable workspace for the patients is an elliptical shape with 800 mm × 400 mm [9], which is shown also in Figure 3. This workspace can be achieved using a manipulator with two degrees of freedom. Appendix A describes the coordinates of the end-effector in the robot coordinate system as a function of the joints' angles. The device must be back-drivable with low inertia and friction for the patients to get started with. The friction will be controlled to simulate the increasing load.

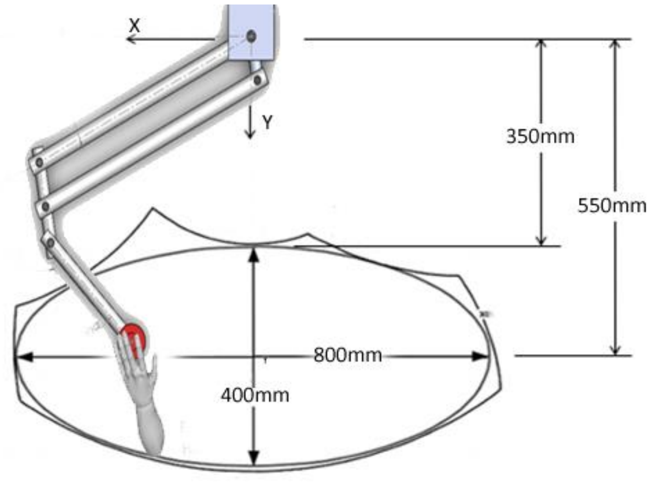


Figure 3: The workspace and the mechanical arm in two dimensions [9]

The mechanical arm will be built using Aluminum. It will be connected to the iron frame by a revolute joint which adds a new degree of freedom. This new degree of freedom will provide therapists with three configurations (horizontal, vertical and inclined in any desired angle) for the same device. Which could be useful for training different set of muscles with different exercises' levels in the rehabilitation session. In addition, a change in configuration angle causes gravity effects.

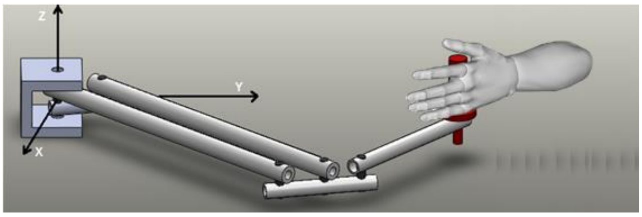


Figure 4: Left view of the manipulator in the horizontal plane (XY)

B. Load system (block 6)

The load is used to stop or reduce the motion of an object which is necessary in virtual environment to simulate the interaction of the patient with the virtual object. As the patient progresses in the game, he/she should be able to interact with different weights of objects. To achieve that, a resistance control is required without annoying the patient or affecting the virtual environment.

Rotary dampers are preferable for this type of device with revolute joints, but these dampers must have some requirements that are necessary such as an adjustable damping ratio and a bidirectional damping effect.

Dampers with adjustable damping ratio are called semi-active dampers which produce only a modulation of the damping forces in the controlled system according to the control strategy employed [10]. Thus the amount of damping can be tuned in real time.

Controlling dampers can be achieved by a command from the computer, but this command must be send wirelessly using ZigBee to Arduino, which in turn will produce the required controlling signal.

C. Sensors and interfacing system (blocks 2, 3, 4)

As the patient holds the end-effector of the mechanical device with his/her arm, the motion that results in the workspace is detected using motion sensors. These sensors translate the change in position into digital electrical signals.

The chosen motion sensors must afford a high resolution for determining the accurate position and the selected configuration. The sensors will be connected to the Arduino which allows receiving the input signals (sensors) and controlling the output signals (dampers).

One of the requirements of the device is portability. To achieve that, a greater movement freedom can be reached using wireless communication through a technology that covers larger area and allows for obstacles avoidance which could exist in the therapy session. Those requirements can be satisfied using a special module called ZigBee with two hardware parts which allow signals to be transmitted and received wirelessly. The first part will be connected to the Arduino and the other part will be connected directly to the computer. Each part is capable to transmit and receive data at the same time.

D. *Computer and software system (blocks 5)*

A computer is required to have a real time load control (dampers), interesting games with different levels, a record for the patient's progress during the session and synchronization between the patient's motion and the game.

The game requirements are achieved through a solid software such as C# which is a programming language that is simple, modern, general-purpose, strong type checking and object-oriented programming language, it was developed by Microsoft [11]. The other requirements depend on the computer specifications.

The data is received from the ZigBee module to locate the position of the end-effector on the screen and determine the used configuration that will be used to choose the suitable game. During the game, the computer must control the level of the game, record the patient's progress during the session and control the load.

IV. Sensors and Actuators

The motion of the mechanism in the workspace is detected using sensors. Three sensors will be used (two for the position) and the third for the configuration, these sensors translate the change in position into digital electrical signals. While the patient is progressing in the game, he/she should be able to interact with different objects' resistances, in order to achieve that, loads control is required.

A. *Encoders*

Sensors are needed to find the exact position of the patient's hand. Because the patient always holds the end-effector during the exercise, finding the end-effector position will give the patient's hand position. The end-effector position can be determined using three angles two of them correspond to the position in the plane. The third angle determines the chosen configuration. To measure these angles, rotary encoders are needed.

The specifications of the encoder must satisfy some requirements to complete the task with a minimum cost. The inertia of the encoder must be small and the frequency response must be high enough to maintain a good accuracy for the device.

Using an incremental encoder means that the initial position of the manipulator is unknown. To solve this problem an initial position of the manipulator with known values of the angles as shown in Figure 2 can be used.

B. *Dampers*

Dampers are passive components that depend on the speed of the affected force or torque. If the speed is small the damping force/torque is small and vice versa. These components are preferred in rehabilitation devices for safety reasons.

In contrast to active control devices, semi-active control devices cannot inject mechanical energy into the controlled system and, therefore, they do not have the potential to destabilize it. Examples of such devices are variable orifice dampers, controllable friction devices and dampers with controllable fluids (e.g., electro-rheological and magneto-rheological fluids) [10].

Ideas for implementing the damper include MR fluid. This method has many advantages like the speed response and the small size but it is available for linear damper only. An idea to construct one is by using rotary disc-type MRF dampers. A final choice will be using a small motor to control a normal rotary bidirectional damper.

V. Communication System

One of the most important features is the flexibility of the device. To achieve that, wireless communication will be used. Thus adds a greater movement freedom. By this way, the rehabilitation device can be used easily while patient is sitting or standing with the ability to move and transfer the device to different places in the room. This is achieved by choosing the appropriate distance between the patient who uses the mechanical device and the screen.

Another feature is portability of the device. The therapist can take the rehabilitation device to patient's home and reconstruct it in an easy way without worrying about wires. Using wires in the rehabilitation device can increase the probability of cutting them because of the mechanical movements of the device. As a result, this will reduce the reliability of the device.

The communication system in the rehabilitation device which is shown in Figure 5, consists of a controller (Arduino), XBee RF modules and interfacing circuit with the computer (XBee Explorer Dongle).

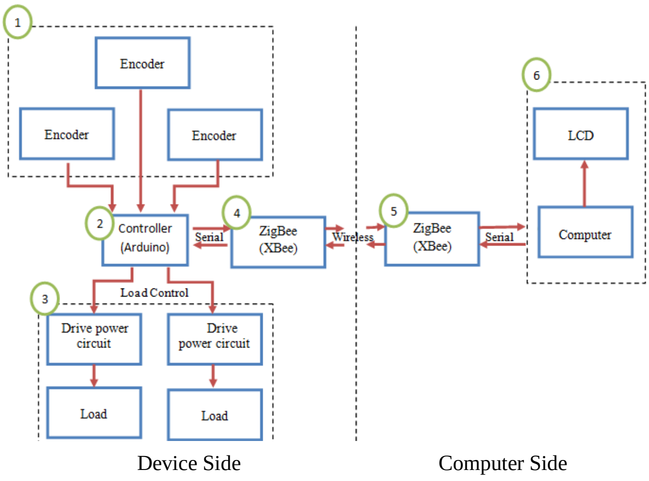


Figure 5: Communication System Block Diagram

VI. Software System

Stroke patients are typically seen for one or two half-hour sessions per day [12]. It was concluded from discussions with physiotherapist Dr. Akram Amro [13] that one session per a day with thirty minutes divided into two parts: about ten minutes for one of the reaching targets games, and the rest is for the following paths games.

Collect Money Game (CMG) is an example of the reaching targets games as shown in the left panel of Figure 6. In this game, the patient tries to move his/her arm in order to catch random targets.

Steady Hand Game (SHG) is an example of the following path games as shown in the right panel of Figure 6. The patient tries to follow the rod in order to finish the path and lead to the target with the least collisions as possible. The rod starts to be thicker as progressing in the game.

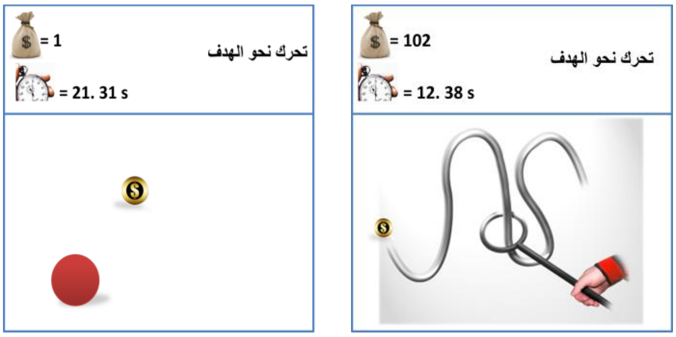


Figure 6: Left panel: CMG. Right panel: SHG

Both games are chosen for therapeutic session. They will be designed to get effective and correct therapeutic exercises. In addition, they will be enhanced to get more motivation and some usage of the cognitive skills.

One of these games is concerned with measuring the therapeutic level, and evaluates the patient's progress. For that reason, this game will be played at the beginning of each therapeutic session.

The game has ten different levels. A patient who finishes a level for three full successive sessions (contains CMG and SHG) without any problems will move to the next level. The main factors between the levels of the game that will change are: 1) The produced load (the maximum added load is 5 N.sec/m [6], [9]), 2) The time needed and the movement's speed to reach a goal. 3) The complexity of motion (targets positions in CMG, wire shape in SHG) and 4) The device configuration (horizontal, vertical or inclined).

On the other hand, the therapeutic process should store and retrieve the patient information and therapeutic progress, which can be achieved by a suitable data management system.

Moreover, the software system must control the signals that come from the sensors, represent it on the LCD Screen, and send the suitable signals to control the load, because all of the project's parts must work simultaneously in real time, to achieve the synchronization.

The user of the system (patient, physiotherapist or even the system administrator) will be allowed to access the patient's profiles, games options, and the device settings by graphical screen that is easy to use. The keyboard or the mouse of the computer can be used directly to enter the user interface screens or to modify the onscreen components.

Some of the evaluation methodologies for the patient progress are: 1) Section C of Rivermead Motor Assessment which is a therapist tool to evaluate the patient's therapeutic case by asking the patient to do fifteen simple arm exercises, and evaluate each movements independently [14]. 2) The progress reports that record the patient's parameters while he/she is playing the therapeutic games. Some of these parameters are the maximum and minimum angles for both the elbow and the shoulder joints which represent the range of motion for these joints, the movement speed of the elbow and the shoulder [15], the average patient’s reaction time and the maximum load that the patient can withstand with which indicates the muscles strength. 3) The games results and the progress of these results [16]. In Collect Money Game (CMG), the patient's performance is evaluated by a progress report that records the number of reached targets and reached bonus targets. However, in Steady Hand Game (SHG) the performance is evaluated by another progress report that records the number of collisions and the percentage of valid points, the number of points located on the ideal path and the completion time [17].

In order to enhance the patient's attention, motivation and motor recovery through brain plasticity, the therapeutic exercises is required to provide patients with some types of feedback such as visual, haptic, auditory and performance feedback.

CMG focuses on the attention and reaction cognitive skills. On the other hand, SHG focuses on the attention and concentration skills to avoid the collisions with the wire. SHG requires some perception skills that are used to choose the correct path through the whole wire to reach the target. It will be updated in a way to include the memory cognitive skills. Figure 7 explains the idea. A picture will be shown on the screen at the beginning of any level, it will then disappear after a short time. A wire will be shown on the screen with the previous picture but this time with new one. Each picture will be placed in different locations on the screen. The patient must move his/her arm navigating the wire and choose the correct target through the correct path.

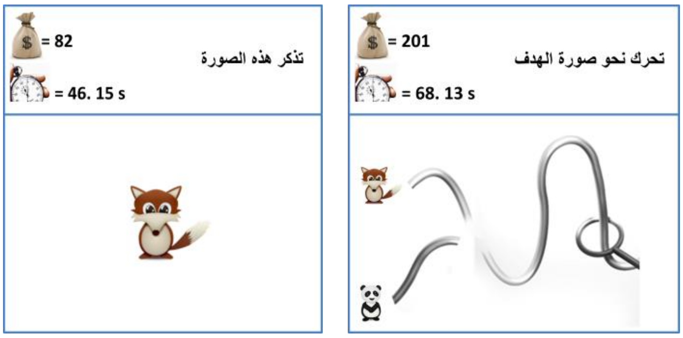


Figure 7: Enhanced SHG

As progressing in the session, the device configuration angle can be changed manually. This will affect the camera position (eye position) with respect to the game's objects only as shown in Figure 8. This allows patients to play the therapeutic games in different arm's positions.

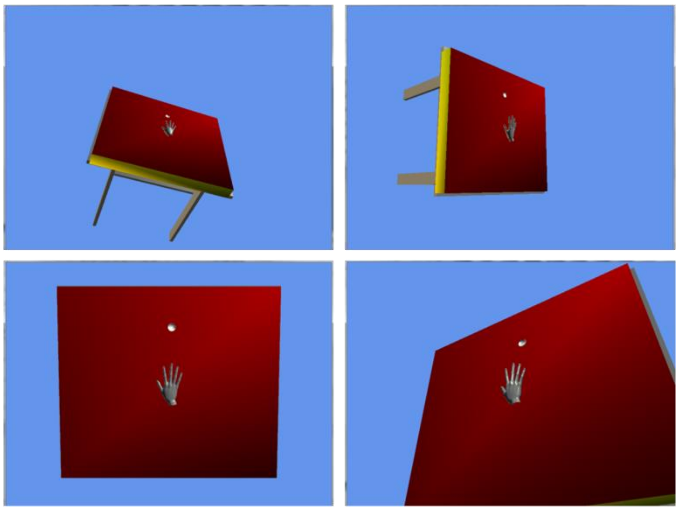


Figure 8: Different snapshots from different positions in CMG game

To locate the position of the hand in the workspace on the LCD screen, it is required that after each sample reading to map the joints' angles of the device on the game's window on the screen.

The device workspace is 800 mm width ranged between (-400 mm and 400 mm), and 400 mm height ranged between (350 mm and -750 mm), where the zero point is 550 mm on the y-axis away from the workspace origin point. On the other hand, the game's window is 800 pixel width, and 600 pixel height. The zero point is at the top left corner of the screen as shown in Figure 9. Suppose that $(\grave{x}, \grave{y})$ is the onscreen point, then:

$$\grave{x} = x + 400 \tag{1}$$

$$\grave{y} = 600 - ((y - 350) * 1.5) \tag{2}$$

where X and Y are the coordinate position and are derived in Appendix A.

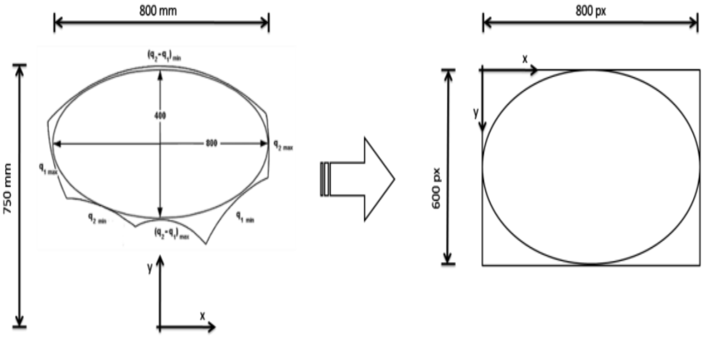


Figure 9: Left panel: The workspace of device. Right panel: The screen workspaces

VII. Conclusion and Future Work

It has been concluded that there is a need for virtual rehabilitation robots in Palestine. These robots can assist therapists during rehabilitation sessions to get the maximum possible benefit to patients.

Adding the third degree of freedom (configuration angle) could be useful for training different set of muscles with different exercises' levels in the rehabilitation session. This new feature allows the transition from designing two dimensions games to design three dimensions games.

Building dampers will be base on semi-active dampers type. This will allow controlling the load at real time without annoying the patient's interaction with the virtual environment.

After implementing the whole device, it should be tested on real cases under the supervision of therapists. And with the ability to design and install other types of games, more exercises can be done using this device to include other types of impairments in addition to strokes.

APPENDIX A: KINEMATIC ANALYSIS OF THE MANIPULATOR

The following equations describe the coordinates of the end-effector in the robot coordinate system as a function of the joints' angles:

$$\mathrm{X} = (L_1.\cos\Theta_2 + L_2.\cos(\Theta_2+\Theta_3)).\cos\Theta_1 \qquad \text{(A.1)}$$

$$\mathrm{Y} = L_1.\sin\Theta_2 + L_2.\sin(\Theta_2+\Theta_3) \qquad \text{(A.2)}$$

$$\mathrm{Z} = (L_1.\cos\Theta_2 + L_2.\cos(\Theta_2+\Theta_3)).\sin\Theta_1 \qquad \text{(A.3)}$$

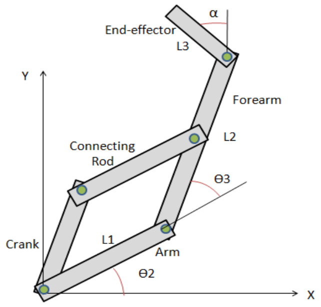


Figure 10: Top view of the manipulator in two dimensions before simplification

Where Θ_1 is the configuration angle. The crank and the connecting rod have no effect on the position of the end-effector which can be excluded from the analysis. To simplify the position equations, it can be assumed that the manipulator has two links only (Arm and Forearm) since the angle of the end-effector link (α) is constant and equals to 36.5°.

The relationship between rotational speed of the joints and hand speed is

$$\begin{bmatrix}\dot{X}\\ \dot{Y}\\ \dot{Z}\end{bmatrix} = \mathrm{J}\begin{bmatrix}\dot{\Theta}_2\\ \dot{\Theta}_3\end{bmatrix} \qquad \text{(A.4)}$$

The Jacobian matrix of the kinematic transformation:

$$J = \begin{bmatrix}(-L_1.\sin\Theta_2 - L_2.\sin(\Theta_2+\Theta_3)).\cos\Theta_1 & -L_2.\sin(\Theta_2+\Theta_3).\cos\Theta_1\\ L_1.\cos\Theta_2 + L_2.\cos(\Theta_2+\Theta_3) & L_2.\cos(\Theta_2+\Theta_3)\\ (-L_1.\sin\Theta_2 - L_2.\sin(\Theta_2+\Theta_3)).\sin\Theta_1 & -L_2.\sin(\Theta_2+\Theta_3).\sin\Theta_1\end{bmatrix} \qquad \text{(A.5)}$$

REFERENCES

[1] Dweik, Samih . Interview. 09 March 2012.

[2] World Health Organization, 2008, "The 10 leading causes of death by broad income group", who.int,

http://www.who.int/mediacentre/factsheets/fs310/en/index.html, June 2011, [Feb. 7, 2003].

[3] Dr. Amro, Akram. Interview. 11 March 2012.

[4] Holden, M. K., "Virtual environments for motor rehabilitation", 2005.

[5] S. Mazzoleni, et al, "Robot-aided therapy on the upper limb of subacute and chronic stroke patients: a biomechanical approach", Scuola Superiore Sant'Anna and Auxilium Vitae Rehabilitation Centre, 2011.

[6] A. Basteris, et al, "A tailored exercise of manipulation of virtual tools to treat upper limb impairment in Multiple Sclerosis".

[7] Dr. Abu-Gazala, Ali. Interview. 01 February 2012.

[8] Oweiw, Munther. Interview. 07 February 2012.

[9] M. Casadio, V. Sanguineti, P. G. Morasso, V. Arrichiello, "Braccio di Ferro: A new haptic workstation for neuromotor rehabilitation", Technology and Health Care 14:123–142, 2006.

[10] E. Guglielmino, T. Sireteanu, C. W. Stammers, G. Ghita and M. Giuclea (2008), "Semi-active suspension control improved vehicle ride and road friendliness", London: Springer.

[11] Ecma International Organization, C# language specification, 4th Edition, June 2006, ECMA-334.

[12] A. Alamri, M. Eid, R. Iglesias, S. Shirmohammadi, and A. El Saddik, "Haptic virtual rehabilitation exercises for post-stroke diagnosis", 2007.

[13] Meeting with physiotherapist Dr. Akram Amro, Mon 30-4-2012,5:00 PM to 5:45 PM.

[14] Collaborative evaluation of rehabilitation in stroke across Europe, Rivermead Motor Assessment.

www.nottingham.ac.uk/iwho/documents/rma.pdf, 2001, [Apr. 3, 2012].

[15] K. Morrow, C. Docan, G. Burdea, A. Merians, "Low-cost virtual rehabilitation of the hand for patients post-stroke", The State University of New Jersey, 2006.

[16] Y. Jung, S.-C. Yeh, J. Stewart, "Tailoring virtual reality technology for stroke rehabilitation: A human factors design", The USC-UT Consortium for the Interdisciplinary Study of Neuro plasticity and Stroke Rehabilitation, 2006.

[17] N. Pernalete, et al, "Development of an evaluation function for eye-hand coordination robotic therapy", California State Polytechnic University, Pomona and Western Michigan University, Kalamazoo USA, June 29 - July 1, 2011.

APPENDIX B

Virtual Rehabilitation Robot (VRR)

Virtual Rehabilitation Robot (VRR)

Khaled A. Azzam[1]*, Mohannad H. Takrouri*[1]*, Feras I. Abu-Alfielat*[2]*, Mamoun Sh. Ideis*[3] *and Karim A. Tahboub*[1]

[1]*Mechanical Engineering Department,* [2]*Electrical Engineering Department,* [3]*Computer Engineering Department*

Palestine Polytechnic University

Hebron, Palestine

Abstract— **Robotic rehabilitation is a promising approach to rehabilitation of post stroke impairments. For that reason, a robotic arm is used for the upper-limb rehabilitation of stroke patients. This project is to study, analyze, design and implement a wirelessly controlled manipulator with three degrees of freedom. The manipulator has the possibility to operate in different planes. It allows patients to perform rehabilitation exercises while playing video games. These games are designed based on the rehabilitation protocol taking into account the patient's age, strength, mental condition and the affected regions of the brain. The device is built, tested, and evaluated with a stroke patient under direct supervision of a physiotherapy specialist. The results demonstrate the importance of the added features such as the inclined plane that is found to correspond to movements needed for daily activities.**

Keywords—Robotic; Rehabilitation; Post stroke; Upper limb; Wirelessly-operated mechanism; Video games.

I. INTRODUCTION

Humans can be subjected to movement impairments due to trauma, stroke or brain injury, cerebral palsy, incomplete spinal cord injuries, or multiple sclerosis. Other impairments could affect musculoskeletal such as bone fractures, muscular dystrophies, limb's burns or even a cut in the muscles or blood vessels [1]. According to World Health Organization (WHO) cerebrovascular accident (CVA) is the second most common cause of death in high and middle income countries and the sixth in low income countries. In 2008 for example about 6.15 million CVA patients died [2]. In Hebron-Palestine, the number of new CVA patients in one year (August 2009-August 2010) reached 139 [3].

In general, rehabilitation aims to help patients to restore, develop or maintain physical, psychosocial, cognitive and/or communication skills according to the type of impairments. While cognitive and communication skills are not important for some types of impairments that have a direct or physical effect on the affected limb such as bone fractures, burns…etc, they must be taken into account as in the case of CVA when the affected organ is the brain. Impairments need rehabilitations immediately after receiving the appropriate clinical treatment. The rehabilitation techniques and exercises differ and they are not the same for all impairments, they depend on the type of impairment, location and age of the patient.

Virtual rehabilitation is the use of virtual reality (VR) and virtual environments (VE) within rehabilitation. VR and VE can be described as a simulation of real world environments through a computer and experienced through a "human-machine interface" [4]. The importance of virtual rehabilitation arises due to the lack of motivation in the traditional rehabilitation with physiotherapy. The patient feels bored as a direct result of monotony and repetition in traditional therapeutic exercises. However, asking patients to be engaged in interesting computer games could enhance their motivation and thus improve the rehabilitation process and results [5] and [6].

This paper presents a virtual rehabilitation robot with 3 degrees of freedom. Section II presents the need for such a robot in the rehabilitation processes. Section III gives an overview for the robot parts, and there functional specifications. Section IV lists the required sensors. Section V presents an overview for the communication system which connects the robot side with the computer side. Section VI describes the software system including games design and database management. Section VII presents preliminary evaluation results.

II. RECOGNITION OF NEED

From preliminary discussion with several occupational and physical therapists like Dr. Akram Amro [3], Dr. Ali Abu-Ghazala [7], Samih Dweik [1] and Munther Oweiwi [8], it has been concluded that there is a need for active-repetitive exercises under the therapists' supervision without the necessity of their direct intervention during exercises. However, therapists face a problem in motivating patients who get bored quickly. Patients' motivation is achieved by linking rehabilitation exercises to well-designed computer games. This is done by letting the interaction with the game to be through an active manipulation of a specially-designed mechanism. In rehabilitation exercises, there are other things which are crucial to complete the exercise correctly rather than moving a weak limb without any resistance in the space. For example, strengthening the affected muscles to restore the normal strength. The specially-designed mechanism is needed for strengthening the muscles and increasing their range of motion rather than only motivating the patient. Otherwise the patient could use a computer mouse or a small piece without the need for such a mechanism.

Usually the patient is asked to move small objects with different masses from one place to another. In addition, the occupied volume of these objects in the clinic is large which could be annoying for both the therapist and the patient. As a result, the required device must allow the interaction between the patient and the virtual environment (game) with some kind of resistance and a record for the patient's progress during the session. As the device introduces a new technology for rehabilitation in Palestine, it should satisfy some features and requirements. According to physiotherapy specialist Dr. Akram Amro [3], a stroke rehabilitation device must allow the following features: 1) Eye-hand coordination, 2) Combined movements (functional movements), 3) Certain amount of resistance to increase the muscle strength, 4) Completion of visual tasks, 5) Passive movements for some cases.

There are other requirements such as: 1) The device must be safe for both patient and therapist, 2) The device including the game must be suitable for patients of different ages, 3) The device must be convenient for patients with different sizes, 4) Suitable for both body sides, 5) Flexibility in the device. For example, if the patient is sitting on a chair rather than lying in the bed, he/she should be able to use the device [7] [8].

III. Conceptual Design and Functional Specification

Before building any device, a set of factors must be considered, they are divided into two groups: the first one is related to the device itself such as: safety, portability, cost, friction, design simplicity, workspace availability, volume occupied by the device, ability to afford heavy weights (also known as mechanical stiffness) and if it needs special components such as an adjustable table or chair. The second is related to the patient such as size and if he/she suffers from left or right hand impairment because it is important for the device to be suitable for both hands, but it is not necessary to be at the same time because the patient has either left or right hand impairment. For that, it is desired to design and produce an electro-mechanical device that relies on computer games as an interaction tool.

The process starts when the patient holds the end-effector of the mechanical device with his/her arm and starts moving in the workspace. Sensors detect this motion and translate the change in position into electrical signals. These sensors are connected to an ARDUINO controller that in turn sends the signals to ZigBee (a special module that allows signals to be transmitted and received wirelessly). The computer can now deal with them as they represent the position coordination in the game that is displayed on the screen, the game could be reaching things for example in which the patient is asked to reach objects that appear sequentially on the screen. Variable loads can be implemented through a linear damper that resists the movement of the patient. The load is changed and controlled by the therapist himself based on the patient situation. The block diagram of the whole system is shown in Fig.1. The device is divided into systems, these are: 1) Mechanical system, 2) Load system, 3) Sensors and interfacing system, 4) Computer and software system.

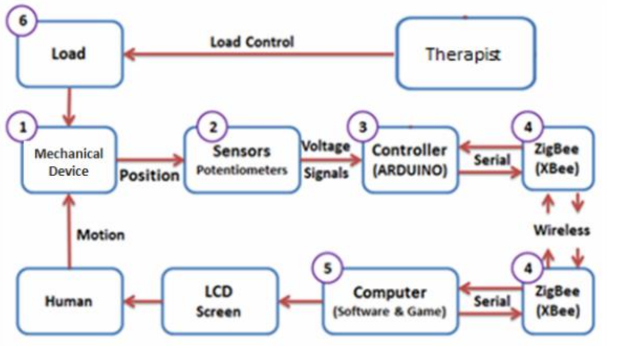


Fig. 1. Block diagram for the whole system

A. *Mechanical Device (block 1)*

The design is based on a parallelogram linkage as shown in Fig. 2. The main features of this arrangement are: good rigidity of the structure, direct drive of the manipulandum, which eliminates any backlash in the force/motion transmission and minimization the overall inertia, because most of the mass is either fixed, or close to the rotation axes [9]. The parallelogram manipulator choice is based on an open source design called Braccio di Ferro [9]. The original design has two degrees of freedom either in vertical or horizontal plane. A new degree of freedom is added, in this project, to the robot that allows working in an inclined plane. In addition, two important features are added to the device: portability through the wireless interfacing between the mechanical device and the computer and the flexibility introduced by the variable height of the robot to be suitable for all patients.

It satisfies the desired mechanical rigidity (which is the ability to withstand with the load that result in this case from the patient) while maintaining a relatively lightweight frame. The workspace for the device is an elliptical shape with 800 mm x 400 mm [9] as shown in Fig. 3 which is suitable for most patients. The device is as well back-drivable with low friction and inertia. The mechanical arm is built using Aluminum and connected to the iron frame by a revolute joint which adds the new degree of freedom. This new degree of freedom provides therapists with three configurations (horizontal, variably inclined and vertical).

B. *Load System (block 6)*

The load system plays an important part in VRR. An increasing load should resist the patient during therapeutic sessions. This load is necessary to increase the strength of the affected upper limbs muscles. To achieve this, the device through a damper resists the patient while moving his/her arm. The patient performs an extra effort to move the hand and with different changing in loads the patient feels the difficulty changing which is related to extra motivation. The changing load is controlled by the therapist himself based on the patient situation.

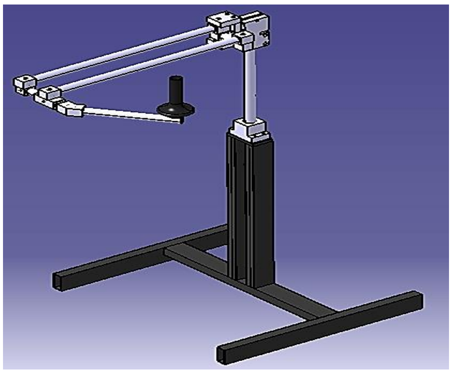


Fig. 2. The mechanical arm and the frame only

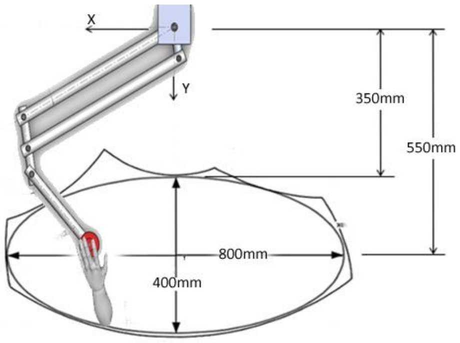


Fig. 3. The mechanical arm in two dimensions

C. *Sensors and Interfacing System (2, 3 and 4)*

As the patient moves the end-effector of the mechanical device, the motion that results in the workspace is detected using motion sensors. The chosen motion sensor must afford a high resolution for determining the accurate position. For this reason, three potentiometers are used (two for the position and the third for the configuration), these sensors translate the change in position into analog electrical signals and provide the information about the absolute position without the need of homing the device before operation. The potentiometers are interfaced with a microcontroller called ARDUINO which through its built in A/D converter converts the analog output of the potentiometers into digital signals.

One important feature of this novel device is portability. To partially achieve that, wireless communication between the mechanical device and the computer is desired. Thus, a special module called ZigBee that allows for signals to be transmitted and received wirelessly is used. ZigBee has two parts: one is connected to ARDUINO and the other is connected directly to the computer. Each part is capable to transmit and receive data at the same time.

D. *Computer and Software System*

A personal computer is required to host and run interesting games with different levels which calls for interacting with the human through the device sensors and wireless communication module (input) and a game screen (output) and to keep patients' progress. The game requirements are achieved through a capable programming language such as C# that is simple, modern, general-purpose, strong type checking and object-oriented. It is developed by Microsoft [10]. During the game, the compute controls the level of the game and records the patient's progress.

IV. Component Selection

The selection of components is based on a concurrent approach rather than a sequential approach. In other words, the components are chosen after an involved study of the requirements for the computer, mechanical, electrical communication and interfacing parts.

A. *Sensors*

Three potentiometers are used for the device. Two 5kΩ potentiometers are chosen for measuring the configuration angle and one of the parallelogram angles. One 50kΩ potentiometer is chosen for measuring the other parallelogram angle.

B. *Micro-Controller*

The chosen micro-controller is ARDUINO Uno. It has many advantages that makes it the preferred choice such as: user friendly, open source codes which are available for many applications and can be downloaded from the company's website, the simulation program, the available shields for many components and one of them is suitable for the wireless modules which allows the connection of modules easily. Other advantages are the upgradability, the modularity and the last important thing is that, it does not need a dedicated programmer as it can be programmed through the same cable that is used for the PC connection.

C. *Wireless Modules*

The required wireless modules are two identical XBees of series one. The device uses two XBee Pro 50mW wire antenna-series one with a range of 90 m. Other specification for the XBee Pro shows in Table I.

TABLE I. XBee Specifications

Specification	XBee PRO
Indoor/Urban Range	Up to (90 m)
Outdoor RF line-of-sight	Range Up to (500 m)
Transmit Power Output	50mW (17 dBm)
Supply Voltage	2.8 – 3.4 V
Transmit Current	45mA (@ 3.3 V)
Operating Frequency	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)
Number of pins	10ADC input pins and8 digital IO pins
Security	128-bit encryption

D. *Computer (Programming Language and Hardware)*

There are many choices for the programming language that can be used for the implementation of the system. C, C++, C#, Java and Visual Basic (VB) are five of popular programming languages for desktop applications programming. While, C++ is the object oriented version of C, C# is a simplified version of C++. Moreover, C# is easy to use, and allows accessing more library routines than C and C++. It contains a special environment for games programming, which is Microsoft XNA Framework that is used to design 2D, and 3D games. It needs XNA Game Studio Toolkit integrated with Microsoft Visual Studio to create a computer game. As a result, C# is the programming language that is used to implement the software system.

A midrange computer is chosen with the specifications shown in Table II.

TABLE II. COMPUTER SPECIFICATIONS

Processor	Inter (R), Core (TM), i5-2430M CPU @ 2.40GHz (4CPUs)
Memory	2048 MB RAM
Graphics	Inter (R) HD Graphics Family (785 MB)
Windows	Microsoft Windows 7

E. *Load System*

As discussed earlier, the load system plays an important part in VRR. An increasing load should resist the patient's movement during sessions. It is controlled manually by the therapist himself based on the patient situation. In general, dampers have three types: passive, active and semi-active. Passive dampers have constant damping ratio which cannot be changed. These types of dampers are not preferred since they lack the variability needed to increase the difficult level which corresponds to the extra effort the patient should do to move the manipulator and improve weak muscles.

The second type is active dampers where external energy is added to the system through an actuator. Using active dampers increases the complexity and cost of the device. Further, they might cause a safety issue by applying excessive active forces on the patient’s arm.

In contrast to active control devices, semi-active control devices cannot inject mechanical energy into the system. An example of such devices is variable orifice damper. This damper consists of a single rod piston with a variable orifice as shown in Fig. 4. Two methods for controlling the orifice are proposed: manual by hand and automatic through a motor. The therapist recommended the manual control with the ability to change the load to a suitable range and to have a self-feeling of the load before engaging the patient [3].

Fig. 5 shows the location of the damper, where it is connected almost parallel to the diagonal of the parallelogram (D) with a small distance from the two joints at both ends of the damper. This is done to avoid extra loads on these two joints and to ensure that no friction happens due to both ends of the damper on these joints.

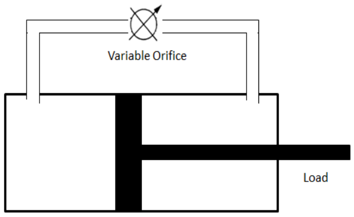


Fig. 4. Schematic of variable-orifice damper

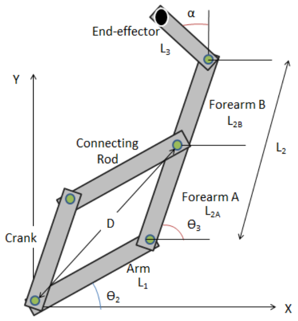


Fig. 5. The mechanical arm in 2-D with the damper position

V. COMMUNICATION SYSTEM

XBee parameters are set through serial port, this means that it has to stabilize a serial connection between the XBee board and computer device. By using a USB adapter, it can communicate with XBee through "USB Serial Port”. XBee Explorer Dongle needs no external cable. It does not provide any access to the radio beyond USB. On the other hand, it is a very small all-in-one device that is easy to carry in a pocket. It is powered via the USB connection. VRR has three options for the power supply. It can be a transformer with 5v DC output, a USB cable or a 9v battery with a 5v regulator for ARDUINO.

ARDUINO in turn supplies the other components (two XBee modules, three potentiometers). To reduce the power consumption, the XBee transceivers do not transmit the data all time. Instead, they transmit the signals after starting the game where the computer sends a request signal to ARDUINO after.

VI. SOFTWARE SYSTEM

Every therapeutic session consists of playing different computer games that are designed on the bases of therapeutic protocols. However, one of these games is dedicated to measuring the patient's level and evaluates the progress. It is played at the start of every session. Furthermore, these games help improving the patient's mental thinking if they require the usage of patient's cognitive skills.

This calls for storing and retrieving the patient information and progress. This is achieved by a suitable database management (DBM) system. Moreover, the software system deals with the signals that come from the sensors, and reflect them on the LCD game screen. The software system is divided into four integrated subsystems as shown in Fig. 6. Therapeutic Computer Games (TCGs) block consists of four different games. TCGs is designed to achieve eye-hand coordination which is the ability to coordinate the information received through the patient's eyes to control and move his/her arm.

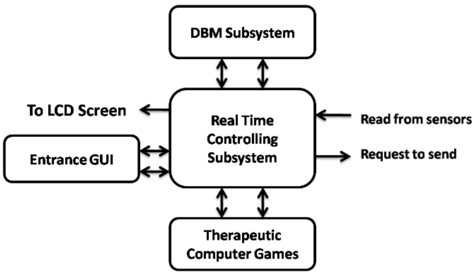
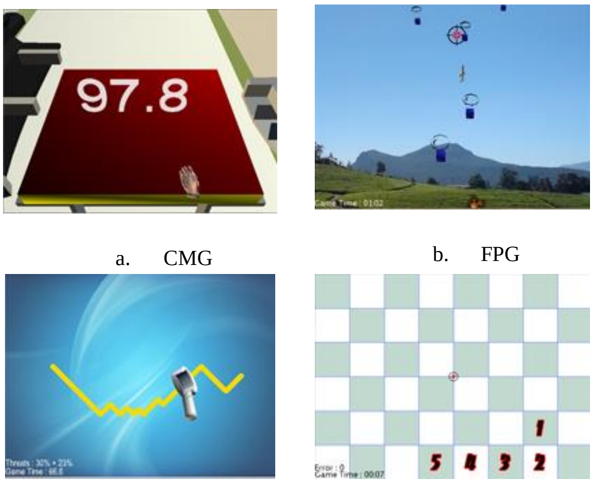


Fig. 6. Block Diagram for the Software System

Reaching targets, following paths, and object manipulation are some categories of therapeutic computer games [11]. Here, reaching targets and following paths are chosen for implementation. The difficulty of these games varies from zero to nine, with ten different levels. Collect Money Game (CMG) and Falling Parachutes Game (FPG) are two examples of reaching targets games. Where some targets are shown on the screen and the patient should move to reach them. Steady Hand Game (SHG) and Memorize Path Game (MPG) are two examples of following paths games. The patient must move in a specific path in order to finish the game. These games are chosen and designed as shown in Fig. 7.



a. CMG

b. FPG

c. SHG

d. MPG

Fig. 7. Therapeutic computer games

In order to enhance the patient's attention, motivation and motor recovery through brain plasticity, the exercises are required to provide patients with some types of feedback. It could be visual, haptic, auditory and performance feedback. Two screens are used, where the primary screen is for the therapist or for the administrator and the other one is for the patient. The patient cannot see anything except the games, while the therapist can see the games embedded in the window that controls them and allows for choosing the patient account with his/her reports.

Patient’s progress can be evaluated by different methodologies. First, according to Section C of Rivermead Motor Assessment by asking the patient to do fifteen simple arm exercises, and evaluate each movement independently [12]. Second, the progress reports record the patient's parameters while he/she is playing the game. Some of these parameters are the maximum and minimum angles for both the elbow and the shoulder joints which represent the range of motion for these joints, the movement speed of the elbow and the shoulder [13], the average patient’s reaction time and the maximum load that the patient can withstand with which indicates the muscles strength. Third, the games results and the progress of these results [14]. In CMG, the patient's performance is evaluated by a progress report that records the number of reached targets and reached bonuses. However, in SHG, the performance is evaluated by the number of collisions, the percentage of valid points, the number of points located on the ideal path and the completion time [15].

The software system is implemented following the Spiral Development Process as shown in Fig. 8. The whole system is divided into seven consecutive phases that are built on top of each other starting with a small nucleolus and ending with the fully integrated system that represents the whole system software. The implementation process is tested after each phase to verify its correctness.

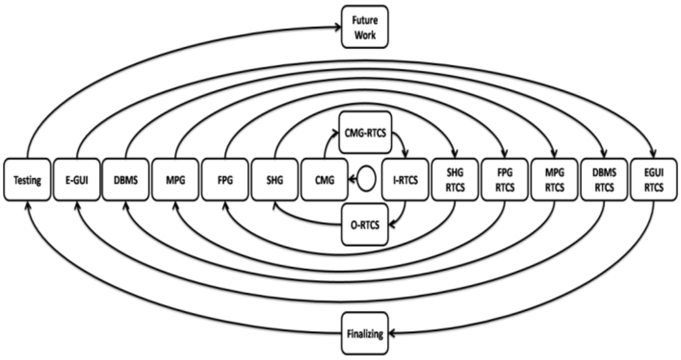


Fig. 8. Spiral development model

VII. Evaluation

The final step for VRR is to test it with a real patient. Physiotherapist Dr. Akram Amro evaluated the device with a stroke patient. The therapist first tested the workspace and the flexibility of the mechanical arm by moving it in the horizontal plane using both the left and the right hands. He changed the height of the device to check if it is suitable for the user in different situations. The load was set at the maximum and the therapist moved the end-effector. He was asked to decide if the load was small compared to the required practice. His response was that, the load was enough for a patient having stroke since the mission is improving the hand's motion and coordination under different loads and not body building.

After trying the four games, Dr. Amro thought that these games are motivating and allow for the completion of cognitive tasks and coordinated motions. All the games with their ability to increase the difficulty of levels by increasing the speed and the number of objects with random positions that are not predictive are very useful to kill the boringness of the traditional rehabilitation exercise.

The final test was the configuration test especially for the inclined plane. One of the main targets of the project is studying whether the inclined plane is suitable for stroke patients. The surprise was that the physiotherapist reported that the inclined plane is the most useful for performing exercises almost similar to activities of daily living like eating or shaving. The vertical plane is tested also which shows that it is useful because the patient has to lift the total weight of the device while performing the exercise so it is useful for muscle strength.

A female patient (NF, 56 years old) was asked to try the device. She preferred laying on the bed because she was not comfortable on the chair. She was asked about her first impression of seeing the device and if it scary. The answer was that the device is user friendly and not a scary device. Dr. Amro tested her concentration and coordination by asking her follow his finger in different positions. This test was performed before deciding which configuration is suitable for her. Dr. Amro decided that she needs the coordination test and this means that the suitable configuration is the horizontal plane. The patient started with Collecting Money Game in two parts one with a load and the other without. She also played Falling Parachutes Game, Memorizing Path Game and Steady Hand Game respectively. She faced initial difficulties in the first game until she got used to it, she found these games interesting especially the second and the third games.

VIII. Conclusion

The device is proved to be suitable for stroke patients with either left or right hand impairment. It is suitable also for the patient if he/she is sitting on a chair, lying on a bed or even standing. It is useful for the completion of cognitive tasks and allows for performing coordinated motions in the horizontal and inclined planes. It is useful for muscles strength especially in the inclined and the vertical planes. The novel feature of the inclined plane is found to be useful for performing activities similar to that of daily living. This device is ready for real clinical testing and for evaluating its effectiveness in physiotherapy following well designed.

Acknowledgment

We would like to express our gratitude to Dr. Akram Amro, Dr. Ali Abu-Gazalli, therapists Samih Dweik and Munther Oweiwi, also Al-Ihsan Society and Al-Ahli Hospital for their support and help. This project has been supported by the Deanship of Graduate Studies and Scientific Research in Palestine Polytechnic University (PPU) through "Distinguished Graduation Projects Fund".

Appendix A: Kinematic analysis of the manipulator

The following equations describe the coordinates of the end-effector as a function of the joints' angles:

$$X= L_1.\cos\theta_2 + L_2.\cos(\theta_2+\theta_3) + L_3.\cos(\theta_2 + \theta_3 + \alpha) \qquad (A.1)$$

$$Y= L_1.\sin\theta_2 + L_2.\sin(\theta_2+\theta_3) + L_3.\sin(\theta_2 + \theta_3 + \alpha) \qquad (A.2)$$

The crank, the connecting rod and the damper have no effect on the position of the end-effector which is excluded from the analysis. The dimensions of the mechanical arm are described in Table A.1.

TABLE A.1. Dimensions of the designed mechanical arm

Arm rod and connecting rod (L_1) [mm]	Length 550
Forearm, segment 1 (L_2) [mm]	Length 187
Forearm, segment 2 (L_3) [mm]	Length 283
Bending angle (α) [deg°]	36.5°

References

[1] Dweik, Samih . Interview. 09 March 2012.

[2] World Health Organization, 2008, "The 10 leading causes of death by broad income group", who.int, June 2011, [Feb. 7, 2003].

http://www.who.int/mediacentre/factsheets/fs310/en/index.html,

[3] Dr. Amro, Akram. Interview. 11 March 2012.

[4] Holden, M. K., "Virtual environments for motor rehabilitation", 2005.

[5] S. Mazzoleni, et al, “Robot-aided therapy on the upper limb of subacute and chronic stroke patients: a biomechanical approach”, Scuola Superiore Sant'Anna and Auxilium Vitae Rehabilitation Centre, 2011.

[6] A. Basteris, et al, "A tailored exercise of manipulation of virtual tools to treat upper limb impairment in Multiple Sclerosis".

[7] Dr. Abu-Gazala, Ali. Interview. 01 February 2012.

[8] Oweiw, Munther. Interview. 07 February 2012.

[9] M. Casadio, V. Sanguineti, P. G. Morasso, V. Arrichiello, “Braccio di Ferro: A new haptic workstation for neuromotor rehabilitation”, *Technology and Health Care 14*:123–142, 2006.

[10] Ecma International Organization, C# Language Specification, 4th Edition, June 2006,ECMA-334.

[11] A. Frisoli, et al, “Robot-mediated arm rehabilitation in Virtual Environments for chronic stroke patient: a clinical study”, Pasadena, CA, USA, May 19-23, 2008.

[12] "Collaborative Evaluation of Rehabilitation in Stroke across Europe, Rivermead *Motor Assessment*", www.nottingham.ac.uk/iwho/documents/rma.pdf.

[13] K. Morrow, C. Docan, G. Burdea, A. Merians, "Low-cost Virtual Rehabilitation of the Hand for Patients Post-Stroke", The State University of New Jersey, 2006.

[14] Y. Jung, S.-C. Yeh, J. Stewart, "Tailoring Virtual Reality Technology for Stroke Rehabilitation: A Human Factors Design", The USC-UT Consortium for the Interdisciplinary Study of Neuro plasticity and Stroke Rehabilitation, 2006.

[15] N. Pernalete,et al, "Development of an Evaluation Function for Eye-Hand Coordination Robotic Therapy", California State Polytechnic University, Pomona and Western Michigan University, Kalamazoo USA, June 29 - July 1, 2011.

CHAPTER 1

Introduction

1.1 Introduction

Humans can be subjected to movement impairments due to trauma, stroke or brain injury, cerebral palsy, incomplete spinal cord injuries, or multiple sclerosis. Other impairments could affect musculoskeletal such as bone fractures, muscular dystrophies, limb’s burns or even a cut in the muscles or blood vessels [1]. Cerebrovascular accident (CVA) also known as a stroke causes the sudden death of some brain cells due to lack of oxygen when the blood flow to the brain is impaired by blockage or rupture of an artery to the brain [2].

According to World Health Organization (WHO) CVA is the second most common cause of death in high and middle income countries and the sixth in low income countries. In 2008 for example about 6.15 million CVA patients died [3]. In Hebron, the number of new CVA patients in one year (August 2009-August 2010) reached 139 [4].

In general, rehabilitation aims to help patients to restore, develop or maintain physical, psychosocial, cognitive and/or communication skills according to the type of impairments. While cognitive and communication skills are not important for some types of impairments that have a direct or physical effect on the affected limb such as bone fractures, burns…etc, they must be taken into account as in the case of CVA when the affected organ is the brain. Impairments need rehabilitations immediately after receiving the appropriate clinical treatment. The rehabilitation techniques and exercises differ and they are not the same for all impairments, they depend on the type of impairment, location and age of the patient.

The rehabilitation training for CVA patients is not only the physical one; it depends on the affected lobe which is related to the lost functions. As a result, it is preferred in the rehabilitation sessions in addition to the physical exercises to include visual tasks with cognitive, attention and memory exercises. The period necessary to restore the limb to the normal condition varies depending on the type of CVA, location, age of the patient and the circumstances of treatment. For this reasons, the recovery is not always happened. A set of rehabilitation exercises for CVA patients is required such as mobility and strengthening exercises, walking programs and normal activities of daily living.

Symptoms of a stroke depend on the brain's affected area. The most common symptom is weakness or paralysis of one side of the body with partial or complete loss of voluntary movement or sensation in a leg or an arm [2].

Hemiplegia and Hemiparesis:

Hemiplegia is paralysis of one side of the body. Hemiparesis is weakness of one side of the body and is less severe than hemiplegia. Both are a common side effect of stroke or cerebrovascular accident [5]. Two cases were seen in Hebron, the first is a male patient (AHM, 81 years old). He suffers from hemiparesis in the left side of his body. He is able to move his hand and carry some weights, but he faces coordination difficulty and weakness of the left hand compared to the right. The second subject is a female (RI, 55 years old). She suffers from hemiplegia in the left side also, but she cannot move her left limbs, while she has a motion sensation if someone moves the affected limbs for her.

1.2 Virtual Rehabilitation (VR)

Virtual rehabilitation is the use of virtual reality and virtual environments within rehabilitation. Virtual reality and virtual environments can be described as a simulation of real world environments through a computer and experienced through a "human-machine interface" [6]. Virtual rehabilitation robot is a promising field of virtual reality that uses robots with an interactive game.

The importance of virtual rehabilitation arises due to the lack of motivation in the traditional rehabilitation with physiotherapy. The patient feels bored as a direct result of monotony and repetition in traditional therapeutic exercises. However, asking patients to be engaged in interesting computer games could enhance their motivation and thus improve the rehabilitation process and results [7] and [8].

A major advantage in all forms of VR is economy of scale; the same VR hardware can be used for various types of patients, as well as for various types of exercises done on those patients [9]. Another advantage is the motivation and interactivity during the therapy session. VR has two challenges which are the expensive cost of the systems and operation usually requires technical expertise [9].

1.3 Project Objectives

1. To build a robotic device that will be used in stroke rehabilitation.
2. To introduce a new technology for the domestic market.
3. To try a new method for virtual loads using frictional dampers.

1.4 Recognition of the Need

From preliminary discussion with several occupational and physical therapists like Dr. Akram Amro [4], Dr. Ali Abu-Ghazala [10], Samih Dweik [1] and Munther Oweiwi [11], it has been concluded that there is a need for active-repetitive exercises under the therapists' supervision without the necessity of their direct intervention during exercises. However, therapists face a problem in motivating patients who get bored quickly.

The importance of virtual rehabilitation arises from the lack of motivation in the traditional rehabilitation with physiotherapy, where the patient feels bored as a direct result of monotony and repetition in therapeutic exercises. In order to enhance the rehabilitation process, computer games are used, which work in parallel with the movement of the patient’s limb to get the patient motivated to do the therapeutic exercises correctly and joyously. Patients' motivation is achieved by linking rehabilitation exercises to well-designed computer games. This is done by letting the interaction with the game to be through an active manipulation of a specially-designed mechanism.

In rehabilitation exercises, there are other things which are crucial to complete the exercise correctly rather than moving a weak limb without any resistance in the space. For example, strengthening the affected muscles to restore the normal strength. The specially-designed mechanism is needed for strengthening the muscles and increasing their range of motion rather than only motivating the patient. Otherwise the patient could use a computer mouse or a small piece without the need for such a mechanism.

Usually the patient is asked to move small objects with different masses from one place to another. In addition, the occupied volume of these objects in the clinic is large which could be annoying for both the therapist and the patient. As a result, the required device must allow the interaction between the patient and the virtual environment (game) with some kind of resistance and a record for the patient's progress during the session.

Stroke and brain injuries requirements:

As the device introduces a new technology for rehabilitation in Palestine, it should satisfy some features and requirements. According to physiotherapy specialist Dr. Akram Amro [4], a stroke rehabilitation device must allow the following features:

1. Eye-hand coordination.
2. Combined movements (functional movements).
3. Certain amount of resistance to increase the muscle strength.
4. Completion of visual tasks.
5. Passive movements for some cases.

There are other requirements such as [10] and [11]:

1. The device must be safe for both patient and therapist.
2. The device including the game must be suitable for patients of different ages.
3. The device must be convenient for patients with different sizes.
4. Suitable for both body sides.
5. Flexibility in the device. For example, if the patient is sitting on a chair rather than lying in the bed, he/she should be able to use the device.

1.5 Previous Studies

The following studies were chosen for using virtual robots in therapy with Braccio di Ferro design. Braccio di Ferro is an open source robotic workstation for neurological rehabilitation for designing a manipulator with two degrees of freedom [12].

1. Robot-aided therapy on the upper limb of subacute and chronic stroke patients: a biomechanical approach

   By S. Mazzoleni, et. al.[13]. This 2 degrees of freedom (DOF) robotic system supports the execution of reaching movements in the horizontal plane only. If the patient is not able to complete the exercise; the robot helps him/her using motors that were fixed at the joints.

2. A tailored exercise of manipulation of virtual tools to treat upper limb impairment in Multiple Sclerosis

   By A. Basteris, et. al.[14]. This device is designed to treat cerebellar and motor symptoms in subjects with Multiple Sclerosis, by controlling a ‘virtual’ tool (a mass-spring system) under the effect of a resistive force. It allows for reaching movements in the horizontal plane only. The task difficulty and the degree of resistance are automatically adjusted to the individual patients’ impairment.

3. Startle reduces recall of a recently learned internal model

   By Z. Wright, et. al.[15].This study focuses on the performance of a recently learned task with a startle effect by using a two DOF haptic robot and an analog tone (used to create an auditory startle stimulus placed directly behind the subjects head).The reaching movements are in the horizontal plane only. It was shown that, the startle stimulus reduced performance of the learned task.

4. Integrating proprioceptive assessment with proprioceptive training of stroke patients

   By V. Squeri, et. al.[16]. This study uses a bimanual planar robot manipulandum. It consists of two identical Braccio di Ferro manipulandum, each with two degrees of freedom, mounted in a mirrored configuration on the same frame. The handles are grasped by the blind folded patient: the paretic hand is passively placed in one of 17 positions and the subject is asked to actively match the paretic hand position in space with the other hand.

5. Bilateral robot therapy based on haptics and reinforcement learning: feasibility study of a new concept for treatment of patients after stroke

   By M. Casadio, et. al.[17].This study uses a modified Braccio di Ferro manipulandum by mounting a horizontal bar at the end-effector for bimanual coordination. The bar is free to rotate around a vertical hinge. The task is to reach the target on the screen with an approximately horizontal bar.

6. Multi-joint arm stiffness during movements following stroke: implications for robot therapy

   By D. Piovesan, et. al.[18].This study was performed to estimate the endpoint stiffness of stroke survivors’ paretic arm during robot mediated therapy trials. Subjects were trained using a hitting task over a large workspace, while a robot provided an aiding force. This force, was aimed at the target, and remained constant until the target was reached, where it was suddenly turned off.

1.6 The Report Outline

Chapter 2 presents the components selection for the device. These components are the mechanical structure of the manipulator, the position sensors, the micro-controller, the wireless modules, the computer (including the required hardware and software) and the dampers.

Chapter 3 is devoted to the conceptual design and the functional specifications for each subsystem of the device with the budget of the project.

Chapter 4 is devoted to the mechanical design of the device. This chapter includes the strength, deflection and the bearing design for the manipulator, the configuration design and the base design including the buckling and weld design.

Chapter 5 summarizes the process of mapping the end-effector workspace into the screen workspace.

Chapter 6 presents a background on wireless technology architecture and types, the interfacing components and the wireless modules with the characteristics for each one.

Chapter 7 presents the software system that includes the Database Management Subsystem (DBMS), the Entrance Graphical User Interface (E-GUI), the Therapeutic Computer Games Software (TCGS) and the Real Time Controlling Subsystem Software (RTCSS).

Chapter 8 is devoted to the software system implementation, code writing and the programming issues. This chapter describes all the phases of the implementation spiral model which is divided into seven phases that depend on each other.

Chapter 9 presents the experimental results for each part of the system as a separate subsystem and the whole device as a complete system.

Chapter 10 is devoted to the evaluation results in which the whole device is tested with a stroke patient under direct supervision of a specialist in physiotherapy.

Chapter 11 presents the conclusion of the whole device as a complete system. In addition, the future works are found in this chapter.

CHAPTER 2

Conceptual Design and Functional Specifications

2.1 Introduction

The virtual rehabilitation robot is composed of different parts and components connected with each other. These parts have different types and shapes with different properties. The design must compromise between these properties to achieve the required shape and performance without affecting safety.

Before building any device, a set of factors must be considered, they are divided into two groups: the first one is related to the device itself such as: safety, portability, cost, friction, design simplicity, workspace availability, volume occupied by the device, ability to afford heavy weights (also known as mechanical stiffness) and if it needs special components such as an adjustable table or chair. The second is related to the patient such as size and if he/she suffers from left or right hand impairment because it is important for the device to be suitable for both hands, but it is not necessary to be at the same time because the patient has either left or right hand impairment.

2.2 Conceptual Design

It is desired to design and produce an electro-mechanical device that relies on computer games as an interaction tool. The process starts when the patient holds the end-effector of the mechanical device with his/her arm and starts moving in the workspace. Sensors detect this motion and translate the change in position into electrical signals. These sensors are connected to an ARDUINO controller that in turn sends the signals to ZigBee (a special module that allows signals to be transmitted and received wirelessly). The computer can now deal with them as they represent the position coordination in the game that is displayed on the screen, the game could be reaching things for example in which the patient is asked to reach objects that appear sequentially on the screen. Variable loads can be implemented through a linear damper that resists the movement of the patient. The load is changed and controlled by the therapist himself based on the patient situation. The block diagram of the whole system is shown in Figure 2.1.

The device is divided into systems, these are:

1. Mechanical device system.
2. Load system.
3. Sensors and interfacing system.
4. Computer and software system.

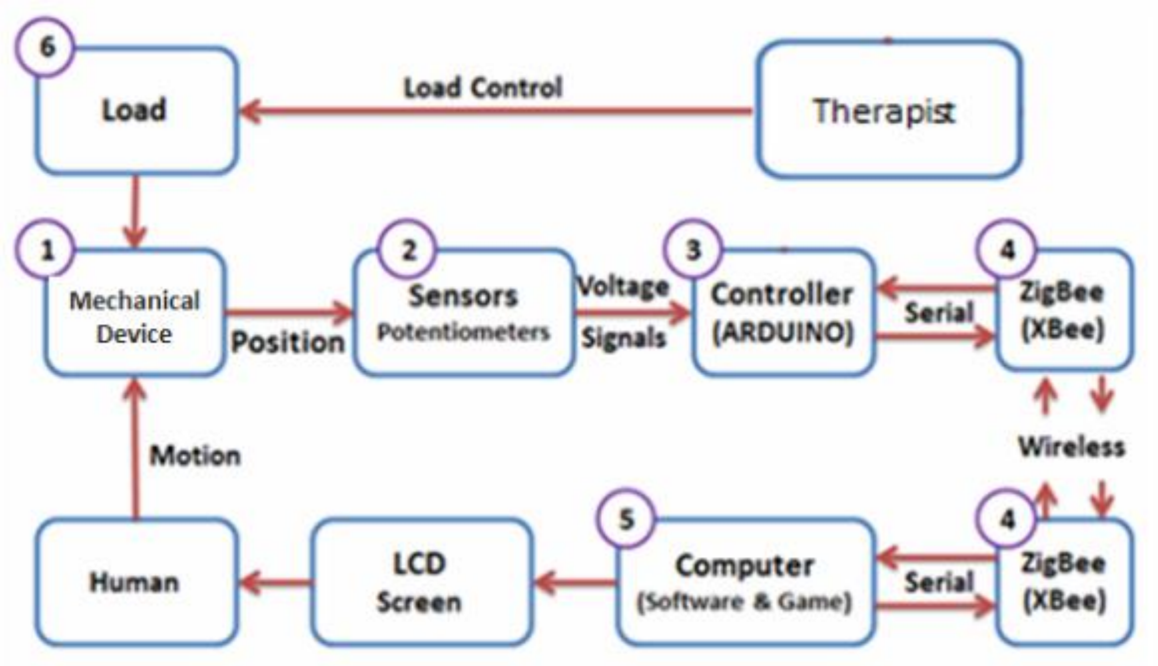


Figure 2.1: Block diagram for the whole system

2.3 Functional Specification

In this section, more details are given for each system with its related blocks in Figure 2.1.

2.3.1 Mechanical device (block 1):

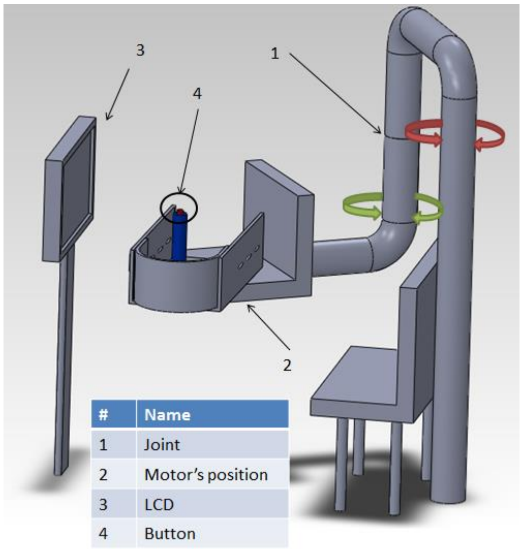
Four different manipulators' designs are studied for the mechanical structure of the device. Some of them are widely used in rehabilitation such as parallelogram and cylindrical manipulators. Other designs are suggested and investigated with a list of advantages and disadvantages for each one.

These designs are:

1. First preliminary design of the virtual rehabilitation robot.
2. Gantry manipulator.
3. Cylindrical manipulator.
4. Braccio di Ferro design.

2.3.1.1 First Preliminary Design of Virtual Rehabilitation Robot

At first, it is thought of making the device similar to the dental clinic chair. The robotic arm shaft moves around its axis (red arrow), connected to another piece which is also rotate about its axis (green arrow) using a joint as shown in Figure 2.2.



#	Name
1	Joint
2	Motor's position
3	LCD
4	Button

Figure 2.2: First preliminary design

The disadvantages of such a design are:

1. If the patient is unable to sit on the chair and cannot perform the exercises except on the bed, he/she cannot use this device.
2. The device is bulky and heavy.
3. It is not portable.

As a result, this design option is not considered anymore.

2.3.1.2 Gantry Manipulator

The second design is based on Cartesian robots which are not widely used in rehabilitation. The idea is using a Gantry manipulator as shown in the left panel of Figure 2.2 with some modifications such as: the device is converted into 2D by preventing the end effector from sliding in the vertical axis, it is made horizontal with a sliding bar at the end of it to insure that the patient’s elbow is fixed in the correct position, and finally, the legs are made short for not obstructing the patient’s vision, as shown in the right panel of Figure 2.3.

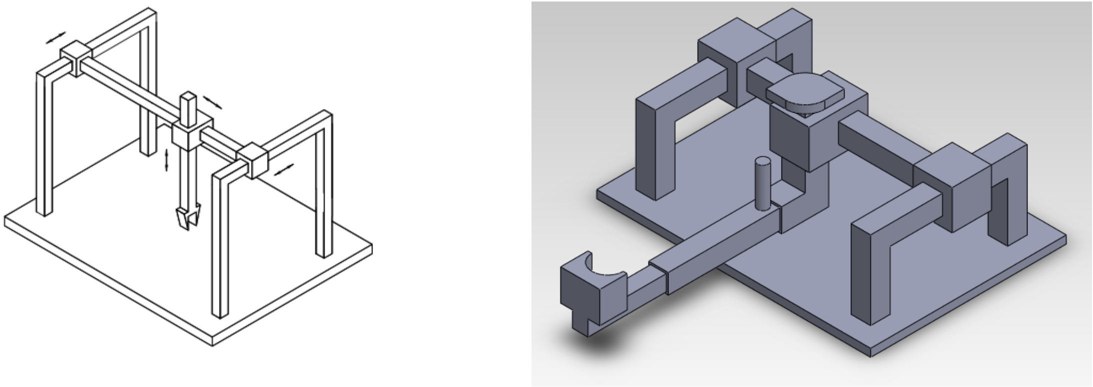


Figure 2.3: Left panel: Gantry manipulator. Right panel: Modified Gantry manipulator

This design has many advantages such as:

1. High portability.
2. It can afford heavyweights.
3. Suitable for installing linear dampers in two dimensions.
4. Easy to find the accurate position of the end effector using linear encoders.
5. The device is not obstructing the patient's vision.
6. It makes available a rectangular workspace with large volume.
7. Suitable for patient with left or right hand impairment.
8. Simple and low cost design.

9. Suitable for patients with different sizes in which they can set on a chair with adjustable height and it is also appropriate for patients using wheelchairs.
10. The volume occupied by the device is relatively not large.

The disadvantages are:

1. The design cannot be used for inclined or vertical configuration.
2. The problem of friction and backlash.
3. As the joints are prismatic; the patient is not moving freely as if they are revolute joints.

As a result, this design option is not considered as an option.

2.3.1.3 Cylindrical Manipulator

This manipulator which is shown in Figure 2.4 is portable, simple in design, not expensive, relatively small volume and lightweight and the position of the end effector is easy to be found from its distance and angle related to the origin. The disadvantages are: friction exists between the end effector and the rail, the workspace is not a rectangular and if it is on the right side for example, it must be moved completely to the other side for patients with left hand impairment. As a result, this design option is not considered as an option.

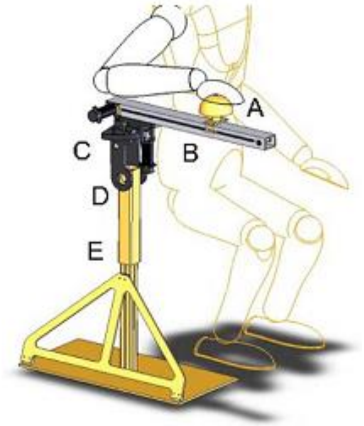


Figure 2.4: Cylindrical manipulator [19]

2.3.1.4 Braccio di Ferro

Braccio di Ferro is an open source design, is intended to foster the dissemination of robot therapy. This design is based on a parallelogram linkage as shown in Figure 2.5, with direct drive by the motors: one motor drives the arm of the manipulandum, which is jointed with the forearm, and the other motor drives a crank, which transmits the motion to the forearm by means of a connecting rod. The main features of this arrangement are: 1) good rigidity of the structure; 2) direct drive of the manipulandum, which eliminates any backlash in the force/motion transmission; 3) minimization the overall inertia, because most of the mass is either fixed, or close to the rotation axes [12].

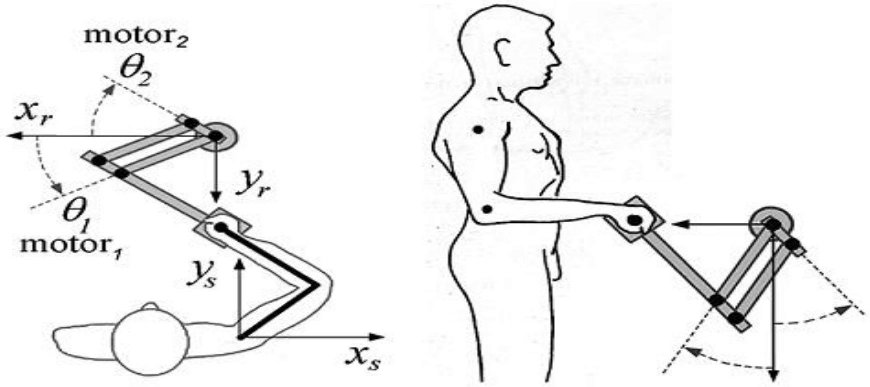


Figure 2.5: Braccio di Ferro. Left panel: Horizontal plane arrangement. Right panel: Vertical plane arrangement [12].

This design is found to be the best suitable one to implement. In addition to the above features, the design is modified to allow for motions in inclined planes and suitable for different heights. The original design has two degrees of freedom either in vertical or horizontal plane. A new degree of freedom is added, in this project, to the robot that allows working in an inclined plane. In addition, two important features are added to the device: portability through the wireless interfacing between the mechanical device and the computer and the flexibility introduced by the variable height of the robot to be suitable for all patients.

This system consists of the mechanical arm with the attached components and the frame of the device as shown in the left panel of Figure 2.6. It satisfies the desired mechanical rigidity (which is the ability to withstand with the load that result in this case from the patient) while maintaining a relatively lightweight frame. The mechanical arm in two dimensions is shown in the right panel of Figure 2.6 and the dimensions of the links are listed in Table 2.1. The workspace for the device is an elliptical shape with 800 mm x 400 mm [12] as shown in Figure 2.7 which is suitable for most patients. The human center with respect to the origin of the manipulator is located 800 mm on the positive of the Y axis which is shown in Figure 2.7. This distance is the recommended one for patients with normal sizes. The device is as well back-drivable with low friction and inertia. The mechanical arm is built using Aluminum and connected to the iron frame by a revolute joint which adds the new degree of freedom. This new degree of freedom provides therapists with three configurations (horizontal, variably inclined and vertical).

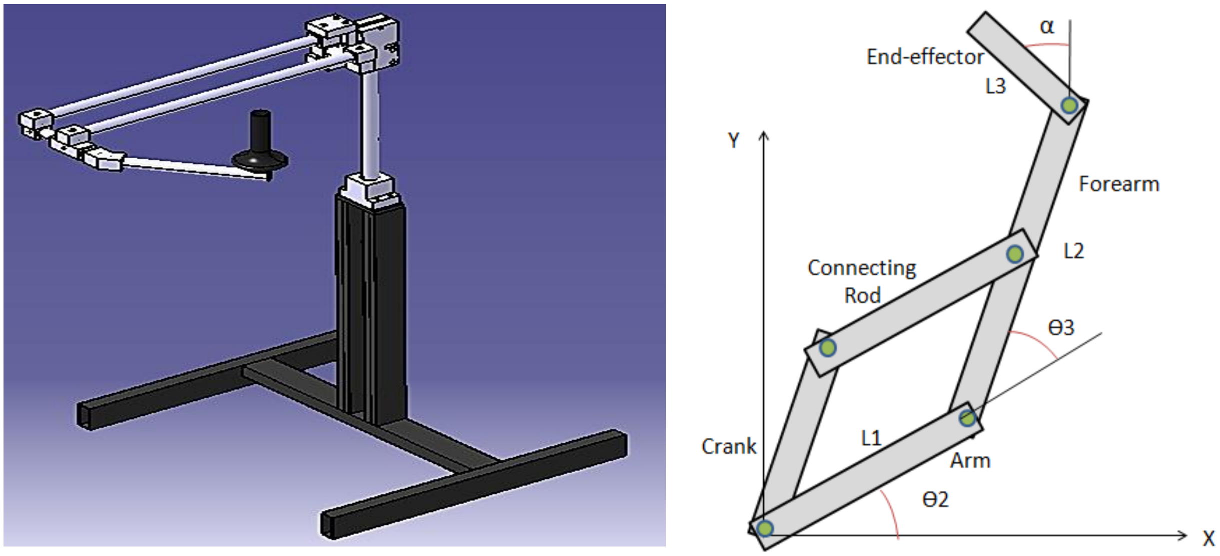


Figure 2.6: Left panel: The mechanical arm and the frame only. Right panel: The mechanical arm in two dimensions.

Table 2.1: Overall features of the designed mechanical arm

Workspace (elliptical) [cm]	80 x 40 [cm]
Arm rod and connecting rod (L_1) [mm]	Length 550; tube Φ 25; thickness 5
Forearm, segment 1 (L_2)	Length 187; tube Φ 25; thickness 5
Forearm, segment 2 (L_3)	Length 283; tube Φ 25; thickness 5
Bending angle (α) [deg°]	36.5°

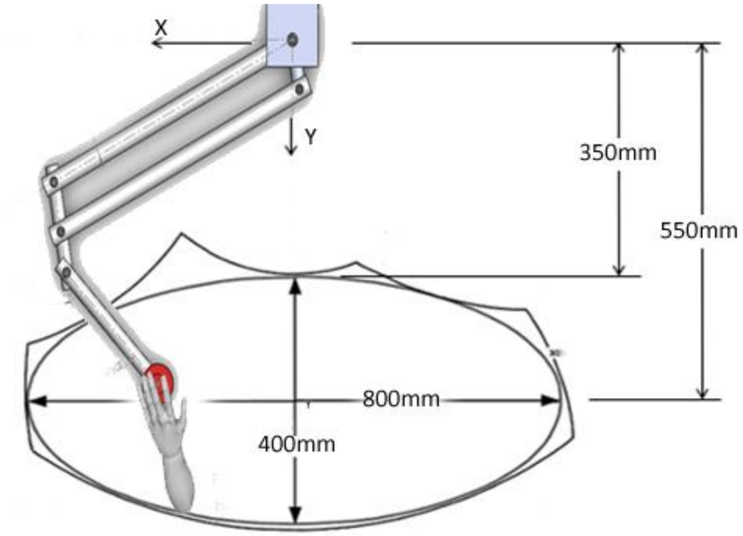


Figure 2.7: Elliptical workspace [12]

2.3.2 Load system (block 6):

The load system plays an important part in VRR. An increasing load should resist the patient during therapeutic sessions. This load is necessary to increase the strength of the affected upper limbs muscles. To achieve this, the device through a damper resists the patient while moving his/her arm. The patient performs an extra effort to move the hand and with different changing in loads the patient feels the difficulty changing which is related to extra motivation. The changing load is controlled by the therapist himself based on the patient situation.

2.3.3 Sensors and Interfacing System (blocks 2, 3, 4):

As the patient moves the end-effector of the mechanical device, the motion that results in the workspace is detected using motion sensors. The chosen motion sensor must afford a high resolution for determining the accurate position. For this reason, three potentiometers are used (two for the position and the third for the configuration), these sensors translate the change in position into analog electrical signals and provide the information about the absolute position without the need of homing the device before operation. The potentiometers are interfaced with a microcontroller called ARDUINO which through its built in A/D converter converts the analog output of the potentiometers into digital signals.

One important feature of this novel device is portability. To partially achieve that, wireless communication between the mechanical device and the computer is desired. Thus, a special module called ZigBee that allows for signals to be transmitted and received wirelessly is used. ZigBee has two parts: one is connected to ARDUINO and the other is connected directly to the computer. Each part is capable to transmit and receive data at the same time.

2.3.4 Computer and Software System (blocks 5):

A personal computer is required to host and run interesting games with different levels which calls for interacting with the human through the device sensors and wireless communication module (input) and a game screen (output) and to keep patients' progress. The game requirements are achieved through a capable programming language such as C# that is simple, modern, general-purpose, strong type checking and object-oriented. It is developed by Microsoft [20]. During the game, the compute controls the level of the game and records the patient's progress.

CHAPTER 3

Components Selection

3.1 Introduction

The selection of components is based on a concurrent approach rather than a sequential approach. In other words, the components are chosen after an involved study of the requirements for the computer, mechanical, electrical communication and interfacing parts.

The components are the position sensors, the micro-controller, the wireless modules, the computer (including the required hardware and software) and the load system. The following subsections explain the reasons for choosing each one in details and summarize the selection procedure that has been done during the first, summer and last semesters.

3.2 Position Sensors

A position sensor is any device that is used for position measurement. It can be either an absolute position sensor or a relative one (displacement sensor). Virtual Rehabilitation Robot has three angles to be measured, starts with the configuration angle (Θ_1) which determines the required plane before the rehabilitation process starts. The other two angles determine the position coordination of the end-effector to be mapped on the screen. The angle between the base and the arm is called (Θ_2) and the angle between the arm and the forearm is called (Θ_3) as explained in Figure 3.1.

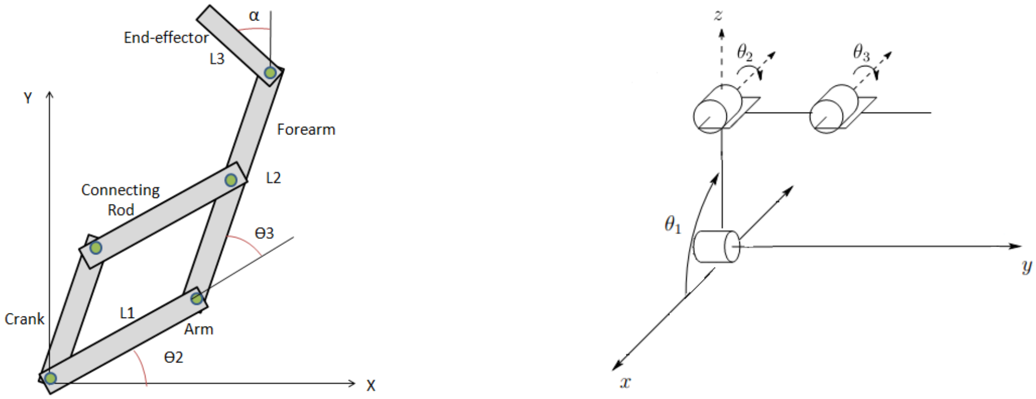


Figure 3.1: Left panel: The mechanical arm in 2-D. Right panel: The vertical configuration ($\Theta_1 = 90^{\circ}$)

The most common angular position sensors are encoders and potentiometers. These two types are chosen as the first decision to be studied. Potentiometers are chosen first as they are simple in structure, inexpensive and relatively accurate comparable to the other inexpensive sensors. The required potentiometers' resistance should be linearly changed as the change in the position. The problem is the lack of this type of potentiometers in the domestic market and that the available potentiometers are nonlinear as the one shown in Figure 3.2. Tests are performed using these nonlinear potentiometers and the results are unaccepted for two reasons. First of all, the accuracy is bad. The second reason is that, the hand's position which corresponds to the coordination of the manipulator which is mapped on the screen is shacking all the time even if the manipulator is fixed.

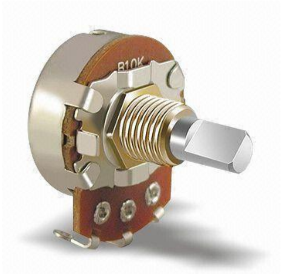


Figure 3.2: Non-linear potentiometer.

After that, the eyes are on rotary encoders. Encoders have two types: absolute encoders and incremental encoders as the one shown in Figure 3.3. The advantage of using the absolute encoder over the incremental one appears in case of the power which supplies the encoder is lost. An absolute encoder can continue from the position which it lies on even if it is changed during the power lost. While the incremental encoder starts counting from zero. For this reason, the absolute encoder is double in the price relative to the incremental one.

Figure 3.3: An incremental encoder.

An incremental encoder can still be used if a home position is found. This position relates the initial position of the device to the beginning of count from zero. If that happens, the incremental encoder can be used instead of the absolute encoder. An initial position for the device is chosen as shown in Figure 3.4. This position is unique for the manipulator. The therapist or the patient should always start the therapeutic session from this position in order to get the correct coordination.

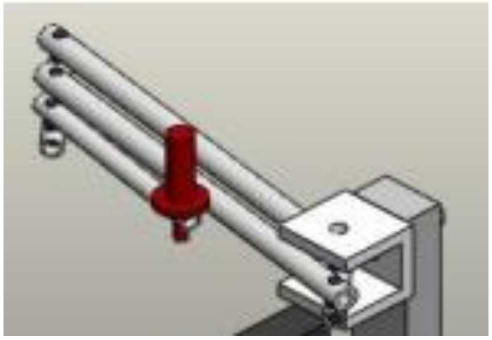


Figure 3.4: The mechanical arm at the initial position (home position).

The required incremental encoder should satisfy some requirements such as: the number of pulses per revolutions, the size and the input/output voltage. As the number of pulses increase, the accuracy increases. The available encoders in the stores are 300, 500, 1000 pulse/rev. the first two types have the same price, so the appropriate choice is the one with the higher accuracy as the 300 pulse/rev is equivalent to 1.2 degrees for each pulse and the 500 pulse/rev is equivalent to 0.72 degree. The accuracy of the encoder with the 500 pulse/rev is accepted in the computer part for position mapping but it is unaccepted for the huge size which is unsuitable for the links' diameters and makes the device ugly.

Another problem appears in the input/output voltage. The input voltage is 12v so an external power supply is needed other than the Arduino which supplies 5v only. The output voltage is 12v which needs voltage regulator for each channel of each encoder to be at the normal range of the Arduino input voltage. Tests are performed on this type of encoder with an external power supply and the common voltage regulator (7805). It has been found that, these regulators are not applicable for encoders since they are slow in voltage regulation and that the encoders have a huge number of pulses. The solution is using optocouplers transistors or using an encoder with the same specifications as the previous one except the input/output voltage is 5v. This encoder is double of the 12v input/output price. Three linear potentiometers are used as shown in Figure 3.5. The three potentiometers are used for the device. Two 5kΩ potentiometers are chosen for measuring the configuration angle and one of the parallelogram angles. One 50kΩ potentiometer is chosen for measuring the other parallelogram angle.

Figure 3.5: Linear increment potentiometer.

3.3 Micro-Controller

The chosen micro-controller is ARDUINO Uno as shown in Figure 3.6. It has many advantages that makes it the preferred choice such as: user friendly, open source codes which are available for many applications and can be downloaded from the company's website, the simulation program, the available shields for many components and one of them is suitable for the wireless modules which allows the connection of modules easily. Other advantages are the upgradability, the modularity and the last important thing is that, it does not need a dedicated programmer as it can be programmed through the same cable that is used for the PC connection.

Figure 3.6: Arduino Uno.

3.4 Wireless Modules

Using wires in the rehabilitation device increases the probability of cutting them because of the mechanical movements of the device; as a result, this reduces the reliability of the device. Wireless networks are designed to support the same standard and the same protocol as wired network supports, but there are some differences between them. The advantage of using wireless is the flexibility because node can communicate without wires. On the other hand, it is more complex to configure than wired network, lower speed compared to wired network and affected by surrounding like: walls (blocking), microwave oven (interference), far distance (attenuation).

Bluetooth, Ultra-Wideband (UWB), ZigBee, and Wi-Fi are four protocol standards for short- range wireless communications with low power consumption. From an application point of view, Bluetooth is intended for a cordless mouse or keyboard, UWB is oriented to high-bandwidth multimedia links, ZigBee is designed for reliable wirelessly networked monitoring and control networks, while Wi-Fi is directed at computer-to-computer connections as an extension or substitution of cabled networks. The comparisons of different wireless technologies are summarized in Table 3.1 [21]. According to the previous applications of wireless technologies, the rehabilitation device can use ZigBee or Bluetooth technology. ZigBee technology is preferred for two reasons: First of all, ZigBee latency is 15 milliseconds, while Bluetooth is 3 seconds. The second reason is that, the ZigBee batteries can run for years, while Bluetooth has to be recharged as often as every day [21].

Table 3.1: Comparison of the wireless technologies [21].

Standard	**Bluetooth**	**UWB**	**ZigBee**	**Wi-Fi**
IEEE spec.	802.15.1	802.15.3a	802.15.4	802.11a/b/g
Frequency band	2.4 GHz	3.1-10.6 GHz	868/915 MHz; 2.4 GHz	2.4 GHz; 5 GHz
Max signal rate	1 Mb/s	110 Mb/s	250 Kb/s	54 Mb/s
Nominal range	10 m	10 m	10 - 100 m	100 m
Nominal TX power	0 - 10 dBm	-41.3 dBm	(-25) - 0 dBm	15 - 20 dBm
Power Consumption	40mA TX, standby 0.2mA	220mA TX, standby 25mA	30mA TX, standby 0.1mA.	200mA TX, standby 20mA
Data protection	16-bit CRC[*]	32-bit CRC	16-bit CRC	32-bit CRC

[*]CRC: Cyclic Redundancy Check

The available series of XBee are series 1 and series 2. Series 1 is preferred for the case of using only two XBee and it allows for point-point, mesh, and star network while series 2 allows only for mesh network. As a result, the required wireless modules are two identical XBee of series 1 as shown in Figure 3.7. The best choice is 1mW wire antenna- series 1 with a range of 30 m but the available in Hebron stores are XBee Pro 50mW wire antenna- series 1 with a range of 90m which are finally used.

Figure 3.7: The XBee 1mW wire antenna- series 1

3.5 Computer

In this section, the computer hardware and the programming language are explained.

3.5.1 Software (Programming Language):

There are many choices for the programming language that can be used for the implementation of the system. C, C++, C#, Java and Visual Basic (VB) are five of popular programming languages for desktop applications programming. While, C++ is the object oriented version of C, C# is a simplified version of C++. Moreover, C# is easy to use, and allows accessing more library routines than C and C++. It contains a special environment for games programming, which is Microsoft XNA Framework that is used to design 2D, and 3D games. It needs XNA Game Studio Toolkit integrated with Microsoft Visual Studio to create a computer game. As a result, C# is the programming language that is used to implement the software system. The comparison of five popular programming languages is shown in Table 3.2.

Table 3.2: Comparison of five popular programming languages

Criteria / Language	C	C++	Java	VB	C#
Object Oriented		✓	✓	✓	✓
.Net Technology				✓	✓
Highly Compatible with Oracle			✓		
Highly Compatible with Microsoft SQL				✓	✓
Advanced Graphics Options			✓	✓	✓
Special Environment for Computer Games					✓

C#; is the programming language that is used to implement the software system. It is simple, modern, general-purpose, strong type checking, object-oriented programming language, which is developed within Microsoft [20]. And it can be used for traditional programming to construct desktop applications, yet for games programming using the Microsoft XNA Framework. C# code which is written in XNA Environment can be easily used on Xbox and Windows. XNA is popular and designed especially for graphics and games programming.

Despite this, XNA is not a computer game programming engine, so the programmer must spend more time and write extra code to create a simple computer game [22].

3.5.2 Hardware:

A midrange computer is chosen with the specification shown in Table 2.3. In order to know the sampling frequency that this computer can deal with per second, a C# code is run on the computer. The results show that 10000 readings require 46501273 ticks and 19885 milliseconds. The chosen computer can deal with 503 readings per second approximately.

Table 3.3: The chosen computer specifications

Processor	Inter (R), Core (TM), i5-2430M CPU @ 2.40GHz (4CPUs)
Memory	2048 MB RAM
Graphics	Inter (R) HD Graphics Family (785 MB)
Windows	Microsoft Windows 7

3.6 Load System

As discussed earlier, the load system plays an important part in VRR. An increasing load should resist the patient's movement during sessions. It is controlled manually by the therapist himself based on the patient situation. In general, dampers have three types: passive, active and semi-active. Passive dampers have constant damping ratio which cannot be changed. These types of dampers are not preferred since they lack the variability needed to increase the difficult level which corresponds to the extra effort the patient should do to move the manipulator and improve weak muscles.

The second type is active dampers where external energy is added to the system through an actuator. Using active dampers increases the complexity and cost of the device. Further, they might cause a safety issue by applying excessive active forces on the patient’s arm.

In contrast to active control devices, semi-active control devices cannot inject mechanical energy into the system. Examples of such devices are dampers with controllable fluids and variable orifice damper. Controllable fluids belong to a family of fluids whose properties depend on the strength of a magnetic or electrical field respectively. The main characteristic of these fluids is their ability to change reversibly from free-flowing, linear viscous liquids, to semi-solids with the yield strength continuously controllable when exposed to either an electric or magnetic field. In the absence of an applied field, these fluids exhibit Newtonian-like behavior [23].

One of the available companies is FLUDICON which produce these types of linear dampers as shown in Figure 3.8. Unlike rotary MR brakes which still under researches only. Based on electro-rheological fluid technology, the damping ratio is electronically fully adjustable with a response time in milliseconds. Other advantages: It has no mechanical valves, low power consumption relative to other types of semi-active dampers and it can be controlled by analog or digital controllers or even via remote control [24].

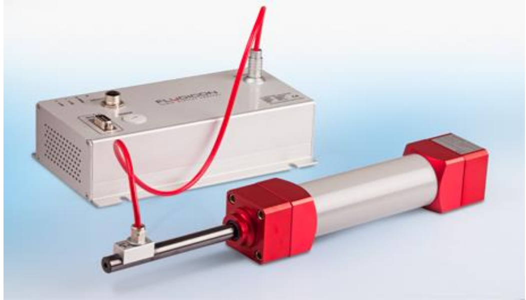


Figure 3.8: Set-up of a Fludicon system consisting of Rhe-Damp and an amplifier [24].

The disadvantages of this type are that: the amplifier runs on 24 V DC with a working range from 0 V to 4000 V which is dangerous, the weight of this system ranges from 2 – 11 kg for each system depending on the used model and finally, this device is expensive. Based on these disadvantages; this idea is canceled.

Variable orifice damper consists of a single rod piston with a variable orifice as shown in Figure 3.9. Two methods for controlling the orifice are proposed: manual by hand and automatic through a motor. The therapist recommended the manual control with the ability to change the load to a suitable range and to have a self-feeling of the load before engaging the patient [4].

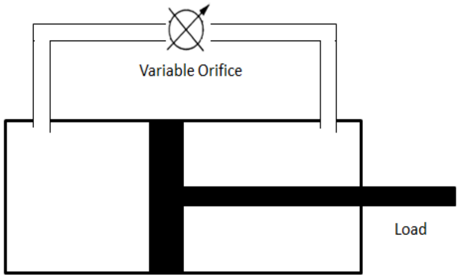


Figure 3.9: Schematic of variable-orifice damper

Figure 3.10 shows the location of the damper, where it is connected almost parallel to the diagonal of the parallelogram (D) with a small distance from the two joints at both ends of the damper. This is done to avoid extra loads on these two joints and to ensure that no friction happens due to both ends of the damper on these joints.

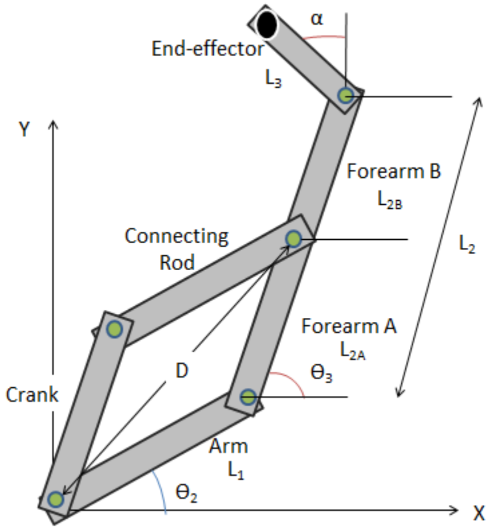


Figure 3.10: The mechanical arm in 2-D with the damper position

3.7 Budget

Table 3.4: Budget

Components	**Price (NIS)**	**Quantity**	**Total (NIS)**
Mechanical Device Components	3000	1	3000
Potentiometer (linear increment)	80	3	240
Controller (Arduino Uno)	250	1	250
ZigBee (XBee 50mW Chip Antenna – Series1)	350	2	700
XBee Explorer Dongle	150	1	150
Breakout Board	50	1	50
Software (MSQL, VS)	200	1	200
			4590

CHAPTER 4

Mechanical Design

4.1 Introduction

Design is an iterative process with many interactive phases. Many resources exist to support the designer, including many sources of information and an abundance of computational design tools [25]. Design is needed to guarantee if the deflection of the manipulator is acceptable or not, also the bearing design is needed (size, type, etc…) for the joints and the buckling calculation is needed to be sure that the device is not bucking if someone use it. Welding design determines the type of weld, electrode specification, size and the length of the weld pattern.

This chapter presents the manipulator design (including the bearings calculations), the design of the configuration joint, the buckling and the welding calculations.

4.2 Manipulator Design

This section is important to determine the maximum allowable force before the links failure which reflects the strength of the manipulator. It is also important to determine the maximum deflection at the end-effector and the bearings size for each joint.

Two design methodologies are used: the first one is the forward methodology or the normal methodology of design. The second one is the reverse design, which is used in this section based on the final dimensions for the links, then check if these dimensions are acceptable or not.

In this section, the reverse design methodology is used where the dimensions of links and the material selection are listed in Table 4.1 and the maximum allowable force before failure is calculated.

Table 4.1: The links dimensions and the used material for the strength calculations

Arm rod and connecting rod (L_1) [mm]	Length 550; tube Φ 25; thickness 5
Forearm, segment 1 (L_2)	Length 187; tube Φ 25; thickness 5
End effector link (L_3)	Length 283; tube Φ 25; thickness 5
Bending angle (α) [deg°]	36.5 °
θ_2 [deg°]	90.0 °
θ_3 [deg°]	20.0 °
Aluminum density	(2710 Kg/m^3)

Figure 4.1 shows the manipulator with the human-hand load (F) found by measuring the human hand weight. It is found in the range of (0.4 – 0.7) kg. The weight is considered 1.0 kg. Therefore, the force (F) is about 10 N.

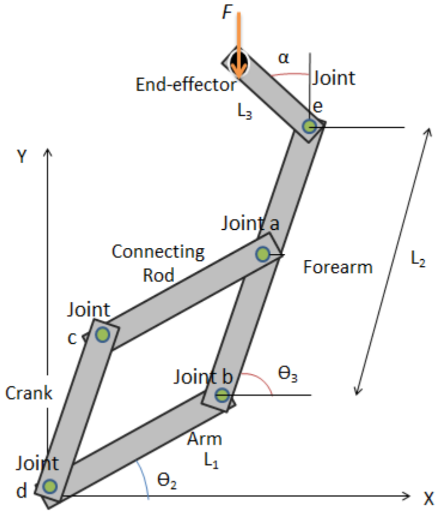


Figure 4.1: The manipulator with the force affected

4.2.1 Inertia of cross section

This section explains the calculations of inertia for each link about the x-axis. The circular cross-section of the links are as shown in Figure 4.2 with an outer diameter D = 25 mm and an inner diameter d = 15 mm. From the materials' properties tables, the yield strength (S_y) is 95 MPa, and the modulus of elasticity (E) is 70 GPa.

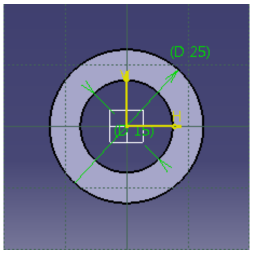


Figure 4.2: Circular cross-section for each link

The second moment of area (I) and the second polar moment of area (J) are needed to calculate the stress and deflection. Since the cross-section is same for all the links in the manipulator; the calculation for I and J will be included in this section only.

The second moment of area (I) is:

$$I = \frac{\pi}{64}\,(D^4 - d^4) \tag{4.1}$$

$$I = \frac{\pi}{64}\,(0.025^4 - 0.015^4)$$

$$\boldsymbol{I = 1.67 * 10^{-8}\ m^4}$$

The second polar moment of area (J) is:

$$J = \frac{\pi}{32}\,(D^4 - d^4) \tag{4.2}$$

$$J = \frac{\pi}{32}\,(0.025^4 - 0.015^4)$$

$$\boldsymbol{J = 3.33 * 10^{-8}\ m^4}$$

All the design calculations are based on a safety factor, the chosen safety factor (n_d) for these calculations is considered to be 3. The allowable strength (σ_{all}) is calculated in equation (4.3):

$$\sigma_{all} = \frac{Sy}{n_d} \tag{4.3}$$

$$= \frac{95}{3}\ \text{MPa}$$

$$\boldsymbol{\sigma_{all} = 31.67\ \text{MPa}}$$

4.2.2 Strength Calculations

4.2.2.1 Strength of the end-effector link

The end-effector link is considered as a cantilever beam as shown in Figure 4.3. The force (F) is applied on the link with a length of (L) and the resulting moment (M) is:

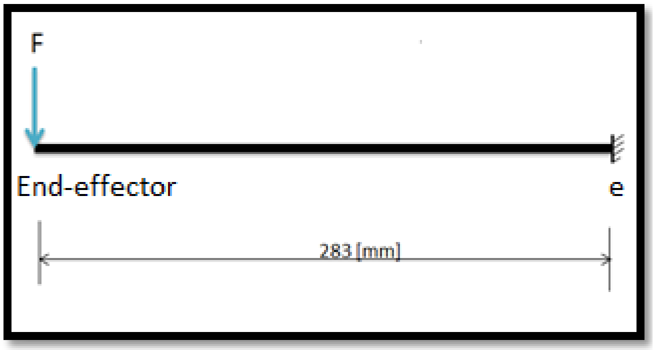


Figure 4.3: End-effector link with the affected load

$$M = F * L \tag{4.4}$$

$$M = F * 0.283$$

$$M = 0.283\, F$$

The bending stress is obtained from equation (4.5):

$$\sigma = \frac{\mathrm{Mc}}{\mathrm{I}} \tag{4.5}$$

Where;

M: The total moment on the beam.
c: The maximum distance from the neutral axis.
I: Second moment of area.

The bending stress on the end-effector link is obtained by the substitution of equations (4.1) and (4.4) in equation (4.5):

$$\sigma = \frac{0.283\ \mathrm{F}*0.025}{1.67*\ 10^{-8}}$$

$$\sigma = 0.212\ \mathrm{F}$$

But, the bending stress is replaced by the allowable stress in equation (4.3). Therefore, the maximum force F that can be applied on the end-effector link before failure is:

F= 150 N

4.2.2.2 Strength of the forearm link

The reaction force (F_r) is obtained from the free body diagram as shown in Figure (4.4), the relation between (F_r) and (F) is explained in equation (4.6).

$$F_r = 2 * F \tag{4.6}$$

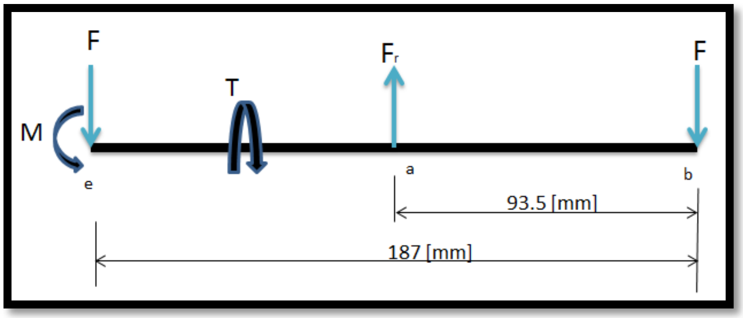


Figure 4.4: Free body diagram for the forearm link

The forearm link is considered to be a cantilever beam as shown in Figure 4.5, the force (F) is applied on the beam and the new resulted moment (M), the reaction (M_r) and torque (T) are calculated using equations (4.7), (4.8) and (4.9) respectively.

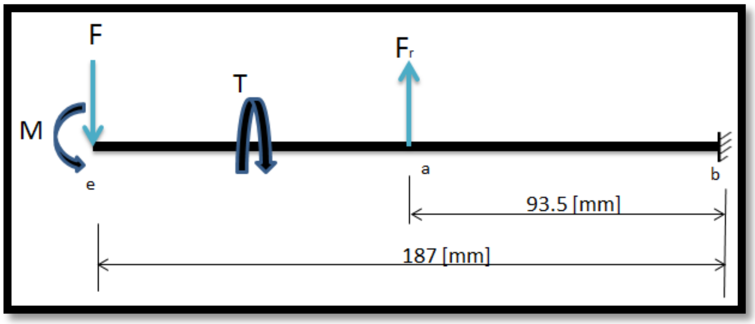


Figure 4.5: Forearm link with the affected load

$$M = F * (0.187 + 0.283 * \cos(36.5))$$

$$\boldsymbol{M = 0.415\ F} \qquad (4.7)$$

$$M_r = F_r * (0.0935)$$

$$= (2*F) * (0.0935)$$

$$\boldsymbol{M_r = -\ 0.187\ F} \qquad (4.8)$$

$$T = F * (0.283 * sin\ (36.5))$$

$$\boldsymbol{T = 0.168\ F} \qquad (4.9)$$

The total bending moment (M_T) is obtained by the summation of equation (4.7) and equation (4.8), as shown in equation (4.10).

$$\boldsymbol{M_T} = 0.228\ F \qquad (4.10)$$

The bending stress on the forearm link is obtained by the substitution of equations (4.1) and (4.10) in equation (4.5).

$$\sigma = 0.17\, F * 10^6 \tag{4.11}$$

The torsional stress is obtained from equation (4.12):

$$\tau = \frac{\mathrm{Tc}}{J} \tag{4.12}$$

Where;

T: The total torque on the beam.

c: The maximum distance from the neutral axis.

J: Second polar moment of area.

The torsional stress on the forearm link is obtained by the substitution of equations (4.2) and (4.9) in equation (4.12):

$$\tau = \frac{0.24\ \mathrm{F}*0.025}{3.33*\ 10^{-8}}$$

$$\tau = 0.063\ \mathrm{F} *10^6 \tag{4.13}$$

The two principle stresses is obtained by using equation (4.14).

$$\sigma_1,\ \sigma_2 = \frac{\sigma_x + \sigma_y}{2} \pm \sqrt{\left(\frac{\sigma_x + \sigma_y}{2}\right)^2 + \tau^2} \tag{4.14}$$

Here, the maximum principle stress is used and the bending stress exists only in one dimension, so that equation (4.14) is reduced to equation (4.15).

$$\sigma_1 = \frac{\sigma}{2} + \sqrt{\left(\frac{\sigma}{2}\right)^2 + \tau^2} \tag{4.15}$$

The maximum principle stress on the forearm link is obtained by the substitution of equations (4.11) and (4.13) in equation (4.15)

$$\sigma_1 = \frac{0.17\, F}{2} + \sqrt{\left(\frac{0.17\, F}{2}\right)^2 + 0.063^2}$$

$$\sigma_1 = 0.19\, F *10^6$$

But, the maximum principle stress is replaced to the allowable stress in equation (4.3). Therefore, the maximum force F that can be applied on the forearm link before failure is:

F = 167 N

4.2.2.3 Strength of the arm link

The arm link is considered as a cantilever beam as shown in Figure 4.6, the force (F) is applied on the beam of length (L) and that new resulted moment (M) and torque (T) are found using equations (4.16) and (4.17), respectively.

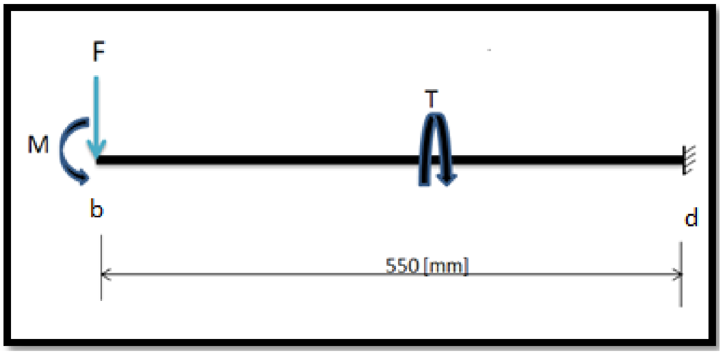


Figure 4.6: Arm link with the affected load

$M = F * (0.55 + 0.187 * \cos(20) + 0.283 * \cos(56.5))$

$$\mathbf{M = 0.88\ F} \tag{4.16}$$

$T = F * (0.283 * \sin(56.5) + 0.187 * \sin(20))$

$$\boldsymbol{T} \mathbf{= 0.3}\ \boldsymbol{F} \tag{4.17}$$

The bending stress on the arm link is obtained by the substitution of equations (4.1) and (4.16) in equation (4.5).

$$\sigma = 0.66\ F \tag{4.18}$$

The torsional stress on the arm link is obtained by the substitution of equations (4.2) and (4.17) in equation (4.12):

$$\tau = 0.113\ F \tag{4.19}$$

The maximum principle stress on the arm link is obtained by the substitution of equations (4.18) and (4.19) in equation (4.15)

$\sigma_1 = 0.68\ F$

But, the maximum principle stress is replaced to the allowable stress in equation (4.3). Therefore, the maximum force F that can be applied on arm link before failure is:

F = 47 N

4.2.2.4 Strength of the connecting rod

The connecting rod link is considered as a cantilever beam as shown in Figure 4.7, the force (F) is applied on the beam and the new resulted moment (M) and torque (T) are found using equations (4.20) and (4.21) respectively.

$$M = F * (0.55 + 0.0935 * cos\ (20) + 0.283 * cos\ (56.5))$$

$$\boldsymbol{M = 0.8\ F} \qquad (4.20)$$

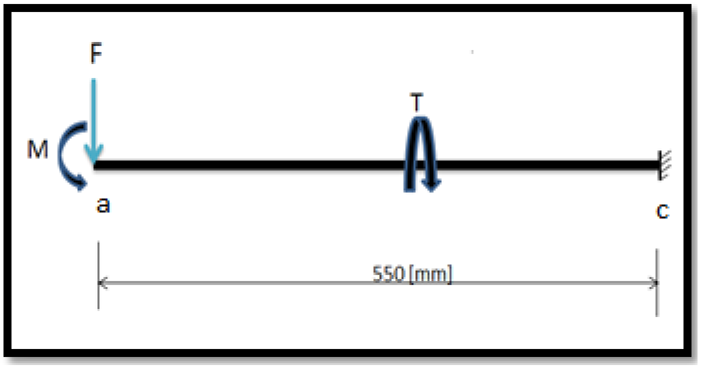


Figure 4.7: Connecting rod link with the affected load

$$T = F * (0.283 * sin\ (56.5) + 0.0935 * sin\ (20))$$

$$\boldsymbol{T = 0.27\ F} \qquad (4.21)$$

The bending stress on the connecting rod link is obtained by the substitution of equations (4.1) and (4.20) in equation (4.5).

$$\sigma = 0.6\ F \qquad (4.22)$$

The torsional stress on the connecting rod link is obtained by the substitution of equations (4.2) and (4.21) in equation (4.12).

$$\tau = 0.1\ F \qquad (4.23)$$

The maximum principle stress on the connecting rod link is obtained by the substitution of equations (4.22) and (4.23) in equation (4.15)

$$\sigma_1 = 0.62\ F$$

But, the maximum principle stress is replaced to the allowable stress in equation (4.3). Therefore, the maximum force F that can be applied on the connecting rod link before failure is:

F = 51 N

4.2.2.5 Strength of the crank link

The crank link is considered as a cantilever beam as shown in Figure 4.8. The force (F) is applied on the beam and with the same previous moment (M) and torque (T) found in equations (4.24) and (4.25), respectively.

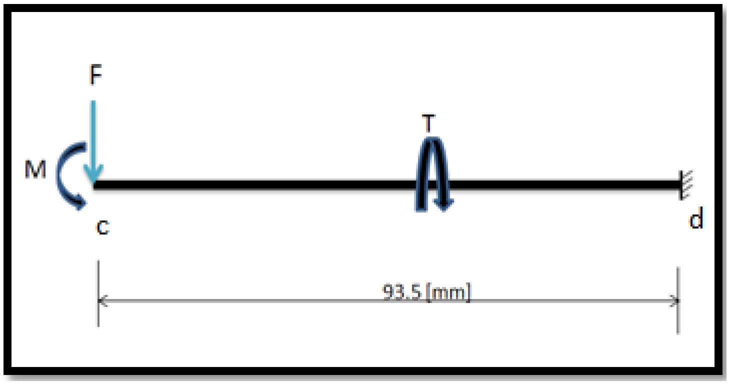


Figure 4.8: Crank link with the affected load

$$M = F * (0.55 + 0.187 * \cos(20) + 0.283 * \cos(56.5))$$

$$\boldsymbol{M = 0.88\ F} \tag{4.24}$$

$$T = F * (0.283 * \sin(56.5) + 0.187 * \sin(20))$$

$$\boldsymbol{T = 0.3\ F} \tag{4.25}$$

The bending stress on the crank link is obtained by the substitution of equations (4.1) and (4.24) in equation (4.5).

$$\sigma = 0.66\ F \tag{4.26}$$

The torsional stress on the crank link is obtained by the substitution of equations (4.2) and (4.25) in equation (4.15).

$$\tau = 0.113\ F \tag{4.27}$$

The maximum principle stress on the crank link is obtained by the substitution of equations (4.26) and (4.27) in equation (4.15)

$$\sigma_1 = 0.68\ F$$

But, the maximum principle stress is replaced to the allowable stress in equation (4.3). Therefore, the maximum force F that can be applied on the connecting rod link is:

$$F = 47\ \text{N}$$

4.2.2.6 Decision Taking for Strengths Analysis

The manipulator strengths that are found in the previous subsections are summarized in Table (4.2). This table shows the maximum allowable force for each link. The minimum allowable force in the manipulator's links before failure is 47 N (on the arm link and crank link).

Table 4.2: The maximum force allowable for each link

Link	Maximum force
End effector link	150 N
Forearm link	167 N
Arm link	47 N
Crank link	47 N
Connecting rod link	51 N

The maximum estimated force for the patient hand is 10 N which is less than 47 N. Therefore, the dimensions for the links and the material choices are acceptable under the stress test.

4.2.3 Deflection Calculations

The patient's hand affects with 10 N at the end-effector. This section calculates the deflection on the end-effector. This deflection is found by the summation of the deflection on the arm link, the forearm and the end effector link. While these calculations can be performed by hand; CATIA software is used instead for its simplicity in deflection calculations.

4.2.3.1 Arm Link Deflection

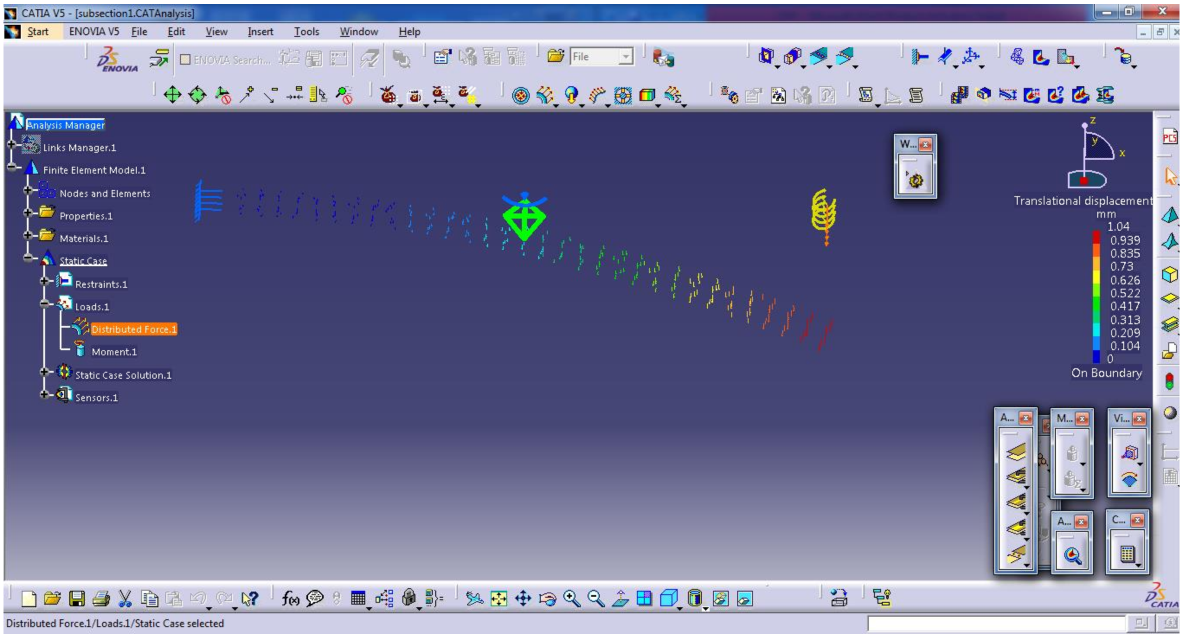
Figure 4.6 shows the arm link deflection with 10 N force (F), 8.8 N.m moment (M) and 3 N.m torque (T). Figure 4.9 shows the different deflection for each position on the arm link using CATIA software. The maximum deflection on this link is:

$$y = 1.04 \text{ mm} \tag{4.28}$$

4.2.3.2 Forearm link deflection

Figure 4.5 shows the forearm link deflection with 10 N force (F), 4.15 N.m moment (M) and 1.68 N.m torque (T). The different deflection for each position on the forearm link appears as shown in Figure 4.10 using CATIA software. The maximum deflection on this link is:

$$\hat{y} = 0.0468 \text{ mm} \tag{4.29}$$



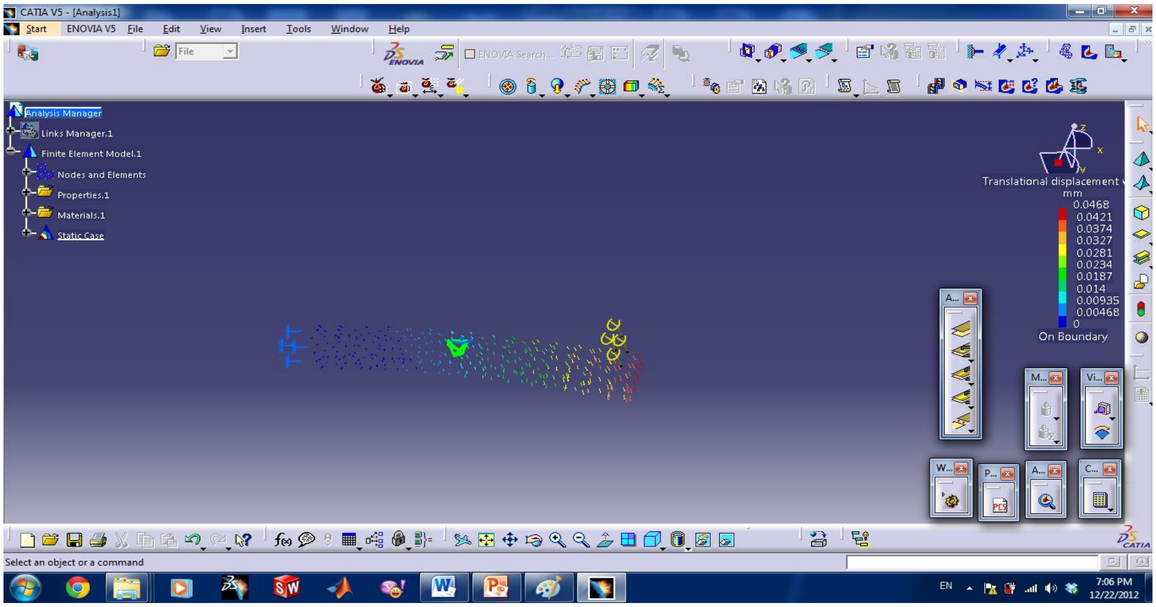
Distributed Force.1/Loads.1/Static Case selected

Figure 4.9: Deflection in the arm link

But, the total deflection on this link equals the maximum deflection found in equation (4.29) and the deflection found in equation (4.28).

$$y = 0.0468 + 1.04$$

$$y = 1.0868 \text{ mm} \tag{4.30}$$



Select an object or a command

Figure 4.10: Deflection in the forearm link

4.2.3.3 End-Effector Deflection

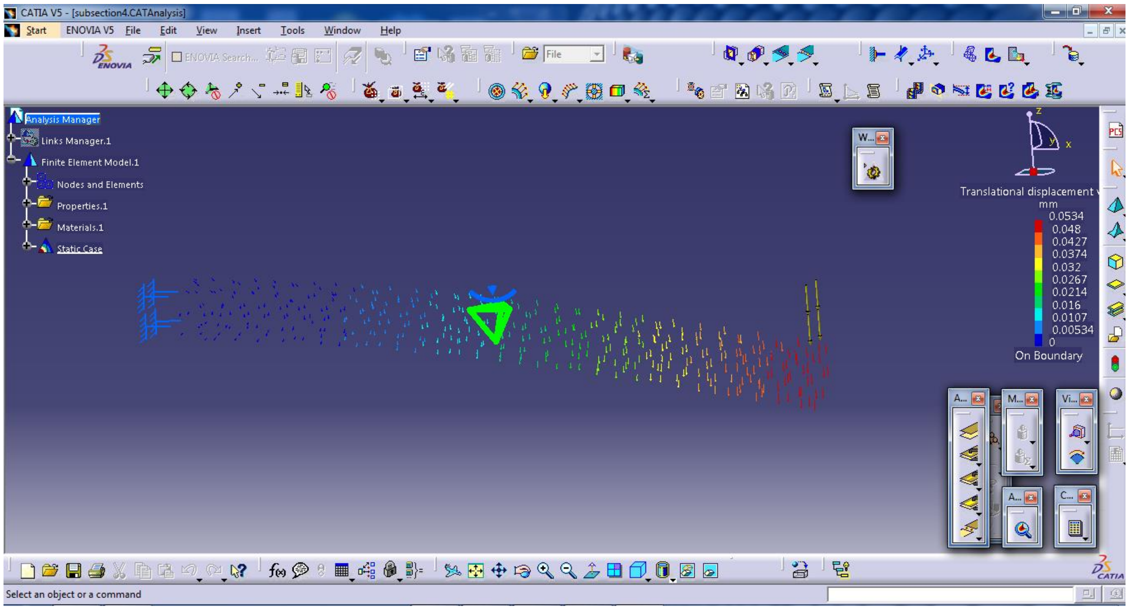
Figure 4.3 shows the end-effector link deflection with 10 N force (F) only. The different deflection for each point on the link appears as shown in Figure 4.11 by using CATIA software, the maximum deflection on the link is:

$$\acute{y} = 0.0534 \text{ mm} \tag{4.31}$$

But, the total deflection on this link equals to the maximum deflection in equation (4.30) and the deflection in equation (4.31).

$$y = 0.0534 + 1.0868$$

$$y = 1.14 \text{ mm} \tag{4.32}$$



Select an object or a command

Figure 4.11: Deflection on the end-effector link

4.2.3.4 Decision Taking for the Deflection Analysis

Equation (4.32) is shown that the maximum deflection corresponding to an applied force of 10 [N] is 1.14 mm which is acceptable for the operation of the device.

4.2.4 Bearing Design

The importance of bearing stems from the friction associated with the relative motion performed under load. This load may be radial, thrust or combinational between them. In this project, the load is a combinational load (radial and thrust).

The calculations of bearing are based on the joint with the maximum reaction force. The other bearings are assumed to have the same size and application. The joint with the maximum force reaction could be at joint “a or b” for the forearm link as shown in Figure 4.12.

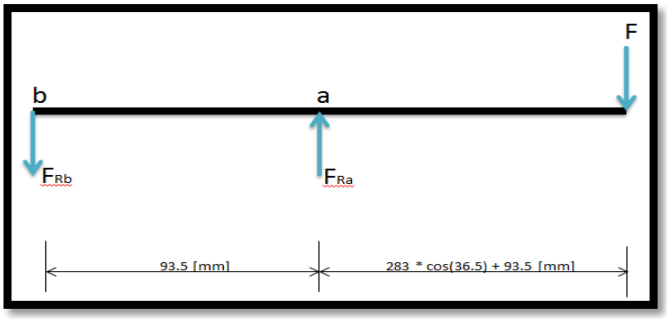


Figure 4.12: Action and reaction forces on joints ‘a’ and ‘b’

$$\Sigma\ F_y = 0.0$$

$$F_{Ra} = F + F_{Rb} \qquad (4.33)$$

$$\Sigma\ M_a = 0.0$$

$$(93.5 + 283 * cos\,36.5) * F\ = 93.5 * F_{Rb}$$

$$F_{Rb}\ = 3.4\ F \qquad (4.34)$$

By substitution of equation (4.34) in equation (4.33), then

$$F_{Ra}\ = 4.4\ F \qquad (4.35)$$

Substituting the force (F) in equations (4.34) and (4.35) with the maximum force to get:

$$F_{Rb}\ = 34\ \mathrm{N}$$

$$F_{Ra}\ = 44\ \mathrm{N}$$

From these results; the maximum reaction force is concentrated on joint ‘a’ with 44 N. This force is considered as a thrust load F_a on the bearing. The maximum radial force F_r is the force produced from the damper. This force is assumed to be 5 N. The bearing design is based on joint ‘a’; because it has the maximum thrust load. The size of bearing ‘a’ is used for all other joints.

The desired life ($\mathrm{L_D}$) is:

$$\mathrm{L_D}\ = \mathrm{L_{Dh}} * \mathrm{n_D} * 60 \qquad (4.36)$$

From the bearing design tables $\mathrm{L_{Dh}}$ is 2000 hours, the desired speed $\mathrm{n_D}$ (rev/min) is obtained from equation (4.37).

$$\mathrm{n_D} = \frac{60*\omega}{2\pi} \tag{4.37}$$

Human hand velocity is 0.8 (m/s) [26], the angular speed ω (rad/s) is 0.91. Substituting ω in equation (4.37), the desired speed ($\mathrm{n_D}$) is 9 (rpm). Substituting $\mathrm{n_D}$ and $\mathrm{L_{Dh}}$ in equation (4.36) to get the desired life with $1.08 * 10^6$ revolutions. The load is not variable (i.e. the load factor a_f is 1.0), and the inner ring is rotating (i.e. rotation factor $V$is 1.0). The design load on bearing (F_e) is obtained by substituting a_f, V and $\mathrm{F_r}$ in equation (4.38).

$$F_e = a_f * V * F_r \tag{4.38}$$

$$F_e = 5 \text{ N}$$

Ball bearing is used (i.e. factor a is 3.0), based on 90% reliability, the basic dynamic load ($\mathrm{C_{10}}$) is 5.15 N obtained by substituting a, $\mathrm{L_D}$ and F_e in equation (4.39) for 1 million revolution life ($\mathrm{L_{10}}$).

$$\mathrm{C_{10}} = \left(\frac{L_D}{L_{10}}\right)^{1/a} * F_e \tag{4.39}$$

According to ($\mathrm{C_{10}}$) result, the standard value is obtained from ball bearing design tables. The angular-contact ball bearing is used for the joints of the manipulator; because it deals with thrust loads. Therefore, the standard dynamic load ($\mathrm{C_{10}}^*$) is 4.94 KN, the standard static load ($\mathrm{C_0}^*$) is 2.12 KN. Using ($\mathrm{C_0}^*$) in the design tables; the constant *e* is 0.014, the constant X_2 is 0.56 and the constant Y_2 is 2.3

$$F_e = X_2 V F_r + Y_2 F_a \tag{4.40}$$

Using equation (4.40) a new F_e appears by substituting the new constants X_2 and Y_2, V and the forces F_r and F_a.

$$F_e = (0.56 * 1.0 * 5) + (2.3 * 44)$$

$$= 104 \text{ [N]}$$

The new basic dynamic load ($\mathrm{C_{10}}$) is found by substituting the new F_e in equation (4.39) with 26.5 N. Since this value is less than the standard dynamic load (4.94 KN), the values of standard and static loads are used. Four rolling-contact bearing are used with specify **a 02-10 mm angular-contact ball bearing**.

4.3 Configuration Joint Calculations

4.3.1 Design of the axial Shaft

The configuration of the device (horizontal, vertical and inclined) depends on the configuration motion presented by the axial shaft as shown in Figure (4.13). This rod rotates about fixed axis.

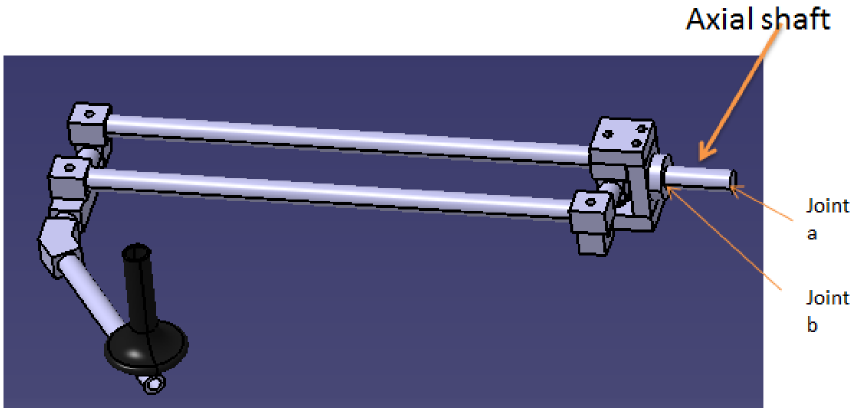


Figure 4.13: Axial shaft

The length of the axial shaft (l) is 75 [mm] and the material is Chroma (7190 Kg/m^3). The strength analysis is used to get the shaft dimension (D_i). Figure 4.14 shows the axial shaft where the bending moment (M) is 8.8 N.m, the torque (T) is 3 N.m and the force (F) is 10 N.

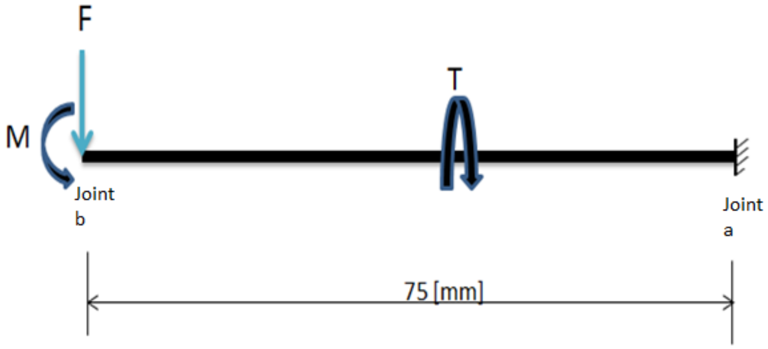


Figure 4.14: Shaft configuration with the affected loads

The factor of safety (n_d) is 3.0, from the materials' properties tables, the yield strength (Sy) is 360 MPa, and the modulus of elasticity (E) is 70 GPa. So, the allowable strength (σ_{all}) is calculated in equation (4.3):

$$\sigma_{all} = \frac{360}{3} \text{ MPa}$$

$$\boldsymbol{\sigma_{all} = 120.0 \text{ MPa}}$$

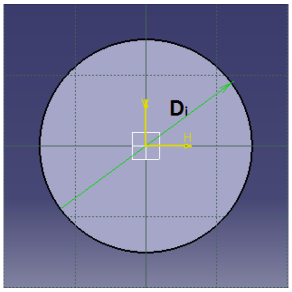


Figure 4.15: Circular cross-section for shaft configuration

The second moment of area (I) is:

$$I= \frac{\pi}{64} D_i^4 \tag{4.41}$$

The second polar moment of area (*J*) is:

$$J= \frac{\pi}{32} D_i^4 \tag{4.42}$$

By substituting equation (4.41), the bending moment (M) and the maximum distance from the neutral axis (c) with d/2, in equation (4.5) the bending stress is:

$$\sigma = \frac{89.681}{D_i^3} \tag{4.43}$$

By substituting equation (4.42), the torque (T) and the neutral axis (c), in equation (4.12) the torsional stress is:

$$\tau = \frac{15.29}{D_i^3} \tag{4.44}$$

The maximum principle stress on the shaft configuration is obtained by the substitution of equations (4.43) and (4.44) in equation (4.15):

$$\sigma_{\max}= \frac{89.681}{D_i^3} + \sqrt{\left(\frac{89.681}{D_i^3}\right)^2 + \left(\frac{15.29}{D_i^3}\right)^2} \tag{4.45}$$

Replace the maximum stress ($\sigma_{\max}$) in equation (4.45) with allowable stress (σ_{all}), therefore the diameter of the shaft configuration is:

$$D_i = 11.5 \text{ mm}$$

But, the diameter (D_i) that is used for the axial shaft is 25 [mm].

4.3.2 Geometrical design for the hole

This section is needed to determine the hole diameter (D_o) in the configuration joint as shown in Figure 4.16. The clearance between the shaft and the hole is needed to produce the horizontal, vertical and inclined situations. Therefore, the diameter of the hole (D_o) is assumed be 26 mm.

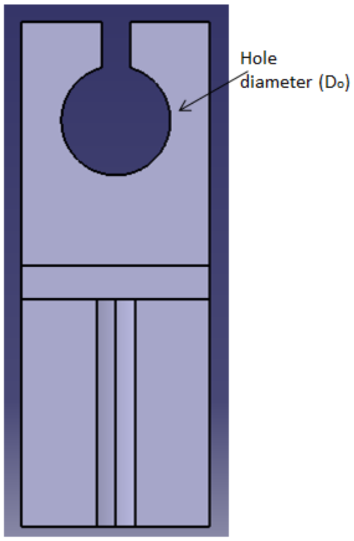


Figure 4.16: The cross section of the configuration angle

4.4 The Buckling Calculations

4.4.1 Design of the vertical shaft

The buckling analysis in the rod shown in the left panel of Figure 4.17 (that determines the length of the device) which is needed to know if the length (l) of the rod and the diameter of the shaft (D_i) is acceptable or not. The middle panel of Figure 4.17 shows the considered forces in the buckling calculations. While the bending moment (M) is 8.8 N.m, the torque (T) is 3 N.m and the force (F) is 10 N.

The middle panel in Figure 4.17 shows the effect of moment (M) and the axial load (F) in the (X-Z) plane and the effect of torque (T) and the axial load (F) in the (Y-Z) plane. Since the moment (M) is greater than (T); the moment (M) is used in the calculations. The moment (M) is represented as an eccentrically load as shown in in the right panel of Figure 4.17. The eccentric distance (e) is:

$$M = F * e$$

$$8.8 = 10 * e$$

$$e = 0.88 \text{ m}$$

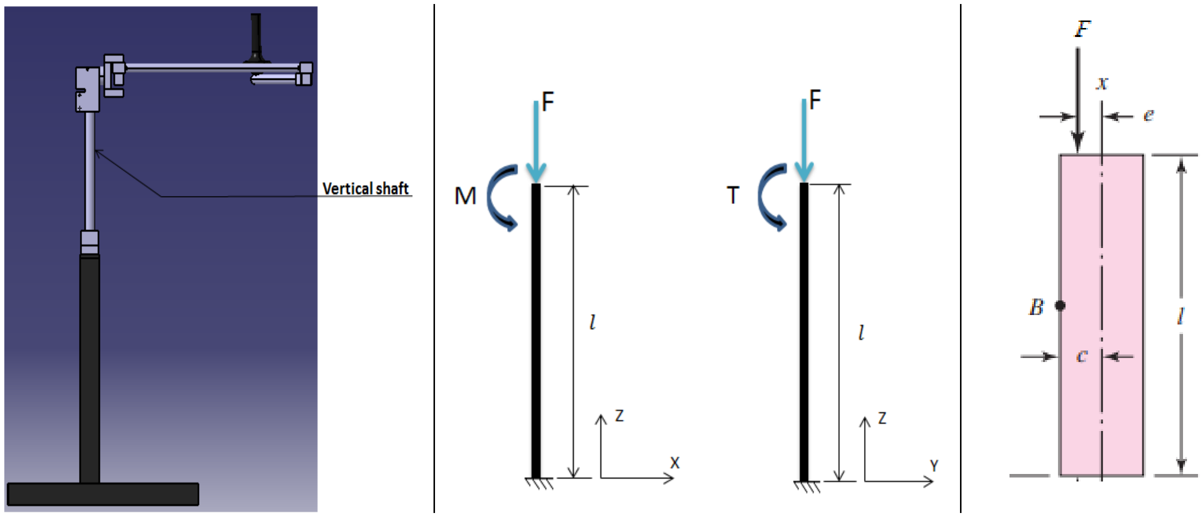


Figure 4.17: Left panel: The shaft buckling. Middle panel: Forces considered in buckling calculations in the two planes. Right panel: Eccentric Force.

The cross section is circular with the shaft diameter (D_i). The second moment of area (I) is obtained using equation (4.41). The radius of gyration (k) is obtained using equation (4.46), the critical strength (σ_{cr}) is obtained using equation (4.47), where the effective length (l_e) is $2l$. The cross section area (A) is obtained from equation (4.48).

$$k = \sqrt[2]{\frac{I}{A}} \tag{4.46}$$

$$\sigma_{cr} = \frac{P}{A}\left(1 + \frac{ec}{k^2}\right) \tag{4.47}$$

$$\mathrm{A} = \frac{\pi}{4} {D_i}^2 \tag{4.48}$$

Replacing the value of (I) and the value of (A) in equation (4.46)

$$k = \frac{D_i}{4} \tag{4.49}$$

The critical stress is obtained by substituting equation (4.48) and equation (4.49) in equation (4.47):

$$\sigma_{cr} = \frac{90.5 + 12.74 D_i}{{D_i}^3} \tag{4.50}$$

The yield strength for the Chroma shaft buckling (S_y) is 360 MPa and young's modulus (E) is 70 GPa. The critical strength equals the yield strength. Therefore, setting the value of σ_{cr} in equation (4.50) to get the value of the (D_i).

D_i= 6.3 mm

Let the factor of safety (n_d) to be 3.0

D_i= 9.1 mm

The limiting slenderness ratio obtained by using equation (4.51).

$$\left(\frac{L}{k}\right)= 0.282\left(\frac{AE}{F}\right)^{1/2} \tag{4.51}$$

By substituting (A), (E) and force (F) in equation (4.51), the limiting slenderness ratio is

$$\left(\frac{L}{k}\right)= 190 \tag{4.52}$$

Substituted (D_i) with 9.1 mm in equation (4.49) to get the value of (k) which is 2.275 mm. The total length of the buckling shaft, by substituting (k) in equation (4.52) is:

L= 432.25 [mm]

The total length (L) used in device is 400 mm. Also the diameter of the shaft (D_i) is 30 [mm] to make device more safe. The deflection calculation is needed to check if these values are accepted or not. Figure 4.18 shows the maximum deflection (y_{max}). The value of (y_{max}) is obtained by substituting the value of (e), (F), (L), (E) and the (I) in equation (4.53).

$$y_{max}= e\left[\sec\left(\sqrt[2]{\frac{F}{EI}}\frac{L}{2}\right)-1\right] \tag{4.53}$$

$y_{max}=$ 19.3 10^{-6} [mm]

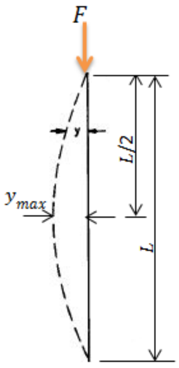


Figure 4.18: The maximum deflection on the buckling shaft

Because the maximum (y_{max}) is very small, the dimension of the buckling design is accepted with the diameter (D_i) 30 mm and the total length (L) 400 mm.

4.4.2 Geometrical Design for the Hole

This section is needed to determine the hole diameter (D_o) for the connection joint as shown in the right side of Figure 4.19 shows a hole with the diameter (D_o) in the joint that has a hole of the buckling shaft. The clearance between the shaft and hole is needed to produce different heights. Therefore, the diameter of hole (D_o) is 31 mm.

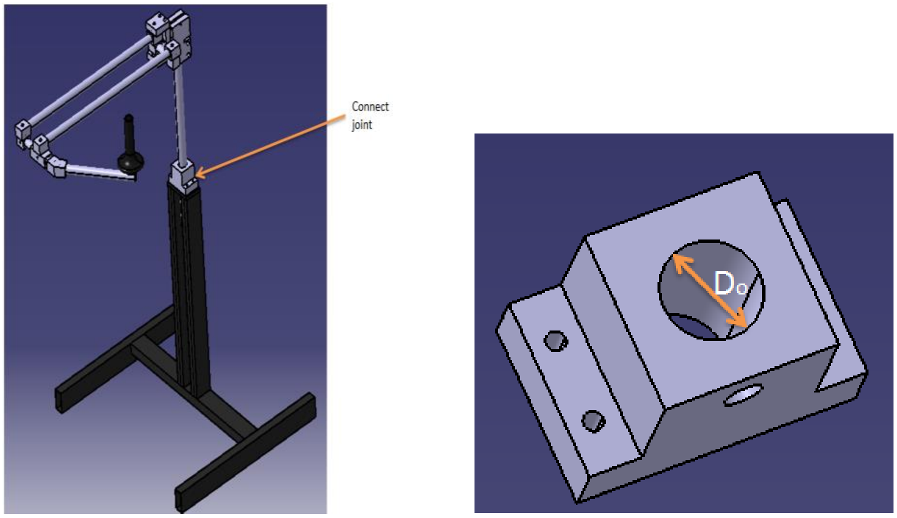


Figure 4.19: Hole of the vertical shaft; left side shows the position of the Connected joint. Right side: The hole diameter.

4.5 The welding design

The welding is needed to connect the base beams. Figure 4.20 shows the welding joints. Filler metal rods and wires are designated by a system established by the American Welding Society (AWS). AWS designation system is E60XX.

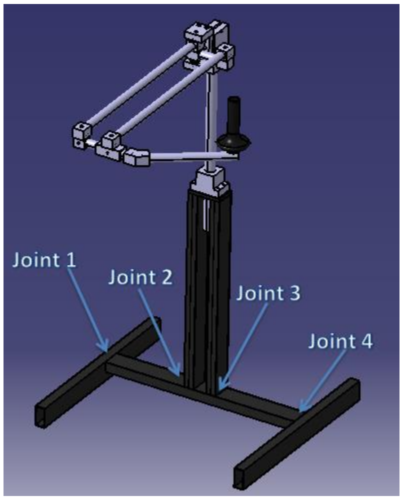


Figure 4.20: The welding joints

Welding for Joints 2 and 3:

Figure 4.21 shows the welding lines needed to connect the steel bar stand and the base bar for the joints 2 and 3.

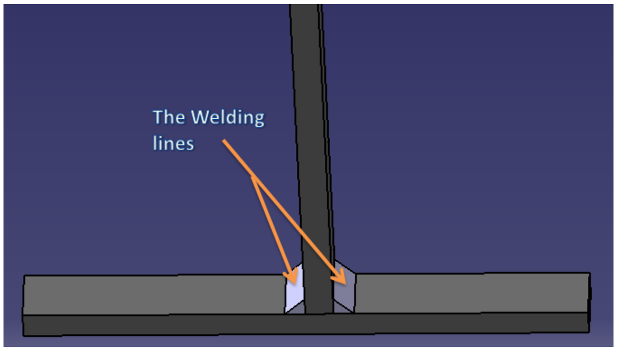


Figure 4.21: Connection between stand and the base for the joints 2 and 3

The stress on joints 2 and 3 is the same (torsion due to torque (T) with 3 N.m and bending due to moment (M) with 8.8 N.m). The left side in Figure 4.22 represent the primary shear (τ`) that is produced from the torsion, and the right side in the same figure represent the secondary shear ($\tau^{``}$) where is that from the bending moment. The distance (d) is 60 mm and distance (b) is 30 mm.

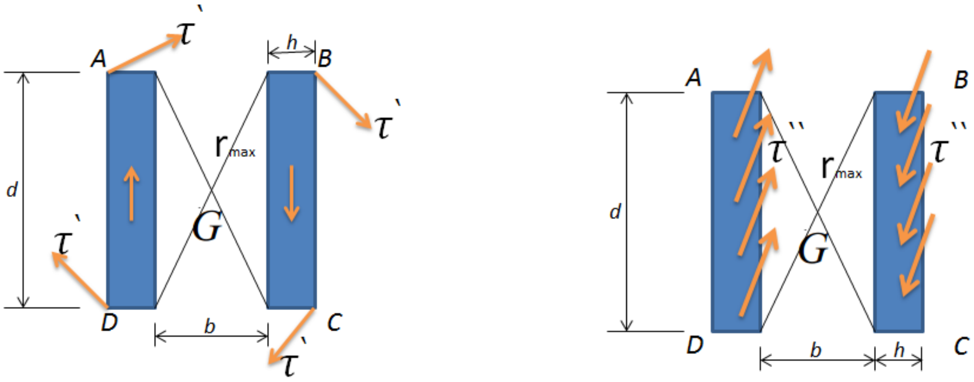


Figure 4.22: Pattern welding for joints 2 and 3: Left panel is for the torsion. Right panel with the bending effect

The primary shear is:

$$\tau^{`} = \frac{T\, r_{max}}{J} \qquad (4.54)$$

Where,

r_{max}: The maximum distance from the center (G) to the point of interest.

J: The second polar moment of area of the weld group about (G) based on the throat area of the weld.

The polar moment of area of the weld group per throat length (J_u) is selected from the weld design tables. This value needs to represent the value of (J) in the equation (4.55).

$$J = 0.707\ h\ J_u \tag{4.55}$$

The leg weld (h) needs to be designed. Replacing equation (4.54) with (4.56) by substituting the value of (J)

$$\tau^{`} = 2.259 * 10^6\ h \tag{4.56}$$

The secondary shear is:

$$\tau^{``} = \frac{M\ y}{I} \tag{4.57}$$

Where,

y: Distance from the center (G) to the point of interest.

I: The second moment of area of the weld group about neutral axis which pass through (G) based on throat.

The second moment of area of the weld group per throat length (I_u) selected from the weld design tables. This value needed to represent the value of (I) in the equation (4.58).

$$I = 0.707\ h\ I_u \tag{4.58}$$

The width of the weld lines (h) needs to be designed. Replacing equation (4.57) to equation (4.59) by replaced the value of (I)

$$\tau^{``} = 5.186 * 10^6\ h \tag{4.59}$$

The maximum shear stress here is founded using equation (4.60).

$$\tau_{max} = \sqrt[2]{(\tau^{`})^2 + (\tau^{``})^2} \tag{4.60}$$

Substituting equations (4.56) and (4.59) in equation (4.60) to get:

$$\tau_{max} = 5.66\ h \tag{4.61}$$

The yield strength (S_y) of the beams is 360 MPa. So the allowable shear stress is:

$$\tau_{all,beam} = 0.4 * S_y = 144 \text{ MPa}$$

The ultimate strength (S_{ut}) for the electrode is 413 MPa, therefore the allowable shear stress is:

$$\tau_{all,attachment} = 0.3 * S_{ut} = 124 \text{ MPa}$$

Now, for the entire joint:

$$\tau_{all} = \min(144{,}124) = 124 \text{ MPa}$$

Setting $\tau_{max} = \tau_{all}$, the value of the leg length (h) is considered 25 mm with two lines where the total length for each line is 60 mm.

Welding for joints 1 and 4:

The bending moment (M), torsion (T) and the bending produced from the force (F) are affected joints 1 and 4 shown in Figure 4.20. The bending (M) which equals 8.8 N.m and the bending from (F) are represented as the total bending (M_T) as shown in equation (4.62). The torque (T) is 3 N.m. The distance (s) from the joint to the force line of action is 0.3 m. Figure 4.23 shows the welding lines needed to connect the beams for joints 1 and 4.

$$M_T = M + F * s \tag{4.62}$$
$$M_T = 11.8 \text{ N.m}$$

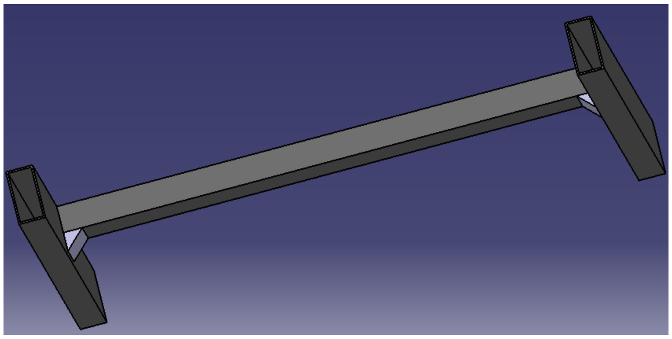
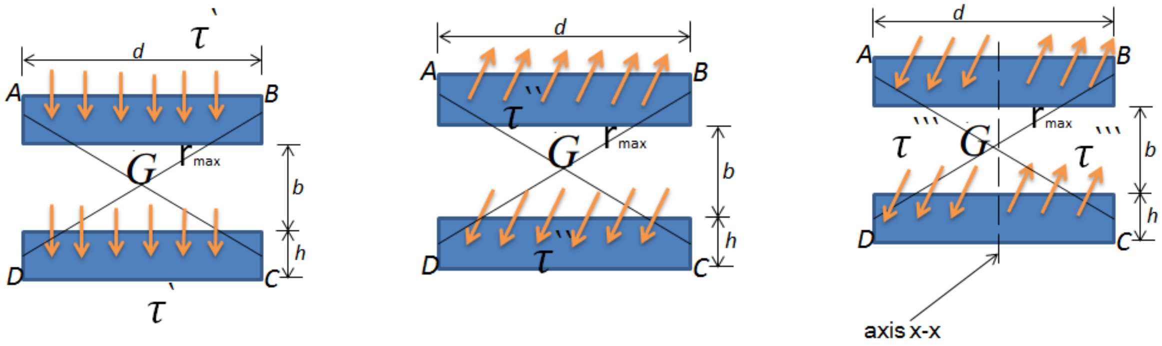


Figure 4.23: Joints 1 and 4 connection

Figure 4.24 shows the weld pattern for the joints 1 and 4 with the load affects. The distance (d) is 60 mm, the distance (b) is 30 mm and the weld leg (h) must be designed.



a. Direct shear **b.** Shear from bending **c.** Shear from torque

Figure 4.24: Welding Pattern for joints 2 and 3: a. Effect of direct force, b. Effect of moment (M) and c. Effect of torque (T)

The force (F) affects with direct shear as shown Figure 4.24.a. While the shear from the total moment (M_T) is show in Figure 4.24.b and the torque (T) with the shear about the axis (x-x) is shown in Figure 4.24.c. The maximum shear is on the corner so, the calculations are based on the maximum distance (r_{max}) from the center (G). The primary shear ($\tau^{`}$) is calculated using equation (4.63).

$$\tau^{`} = \frac{F}{1.414\, h\, b}$$

$$\tau^{`} = \frac{236}{h} \tag{4.63}$$

The secondary shear ($\tau^{``}$) is evaluated using equation (4.57) and the second moment of area of the weld group per throat length (I_u) is selected from the weld design tables. This value is needed to represent the value of (I) in the equation (4.58). The width of the weld lines (h) is need to complete the design. Represent equation (4.57) into equation (4.64) by replacing the value of (I).

$$\tau^{``} = 9.272 * 10^6\, h \tag{4.64}$$

The third shear ($\tau^{```}$) is represented in equation (4.65):

$$\tau^{```} = 3.536 * 10^6\, h \tag{4.65}$$

The maximum shear stress (τ_{max}) is represented in equation (4.66). The yield strength (S_y) of the beams is 360 MPa.

$$\tau_{max} = \frac{\sqrt[2]{1+1.768*10^9 h^2}}{h} \tag{4.66}$$

The allowable shear stress in the member is:

$\tau_{all,member}$= 0.4 * S_y= 144 MPa

The ultimate strength (S_{ut}) for the weld rod is 413 MPa, therefore the allowable shear stress is:

$\tau_{all,attachment}$= 0.3 * S_{ut}= 124 MPa

The design is based on the minimal allowable shear stress between the beam and the weld rod, therefore the considered allowable strength is:

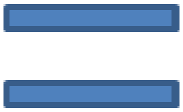
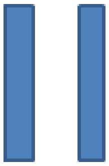
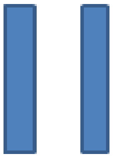
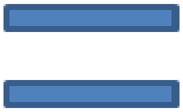
τ_{all}= min (144,124) =124 MPa

Setting τ_{max}= τ_{all}, therefore the value of h is considered 20 mm.

Welding Summary:

Table 4.3 shows the result of the welding design for each joint (1, 2, 3 and 4):

Table 4.3: Design results

Joint	Pattern	Electrode	Type	Size (h)	Total length
1		E60XX	Fillet Weld	20 [mm]	60 [mm]
2		E60XX	Fillet Weld	25 [mm]	60 [mm]
3		E60XX	Fillet Weld	25 [mm]	60 [mm]
4		E60XX	Fillet Weld	20 [mm]	60 [mm]

CHAPTER 5

Position Mapping

5.1 Introduction

This chapter aims at providing the concept of position mapping with the performed calculations. Mapping the position of the end-effector on the computer screen starts with the realization of the mechanical manipulator's links and joints. The next step is to convert the joints' angles into X & Y coordination using forward kinematics. Finally, the manipulator workspace must be mapped into the computer screen workspace to allow a realistic interaction between the patient while moving the end-effector and the different computer games.

The process starts when the patient moves the end-effector of the mechanical device with his/her arm. During this motion in the workspace, position sensors detect this motion and translate the change in position into electrical signals. Three potentiometers are used (two for the end-effector position in the plane and the third for the configuration). These sensors translate the change in position into analog electrical signals.

The potentiometers are connected to ARDUINO which receives these input analog signals, converts them into digital signals using the Arduino built-in analog to digital converter and sends them to ZigBee serially. This ZigBee sends the data to the other ZigBee which is connected to the computer. The computer processes with these signals now as they represent the position coordination in the game that is displayed on the screen.

5.2 Overall Features of the Manipulator

This system has three degrees of freedom due to its three revolute joints. Two of them are for the motion of the end-effector in the plane and the third is for the configuration of the device (horizontal, vertical or inclined). The system is considered of two degrees of freedom after choosing the appropriate configuration which indicates the suitable exercise for the patient. The mechanical arm in the two dimensions and the workspace are shown in Figure 5.1. The dimensions of the links are listed in Table 5.1. The crank and the connecting rod have no effect on the position of the end-effector which can be excluded from the analysis.

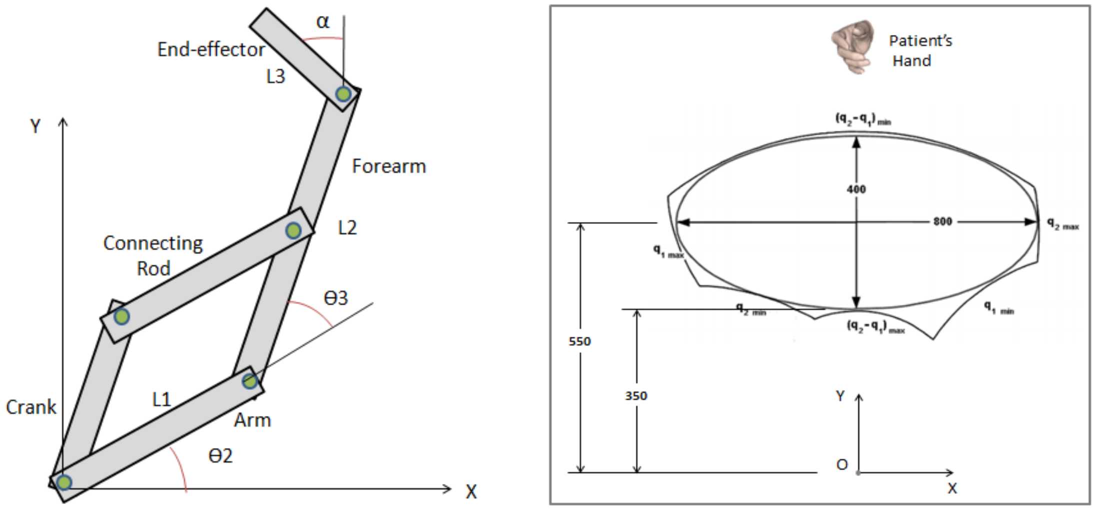


Figure 5.1: Left panel: The mechanical arm in 2D. Right panel: Workspace

Table 5.1: Overall features of the designed mechanical arm

Workspace (elliptical) [cm]	80 x 40
Arm rod and connecting rod (L_1) [mm]	Length 550
Forearm, segment 1 (L_2) [mm]	Length 187
Forearm, segment 2 (L_3) [mm]	Length 283
Bending angle (α) [deg°]	36.5°

5.3 Forward Kinematics

The forward kinematic equations define a function between the cartesian positions and the joint positions. The position of the manipulator's end-effector in the XY plane is found using equations (5.1) and (5.2). the left panel of Figure 5.2 describes the three revolute joints and the right panel of the same shows the left view of the manipulator in the horizontal plane (XY) with $\Theta_1 = 0°$.

$$X = L_1.\cos\Theta_2 + L_2.\cos(\Theta_2 + \Theta_3) + L_3.\cos(\Theta_2 + \Theta_3 + \alpha) \tag{5.1}$$

$$Y = L_1.\sin\Theta_2 + L_2.\sin(\Theta_2 + \Theta_3) + L_3.\sin(\Theta_2 + \Theta_3 + \alpha) \tag{5.2}$$

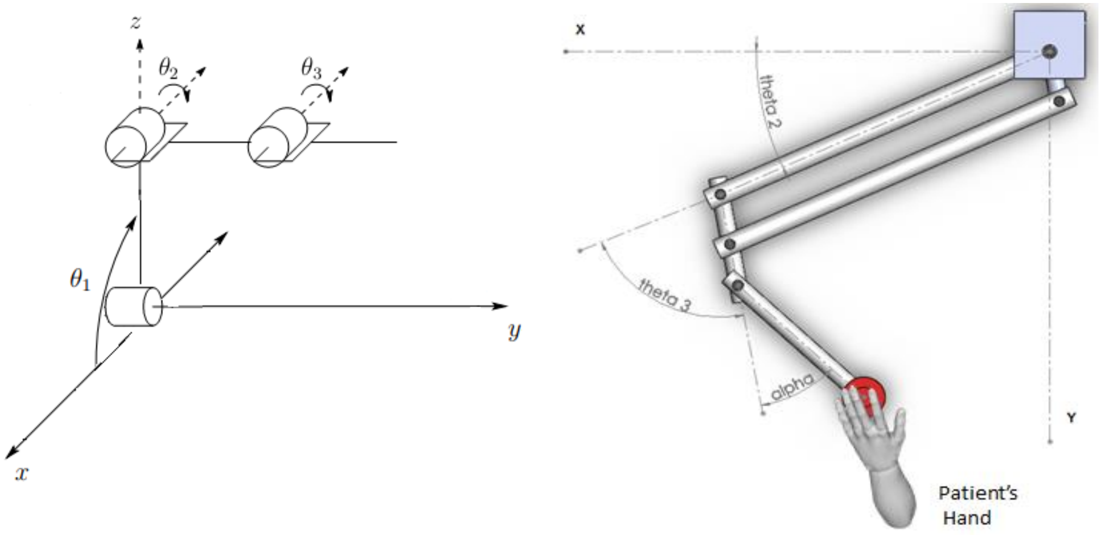


Figure 5.2: Left panel: The manipulator in the vertical configuration ($\theta_1 = 90^{\circ}$). Right panel: Top view in the horizontal plane ($\theta_1 = 0^{\circ}$).

5.4 Interfacing

Three potentiometers are used (two for the position and the third for the configuration). These sensors translate the change in position into analog electrical signals. The potentiometers are connected to ARDUINO microcontroller which receives the analog signals. The input analog signal is a varying voltage between 0-5V. ARDUINO converts any analog voltage between 0-5V to a digital number ranging from 0-1023 using the built-in analog to digital converter.

The three angles θ_1, θ_2 and θ_3 are measured practically for the device at different positions and the digital number P corresponding to each measurement is recorded in tables. For example, the angle θ_3 is measured and listed in table 5.2 with the corresponding digital number P_3. These numbers are read from the interfacing ARDUINO software. The next step is to plot these values using Matlab and find the equation by performing linear fitting. Figure 5.3 shows the data (blue line) and the linear fitting (red line).

Table 5.2: Digital values (P_3) correspond to each angle (θ_3)

θ_3	20	30	40	50	60	70	80	90	100	110	120	130	140	150
P_3	161	207	254	283	314	355	394	430	466	495	529	568	600	633

The obtained relationship between P_3 and θ_3 is:

$$\theta_3 = 0.28 * P_3 - 28 \tag{5.3}$$

The same method is used for obtaining the relations between θ_1 and θ_2 with P_1 and P_2 respectively.

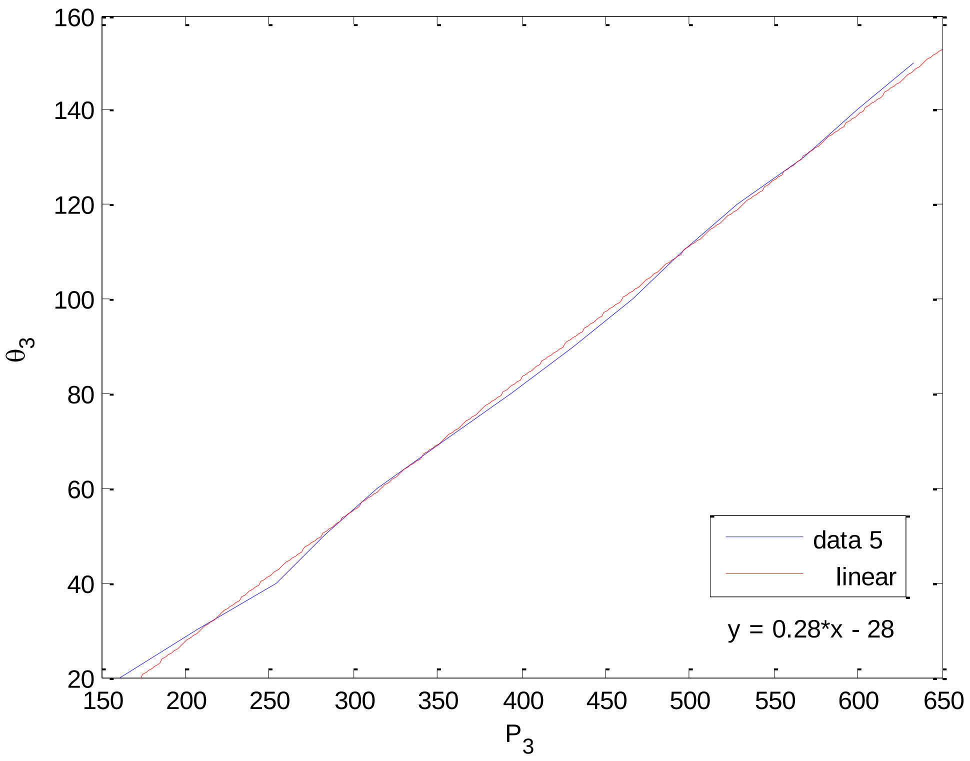


Figure 5.3: The relation between P_3 and θ_3 with the linear fitting

The obtained relationship between P_2 and θ_2 is:

$$\theta_2 = -0.25 * P_2 + 210 \tag{5.4}$$

The obtained relationship between P_1 and θ_1 is:

$$\theta_1 = 0.26 * P_1 - 94 \tag{5.5}$$

5.5 Mapping between the Manipulator Workspace and the Computer Screen Workspace

The computer first calculates the coordination X and Y of the end-effector by substituting the digital numbers that represent the angles in equations 5.3 and 5.4 to find θ_2 and θ_3 respectively. After that, these angles are used to find the coordination of X and Y for the end-effector by substituting them in equations 5.1 and 5.2 with the lengths of L1, L2 and L3 and the angle α.

The manipulator workspace is 800 mm width which ranges between (-400 mm, 400 mm), and 400 mm height which ranges between (350 mm-750 mm) as shown in the left panel of Figure 5.4. The zero point is 550 mm away on the y-axis from the workspace origin. However, the game's window is 800 pixel width, and 600 pixel height. The zero point is at the top left corner of the screen as shown in the right panel of Figure 5.4.

Suppose that $(\grave{X}, \grave{Y})$ represents the point on the screen, then:

$$\grave{X} = X + 400 \tag{5.6}$$
$$\grave{Y} = 600 - ((Y - 350) * 1.5) \tag{5.7}$$

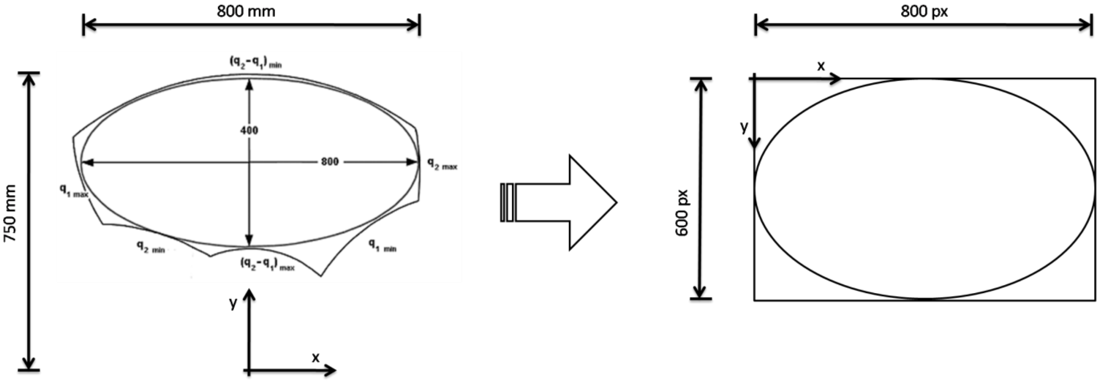


Figure 5.4: Manipulator and screen workspaces

The configuration angle $\Theta_1$is mapped directly to show the inclination of the game corresponds to inclination of the mechanical arm.

CHAPTER6

Communication System

6.1 Introduction

This chapter provides a background on wireless technology, the architecture and type of wireless networks, the characteristics of interfacing components and design and implementation for the rehabilitation robot. Wireless communication technology is becoming now an important part of modern life, from personal area network to local area network.

6.1.1 Importance of Using Wireless Technology in the Rehabilitation Robot

One of the most important features is the flexibility of the device, to achieve that a greater movement freedom can be reached using wireless communication. By this way, the rehabilitation robot can be used easily while patient is sitting or standing with the ability to move and transfer the robot to different places in the room by choosing the appropriate distance between the patient who uses the mechanical device and the screen. Another feature is the portability of the device, so the therapist can take the device to the patient’s home and reconstruct it in an easy way without worrying about wires.

6.1.2 Problem Definition

Using wires in the rehabilitation robot can increase the probability of cutting them because of the mechanical movements, so this will reduce the reliability of the device. To solve this problem, the patient can move the robot while playing the game. This can be done without a need for using cables between the mechanical robot and computer by using a wireless technology. As patient holds the end-effector of the mechanical robot with his/her arm, the motion that results in the workspace is detected using motion sensors; these sensors translate the change in position into electrical signals. These signals are then transmitted to computer using XBee. The computer in turn processes these signals and displays them on the screen.

6.1.3 Infrared Wave's Technology and Radio Transmission

Wireless network can use different basic transmission technologies, infrared light or radio transmission. Infrared waves have frequencies range from 300 GHz to 400 THz (wavelengths from 1 mm to 770 nm). The main advantages of infrared technology are simplicity, sender and receiver are very cheap and electrical devices do not interfere with infrared transmission. The main disadvantages of infrared technology are limited transfer rate, cannot penetrate walls or other obstacles and can be used for short-range communication. Radio transmission uses the license-free ISM band at 2.4 GHz. The main advantages of radio transmission are that, it can cover larger area, can penetrate walls and it has higher transmission rates. The main disadvantages of radio transmission are; it is only permitted in certain frequency band and very limited ranges of license-free bands are available but they are not the same in all countries [27]. For these reasons, radio transmission is used.

6.1.4 The Architecture of Wireless Networks

The architecture of a wireless network is divided into physical and logical architecture. Physical architecture is represented by wireless network topologies and hardware devices. Logical architecture is the structure of the standards and protocols that make a connection between nodes (physical devices) and control data flow between them.

6.2 ZigBee Technology

This section provides a background on ZigBee technology, general characteristics, device type, and architecture and frame format.

6.2.1 ZigBee

ZigBee, also known as IEEE 802.15.4, is a communications standard designed for low-power short-range communications between wireless devices. It is classified as a wireless personal area network (WPAN) [28]. The ZigBee standard has seen increasing interest from both commercial and military customers for applications such as wireless sensor networks, home devices, and industrial control [28].

6.2.2 ZigBee General Characteristics [29]:

- Dual Physical layer (2.4GHz and 868/915 MHz)
- Data rates of 250 kbps (2.4 GHz), 40 kbps (915 MHz), and 20 kbps (868 MHz)
- Low power (battery life multi-month to years)
- Channel access using Carrier Sense Multiple Access with Collision Avoidance
- Yields high throughput and low latency for low duty cycle devices like sensors and controls
- Multiple topologies: star, peer-to-peer, mesh
- Addressing space of up to 64 bit IEEE address devices, 65,535 networks
- Range: 50m typical (5-500m based on environment)

6.2.3 ZigBee Layer and Architecture

The ZigBee stack architecture is made up of a set of block called layers. Each layer performs a specific set of services for the layer shown in Figure 6.1.

Physical layer (PHY layer)

It was designed to accommodate the need for a low cost yet allowing for high levels of integration. The use of direct sequence Spread Spectrum allows the analog circuitry to be very simple and very tolerant towards inexpensive implementations [30]. In physical layer, digital stream is modulated before transmission. Binary Phase Shift Keying (BPSK) with 1 bit per symbol is used in the 868 and 915 MHz bands, and Offset Quadrature Phase-Shift Keying (O-QPSK) with two bits per symbol is used in the 2.4 GHz band. The data rate is 250 kb/s per channel in the 2.4 GHz band, 40 kb/s per channel in the 915 MHz band, and 20 kb/s in the 868 MHz band. Table6.1 summarizes these characteristic [30].

Table 6.1: Frequency bands and data rates for IEEE 802.15.4 PHY layer [30].

Band	Modulation	Bit rate	Symbol rate
868-868.6 MHz	BPSK	20 kb/s	20 kS/s
902-928 MHz	BPSK	40 kb/s	40 kS/s
2400-2483.5 MHz	O-QPSK	250 kb/s	125kS/s

Media access control

The Medium Access Control (MAC) layer controls access to the radio channel using the Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) mechanism and communicates with positive acknowledgement for successfully received packets. The channel access supports both beacon (e.g., star topology) and beaconless (e.g., peer-to-peer topology) modes. Other characteristics of MAC layer include synchronization, dynamic channel selection, frame reception, and acknowledge security and error correction [30].

Network layer

The responsibilities of the ZigBee network layer are:

- Starting a network; the ability to successfully establish a new network
- Joining and leaving network
- Configuration a new device
- Addressing
- Synchronizations within a network
- Routing
- Security

Application layer

The ZigBee application layer consists of the application support sub-layer (APS), the ZigBee device object (ZDO) and the manufacturer-defined application objects. The responsibilities of the APS sub-layer include maintaining tables for binding, which is the ability to match two devices together based on their services and their needs, and forwarding messages between bound devices. Another responsibility of the APS sub-layer is discovery, which is the ability to determine which other devices are operating in the personal operating space of a device. The responsibilities of the ZDO include defining the role of the device within the network ZigBee initiating and/or responding to binding requests and establishing a secure relationship between network devices [31].

6.2.4 ZigBee Device Type

There are three types of ZigBee devices: ZigBee coordinator (ZC), ZigBee router (ZR) and ZigBee end device (ZED). The coordinator and router are referred to as full function device (FFD). The end device is called reduced function device (RFD) [31].

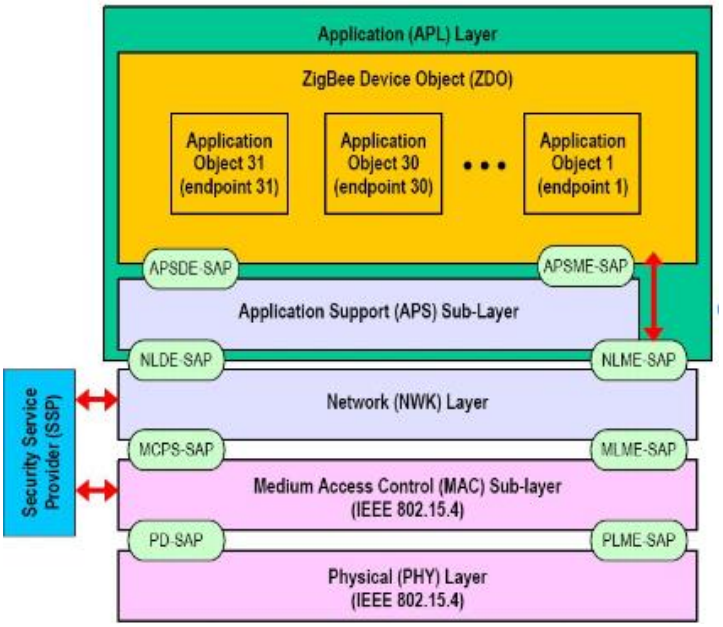

Application (APL) Layer
ZigBee Device Object (ZDO)
Application Object 31 (endpoint 31)
Application Object 30 (endpoint 30)
Application Object 1 (endpoint 1)
APSDE-SAP
APSME-SAP
Application Support (APS) Sub-Layer
NLDE-SAP
NLME-SAP
Security Service Provider (SSP)
Network (NWK) Layer
MCPS-SAP
MLME-SAP
Medium Access Control (MAC) Sub-layer (IEEE 802.15.4)
PD-SAP
PLME-SAP
Physical (PHY) Layer (IEEE 802.15.4)

Figure 6.1: Summarizing the ZigBee layers [31]

Full function device (FFD)

- ❖ Can function in any topology
- ❖ Capable of being the network coordinator
- ❖ Can talk to any other device

Reduced function device (RFD)

- ❖ Limited to star topology
- ❖ Cannot become a network coordinator
- ❖ Talks only to a network coordinator
- ❖ Very simple implementation

6.2.5 Frame Structure

The frame structures as shown in Figure 6.2 have been designed to keep the complexity to a minimum while at the same time making them sufficiently robust for transmission on a noisy channel. Each successive protocol layer adds to the structure with layer specific headers and footers. The IEEE 802.15.4 MAC defines four frame structures [32]:

1. A beacon frame, used by a coordinator to transmit beacons.
2. A data frame, used for all transfers of data.
3. An acknowledgment frame, used for confirming successful frame reception.
4. A MAC command frame, used for handling all MAC peer entity control transfers.

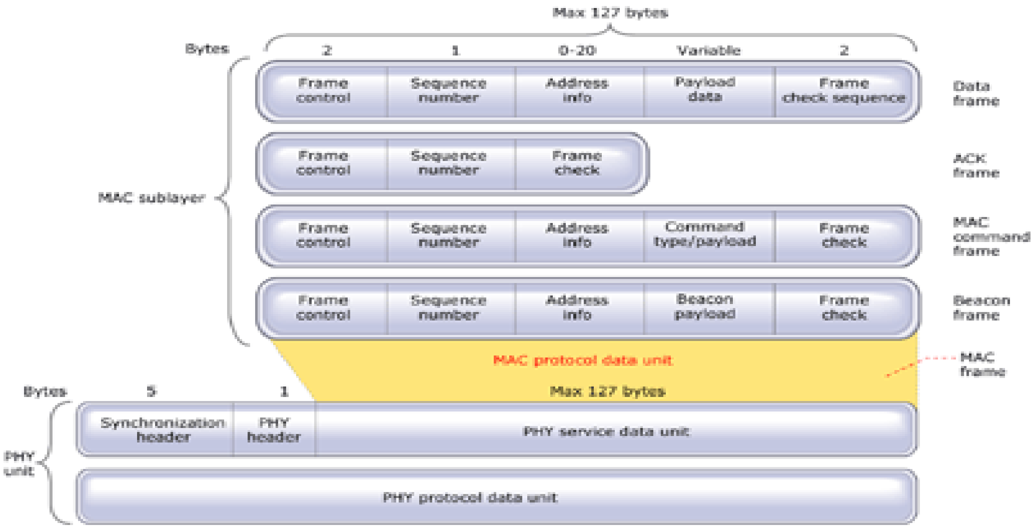


Figure 6.2: Data frames in ZigBee message [32]

6.2.6 Digi ZigBee Technology

Digi is a member of the ZigBee Alliance and has developed a wide range of networking solutions based on the ZigBee protocol. XBee provides devices an easy-to-implement solution that provides functionality to connect to a wide variety of devices. This is the very popular 2.4 GHz XBee module from Digi. This module allows a very reliable and simple communication between microcontrollers, computers, systems, really anything with a serial port, point to point and multi-point network are supported [33].

6.3 Communication System Design

This section provides functional specifications for the communication system in the rehabilitation robot. The communication stages are summarized in the block diagram shown in Figure 6.3.

6.3.1 Controller (ARDUINO) (block 2)

Although XBee can be used without additional controller but in this robot, it has many inputs and outputs working at the same time, also there is a need for processing the potentiometers' and loads' signals before the connection with the XBee, so it needs an additional controller between these sensors and the XBee. The potentiometers' signals are received by the controller (ARDUINO) to locate the position of the end-effector on the screen and determine the used configuration.

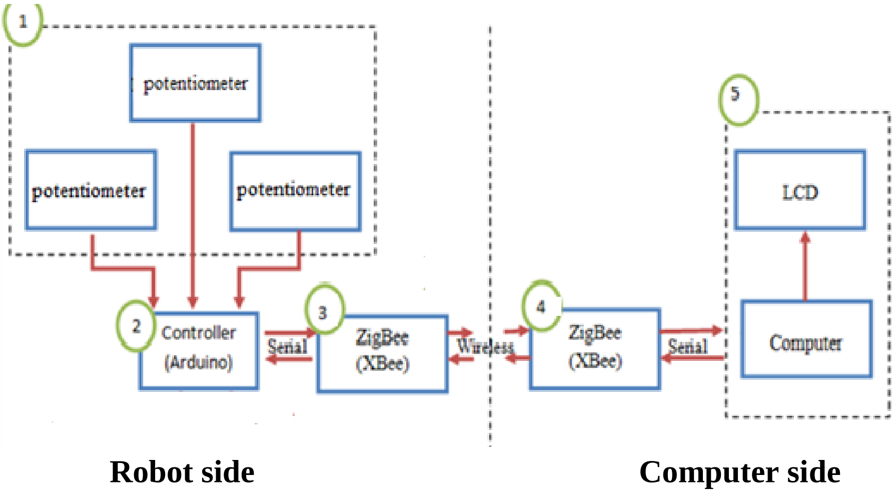


Figure 6.3: Block diagram

6.3.2 ZigBee (block 3, 4)

Each part is capable to transmit and receive data at the same time, the XBee in the computer side is connected with the computer by using an interfacing circuit and the other part will be connected to the mechanical arm using ARDUINO that connects the inputs (sensors). Next, these signals are connected to XBee RF module. The XBee is capable of directly sending these DIO signals from one module to another without any additional hardware. In computer side, there is another XBee RF module, this module receives these signals, and enters them to the computer. Transmission range in ZigBee is usually between 30 and 100 meters [32], depending on parameters such as path loss, energy overhead per hop and the transmitted energy per unit distance.

6.3.3 Computer (block 5)

Controller (ARDUINO) and XBee transmitter and receiver can be configured using computer software (more details will be explained in chapter seven). The XBee at the computer side is connected directly with the computer using an USB adapter. To reduce the power consumption, the XBee transceivers do not transmit the data all time but transmit the signals when the patient start play the game, the computer send request signal to the ARDUINO after that the data exchange between them. The subsystem flowchart in Figure 6.4 explains that.

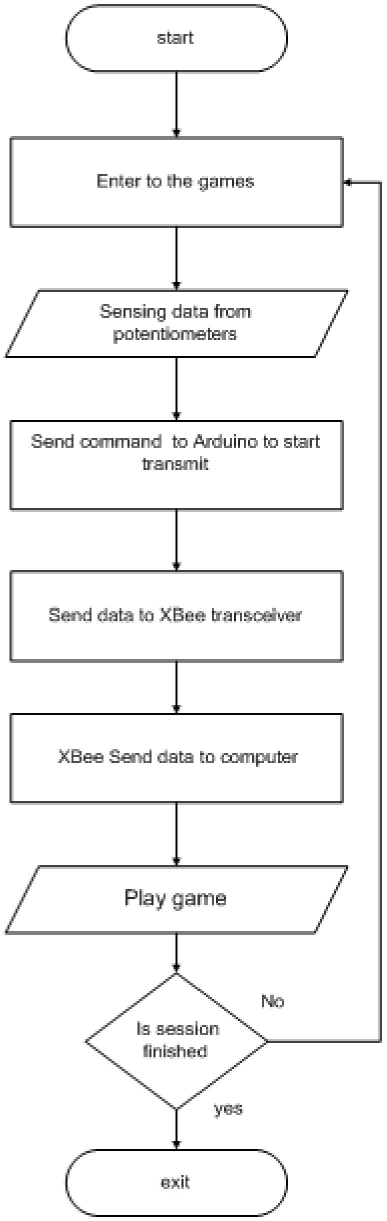

start
Enter to the games
Sensing data from potentiometers
Send command to Arduino to start transmit
Send data to XBee transceiver
XBee Send data to computer
Play game
Is session finished
No
yes
exit

Figure 6.4: Communication subsystem flowchart

6.4 Components of wireless interfacing in the rehabilitation robot

The communication system consists of a controller (ARDUINO), XBee RF modules and interfacing circuit with the computer. These components are explained in details in the following sub-sections.

6.4.1Controller (ARDUINO)

ARDUINO which is shown in Figure 6.5 is an open source microcontroller system. It is a very popular and easy to use, flexible, and fast to develop. ARDUINO Uno is one type of the microcontrollers. It has 14 digital input/output pins, a 16 MHz crystal oscillator, a USB connection, and a reset button. A 5v power battery is needed to get started. The ARDUINO Uno can be programmed with the ARDUINO software [34] as shown Figure 6.6.

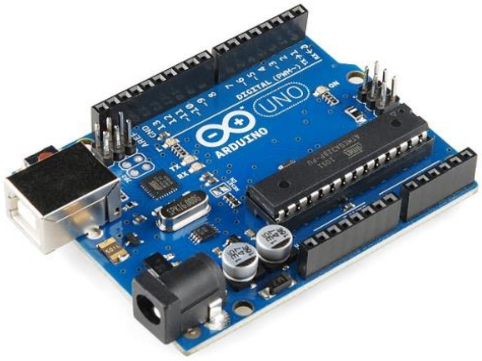
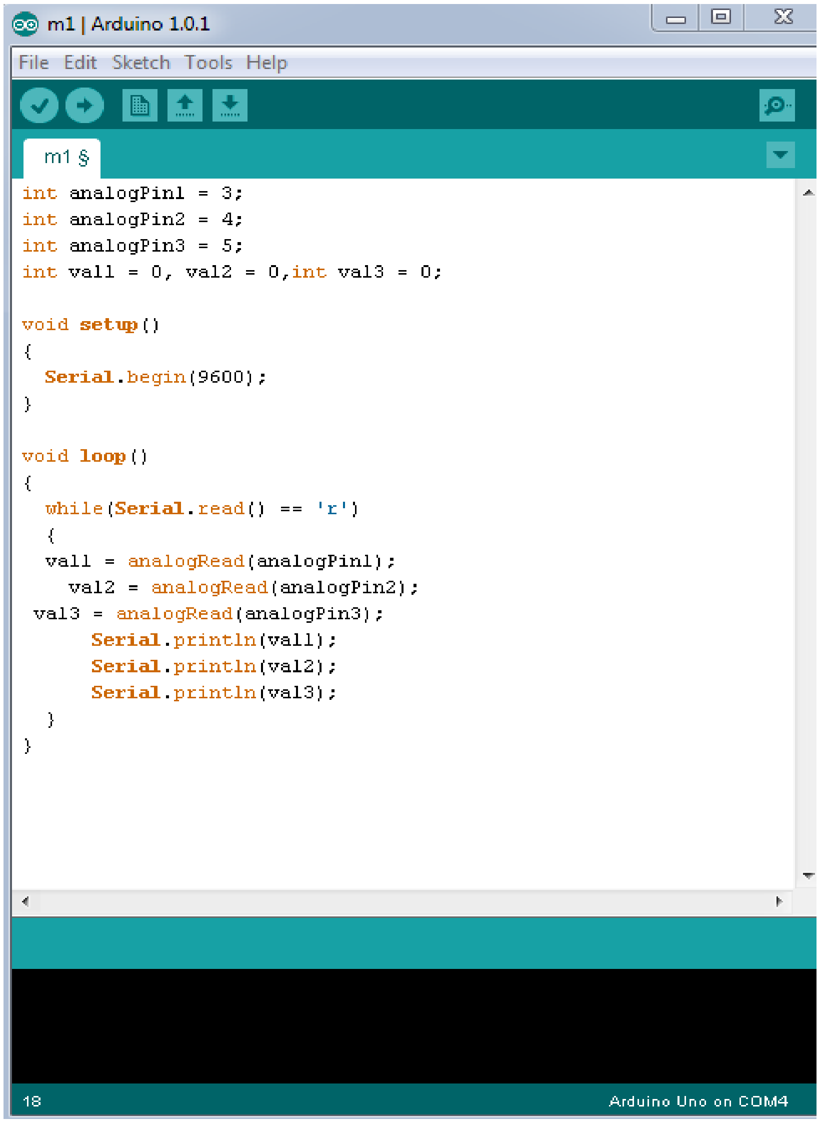


Figure 6.5: ARDUINO board

The ARDUINO hardware and software are both open source, which means the code, and the schematics, design, etc. are all open for anyone to take freely. The ARDUINO code for reading the three potentiometers is written in ARDUINO software as shown Figure 6.6. The characteristic of the ARDUINO Uno Microcontroller is explained in the Table 6.2 [34].

Table 6.2: The specification of ARDUINO Uno

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz



```
int analogPin1 = 3;
int analogPin2 = 4;
int analogPin3 = 5;
int val1 = 0, val2 = 0,int val3 = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  while(Serial.read() == 'r')
  {
  val1 = analogRead(analogPin1);
    val2 = analogRead(analogPin2);
 val3 = analogRead(analogPin3);
      Serial.println(val1);
      Serial.println(val2);
      Serial.println(val3);
  }
}
```

Figure 6.6: The ARDUINO code for reading the three potentiometers

External microcontrollers bring several important advantages to a wireless device, while the basic XBee radios can be a source of sensor data, they cannot be programmed to perform logical information processing. While XBee has many pins, it takes only four of them to create a working connection to communicate wirelessly by using the built in serial communications protocol. Two of the four connecting are for power source and the others are for transmitting and receiving data. The XBee pins are spaced 2 mm apart, so the XBee cannot be placed directly into a breadboard. A basic breakout board as shown in Figure 6.7 is the least expensive adapter for connecting an ARDUINO. Power is taken from the 5V pin of the ARDUINO and regulated on board to 3.3VDC before being supplied to the XBee [34].

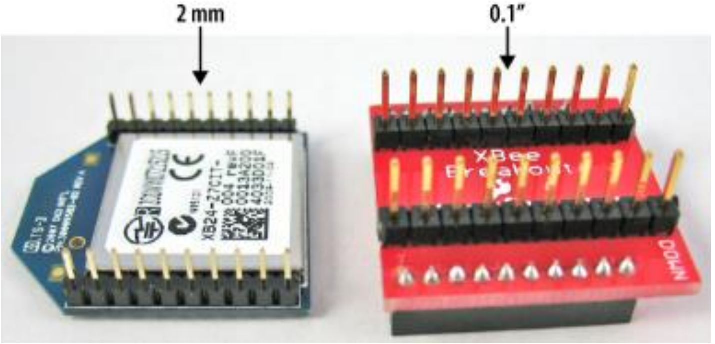


Figure 6.7: Breakout board showing pin spacing [34]

6.4.2 XBee RF modules

There are two basic varieties of XBee radio physical hardware [34]:

1. XBee Series 1 hardware these radios use a microchip made by free scale to provide simple, standards-based point-to-point communications and is great for simple cable replacements and smaller-sized systems.
2. XBee Series 2 hardware these radios use a microchip designed with larger sensor networks.

Both the Series 1 and Series 2 radios are available in two different transmission powers, regular and PRO (The regular version is called simply an XBee). The XBee-PRO radio in Figure 6.8 has more power and is larger and more suitable for using in this robot.

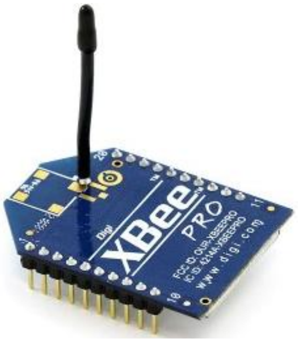


Figure 6.8: XBee modules [34]

The XBee Series 1 RF Modules regular version standards-based point-to-point communications, it is engineered to operate within the ZigBee protocol and support the needs of low-cost, low-power wirelessly. The modules require minimal power and provide reliable delivery of data between remote devices. The modules operate within the ISM 2.4 GHz frequency band [30]. The specifications of the XBee are summarized in the Table 6.3 [33].

Table 6.3: Specifications of the XBee Series 1 RF Modules PRO version [33].

Indoor/Urban Range	Up to (90 m)
Outdoor RF line-of-sight	Range Up to (500 m)
Transmit Power Output	50mW (17dBm)
RF Data Rate	250,000 bps
Receiver Sensitivity	-102dBm
Supply Voltage	2.8 – 3.4 V
Transmit Current	45mA (@ 3.3 V)
Operating Frequency	ISM 2.4 GHz
Dimensions	0.960” x 1.087” (2.438cm x 2.761cm)
Number of pins	10ADC input pins and8 digital IO pins
Programming	AT or API command set
Security	128-bit encryption

Each XBee radio has a tiny computer, this internal microcontroller runs a program, also known as firmware, that performs all its addressing, communication, security, and utility functions. It can configure this firmware with different settings that define its local address, which type of security is enforced and how it should read sensors connected to its local input pins. To change or upgrade the firmware, a program called X-CTU can be downloaded from the Digi website, and this program is free. The pin layout of the XBee PRO Series 1 RF Modules is as shown in Table 6.4. The important pins are power, ground, serial RX/TX, and a digital out for the reset as shown Figure 6.9.

Table 6.4: The pin layout of the XBee PRO Series 1 RF Modules [33].

Pins	
1 VCC	3.3v
2 DOUT	serial RX
3 DIN	serial TX
4 not connected	
5 RESET	digital out
6-9 not connected	
10 GND	0v/GND
11-20 not connected	

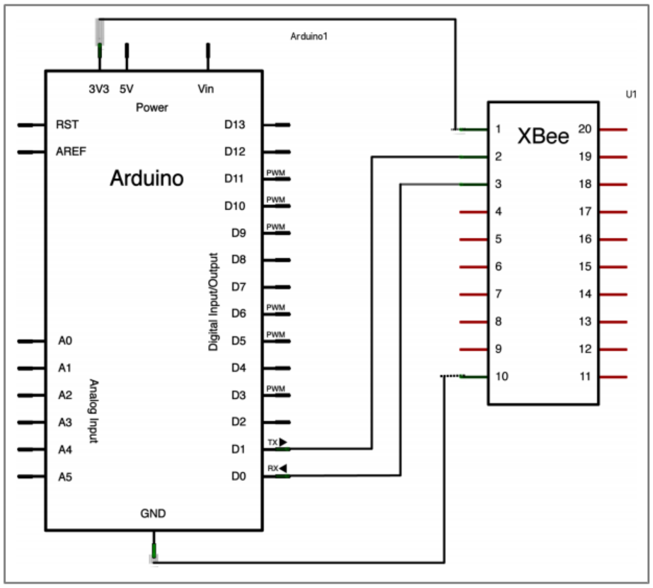


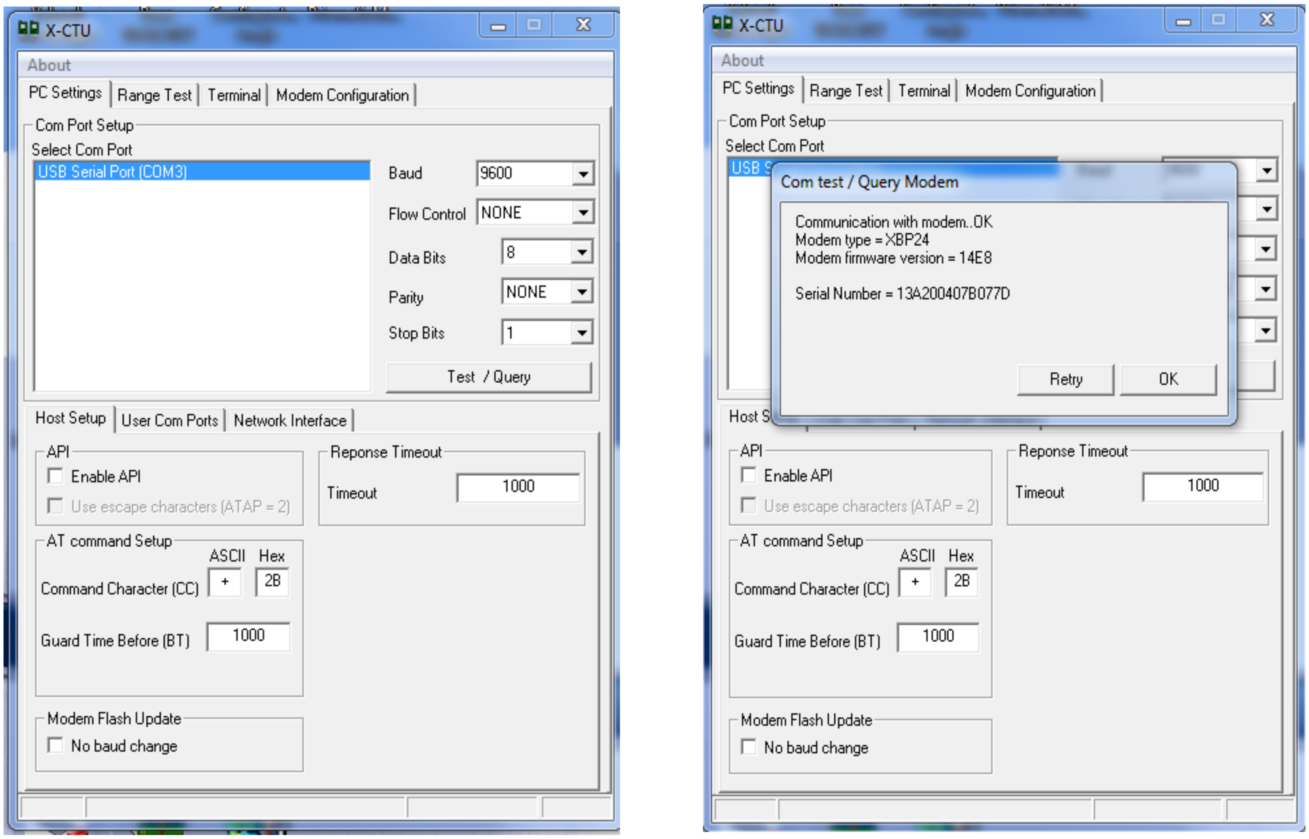
Figure 6.9: ARDUINO adapter hack schematic

XBee Addressing

Every RF data packet sent over-the-air contains a Source Address and Destination Address field in its header. The RF module conforms to the 802.15.4 specification and supports both short 16-bit addresses and long 64-bit addresses [32].

XBee configuration:

XBee parameters are set through serial port by stabilizing a serial connection between the XBee board and computer device using a USB adapter through "USB Serial Port”. Using X-CTU software, the firmware can be upgraded, the parameters can be updated and tests can be performed easily as shown in Figure 6.10. One of the XBees are connected on COM port (COM 3 here) and baud rate is set to 9600 (default).



X-CTU

About

PC Settings | Range Test | Terminal | Modem Configuration

Com Port Setup

Select Com Port

USB Serial Port (COM3)

Baud 9600

Flow Control NONE

Data Bits 8

Parity NONE

Stop Bits 1

Test / Query

Host Setup | User Com Ports | Network Interface

API

☐ Enable API

☐ Use escape characters (ATAP = 2)

Reponse Timeout

Timeout 1000

AT command Setup

ASCII Hex

Command Character (CC) + 2B

Guard Time Before (BT) 1000

Modem Flash Update

☐ No baud change

X-CTU

About

PC Settings | Range Test | Terminal | Modem Configuration

Com Port Setup

Select Com Port

USB S

Com test / Query Modem

Communication with modem..OK
Modem type = XBP24
Modem firmware version = 14E8

Serial Number = 13A200407B077D

Retry OK

Host S

API

☐ Enable API

☐ Use escape characters (ATAP = 2)

Reponse Timeout

Timeout 1000

AT command Setup

ASCII Hex

Command Character (CC) + 2B

Guard Time Before (BT) 1000

Modem Flash Update

☐ No baud change

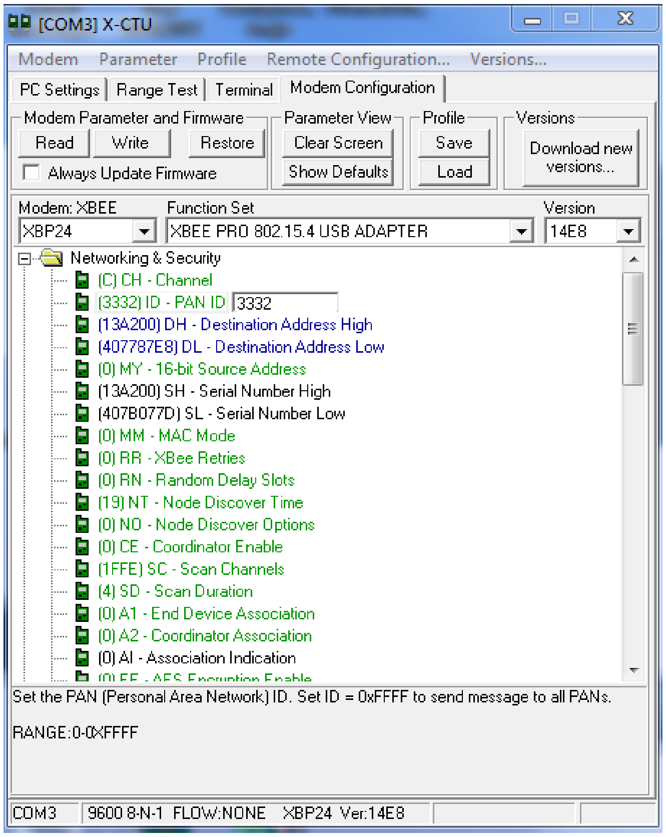
Figure 6.10: Computer setting

The next step is to go to Modem Configuration and click the Read Button as shown Figure 6.11. In this step more parameters are set as the following:

- Select the "**modem**" type "XBP24"
- **"Function set"** is the firmware in side XBee. It is responsible for different topologies and configuration. Here, the XBee module at computer side uses "USB ADAPTER" and the other module in the robot side is used as XBEE PRO 802.15.4.
- Check the "**Always update firmware"** box.
- Click "**Write**" to load the new firmware.

Set PAN ID and SC (Scan Channel) to an arbitrary value as shown Figure 6.12. All devices in the network must use same value of PANID and same channel of SC. ID is the identifier by which the XBee will know that they belong to the same network.

- **PAN ID** in this network = 3332.
- **CH** = C.
- **SC**=1FFE



Modem: XBEE: XBP24
Function Set: XBEE PRO 802.15.4 USB ADAPTER
Version: 14E8

Networking & Security
- (C) CH - Channel
- (3332) ID - PAN ID 3332
- (13A200) DH - Destination Address High
- (407787E8) DL - Destination Address Low
- (0) MY - 16-bit Source Address
- (13A200) SH - Serial Number High
- (407B077D) SL - Serial Number Low
- (0) MM - MAC Mode
- (0) RR - XBee Retries
- (0) RN - Random Delay Slots
- (19) NT - Node Discover Time
- (0) NO - Node Discover Options
- (0) CE - Coordinator Enable
- (1FFE) SC - Scan Channels
- (4) SD - Scan Duration
- (0) A1 - End Device Association
- (0) A2 - Coordinator Association
- (0) AI - Association Indication
- (0) EE - AES Encryption Enable

Set the PAN (Personal Area Network) ID. Set ID = 0xFFFF to send message to all PANs.

RANGE:0-0XFFFF

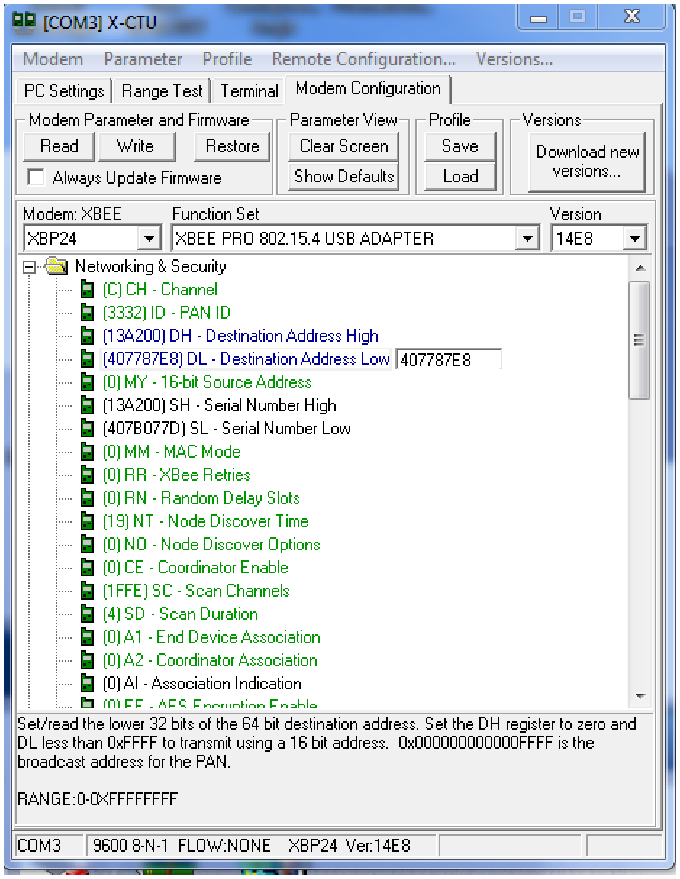
COM3 | 9600 8-N-1 FLOW:NONE | XBP24 Ver:14E8

Figure 6.11: Setting the modem configuration

Set DH and Dl which are the source and destination addresses of the XBee module as shown Figure 6.13. The address of the XBee module in computer side is DH=13A200, DL=4077877E8, which is a specific destination, and the address of the XBee module in robot side is DH=13A200, DL=407B077D, which is also a specific destination.

6.4.3 USB Adapters (interfacing circuit with computer)

A USB Adapter is used to connect the XBee to a USB port on the computer. XBee Explorer Dongle which is shown in Figure 6.13 is one of the smallest adapters, it needs no external cable. The Dongle does not provide any access to the radio beyond USB. On the other hand, it is a very small all-in-one device that is easy to carry in a pocket. The XBee Explorer Dongle can be powered via the USB connection**.**


[COM3] X-CTU
Modem Parameter Profile Remote Configuration... Versions...
PC Settings | Range Test | Terminal | Modem Configuration
Modem Parameter and Firmware
Read
Write
Restore
Parameter View
Clear Screen
Show Defaults
Profile
Save
Load
Versions
Download new versions...

□ Always Update Firmware

Modem: XBEE: XBP24
Function Set: XBEE PRO 802.15.4 USB ADAPTER
Version: 14E8

Networking & Security
(C) CH - Channel
(3332) ID - PAN ID
(13A200) DH - Destination Address High
(407787E8) DL - Destination Address Low 407787E8
(0) MY - 16-bit Source Address
(13A200) SH - Serial Number High
(407B077D) SL - Serial Number Low
(0) MM - MAC Mode
(0) RR - XBee Retries
(0) RN - Random Delay Slots
(19) NT - Node Discover Time
(0) NO - Node Discover Options
(0) CE - Coordinator Enable
(1FFE) SC - Scan Channels
(4) SD - Scan Duration
(0) A1 - End Device Association
(0) A2 - Coordinator Association
(0) AI - Association Indication
(0) EE - AES Encryption Enable

Set/read the lower 32 bits of the 64 bit destination address. Set the DH register to zero and DL less than 0xFFFF to transmit using a 16 bit address. 0x000000000000FFFF is the broadcast address for the PAN.

RANGE:0-0XFFFFFFFF

COM3 | 9600 8-N-1 FLOW:NONE | XBP24 Ver:14E8

Figure 6.12: Setting destination address

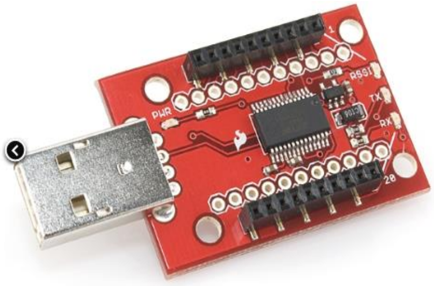


Figure 6.13: XBee Dongle [35]

6.5 Serial communication

Data is transmitting between computer and XBee modules using serial communication. Serial communication uses a transmitter to send data, one bit at a time, over a single communication line to a receiver. Each transmitted character is packaged in a character frame that consists of a single start bit followed by the data bits, the optional parity bit, and the stop bit or bits [36]. Serial communications depends on the two UARTs (the microcontroller's and the RF module's) to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits), by default, XBee RF Modules operate in Transparent Mode. When operating in this mode, the modules act as a serial line replacement all UART data received through the DI pin is queued up for RF transmission. When RF data is received, the data is sent out the DO pin. When serial data enters the RF module through the DI pin, the data is stored in the DI Buffer until it can be processed [37]. A serial communication session is opened between the software system and the external controller (ARDUINO), to receive inputs from sensors through XBee RF module. Serial connection established using COM port with a USB cable on RS232 standards. Input and Output Real-Time Controlling Software can deal with reading and writing by C# code that shown in Figure 6.14.

```
using System;
using System.Collections.Generic;
using System.IO.Ports;

namespace VirtualRehabilitation
{
classSerialInterface
{
SerialPort SP
int Serial_Rate = 9600
public SerialInterface(String COM_Port
        {
            SP = newSerialPort(COM_Port, Serial_Rate
            SP.Open();
        }

publicvoid SerialRead(refint Theta1, refint Theta2,
refint Theta3)
        {
            SP.Write("r
string Line = SP.ReadLine();
int Delimiter1 = Line.IndexOf(":");
int Delimiter2 = Line.LastIndexOf(":");
            Theta1 = int.Parse(Line.Substring(0, Delimiter1
            Theta2 = int.Parse(Line.Substring(Delimiter1 + 1,
Delimiter2)); //
            Theta3 = int.Parse(Line.Substring(Delimiter2));
return;
        }

publicvoid SerialWrite(int Load1, int Load2)
        {
            SP.Write("w");
            SP.WriteLine(Load1 + ":" + Load2
        }
    }}
```

Figure 6.14: C# Code for Reading and Writing from and into Serial Connection

6.6 Data

The data consist of three values of potentiometers as shown in Figure 6.15. Each potentiometer has a voltage range from 0-5 volt. This value changes according to the change of mechanical arm. These values enter the ARDUINO. They are converted from analog to digital and transmitted over the XBee module and received from the XBee module in computer side. The computer deals with these digital values which range between 0-1023. Each value of the potentiometer is converted to 10 bit (0000000000-111111111).

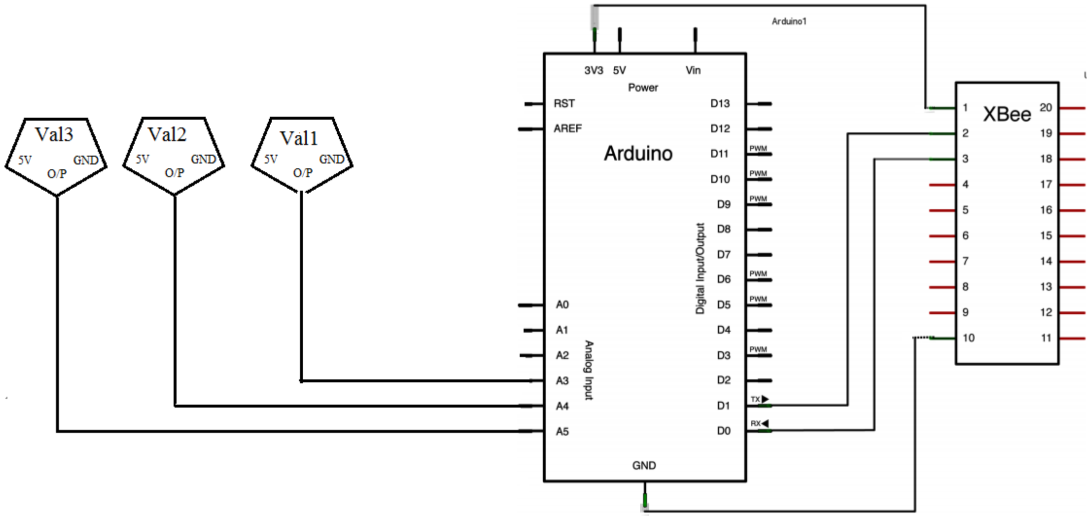


Figure 6.15: Potentiometers, ARDUINO and XBee

The data frame in the Virtual Rehabilitation Robot consist of values three angles to be measured, starts with the configuration angle (Θ_1) (val1) which determines the required plane before the rehabilitation process starts, and separate data between the first and second potentiometers (new line \n equivalent to 10 in ASCI code). The other two angles determine the position coordination of the end-effector to be mapped on the screen. The angle between the base and the arm is called (Θ_2) (val2) and the angle between the arm and the forearm is called (Θ_3) (val3) and separate data between the second and third potentiometers (new line \n equivalent to 10 in ASCI code). This frame is transmitted ten times in second. Each value of the potentiometer is represented by 16 bit as shown in Table 6.5.

Table 6.5: The size of the sent data

Potentiometer(val1)	\n	Potentiometer(val2)	\n	Potentiometer(val3)
16 bit	8bit	16bit	8bit	16bit
	0000001010		0000001010	

And the transmit frame consist of request command (r) equivalent in ASCI code is 114 in decimal. So the size data every second: require is three potentiometer need to 48 bit and sequence between potentiometer is 16 bit, This frame transmit ten time in second so the total data size is approximation 640 bit source data , and needed to channel coding additional bit approximating equal 20 byte =160 bit. The XBee module can transmit 250000 bit per second which is enough to use it in virtual rehabilitation robot

6.7 Power consumption

The power consumption can be calculated by determining the consuming power of each electrical component.
The XBee module can consume power according data sheet of XBee is,

- Operating Current (Transmit) is 170mA @3.3 V
- Operating Current (Receive) is 45 mA @3.3 V
- Idle Current (Receiver off) 15mA
- Power-down Current 10 uA at 25C
- Power consumption for ARDUINO is 5V and current is 40mA.
- Voltage from the ARDUINO is 5V to run potentiometers and the DC Current per I/O Pin is 40mA and the power consumption for each potentiometer is I*V=200mW

6.8 Power supply

The power supply has three options: a 5v transformer, a USB cable or a 9v battery with a 5v regulator for the ARDUINO. The other components (XBee module, potentiometers) take power directly from ARDUINO.

6.9 Delay

The delay is an important factor in the device to ensure playing in real time as the nature of the sent data requires fast transmission. The delay defines how long it takes for an entire message to completely arrive at the destination from the time the first bit is sent out from the source. The delay is made of four components: propagation time, transmission time, queuing time and processing delay. In order to know how much sampling frequency that this computer can deal with per second without using XBee, a simplified C# code that is shown in Figure 6.16 has been run on the computer. The results show that, 10000 readings requires 19885 ms, where the chosen computer can deal with 503 readings per second approximately and each reading need approximate 2 ms.

The test code consists of:

1. Virtual serial reading for three angles.
2. Normalize three angles.
3. Calculate the x-y position for the end-effector.
4. Normalize x and y results.
5. Positioning of the onscreen icon on the normalized x-y position.

```
publicvoid Test() // Test Function ............................................
{ // ...........................................................................
Stopwatch stopwatch = newStopwatch(); // Create Stopwatch Opject .........
    stopwatch.Start(); // Start Ticks and Milliseconds Timer ...................
    Theta3 = 0; //..............................................................
int X = 0, Y = 0; // .......................................................
for (int i = 0; i < 10000; i++) // Do This 10000 Times ...................
    { // Test For Loop .........................................................
int Theta1 = 0, Theta2 = 0, Theta3 = 0; //.............................
int X = 0, Y = 0; // ...................................................
        SerailRead(ref Theta1, ref  Theta2, ref Theta3); // .................(1)
        Normalize(ref Theta1, ref Theta2, ref Theta3); // ...................(2)
        CalcXY(ref Theta1, ref Theta2, ref Theta3, ref X, ref Y); //........(3)
        Normalize(ref X, ref Y); // .........................................(4)
        SetEndEffector(ref X, ref Y); // ....................................(5)
    } // End of Loop ...........................................................
    stopwatch.Stop(); // Stop Ticks and Milliseconds Timer .....................
MessageBox.Show("Ticks: " + stopwatch.ElapsedTicks + // Out Results .......
"\nmS    : " + stopwatch.ElapsedMilliseconds); // ............
} // End .......................................................................
```

Figure 6.16: C# Code for Performance Testing

Propagation Time: measures the time required for a bit to travel from the source to the destination. The propagation time is calculated by dividing the distance by the propagation Consider average distance equal 10 m and propagation speed equal speed of light 3*10^8 m/s Propagation Time is very small and equal 30 nanosecond. Transmission Time: the time required for transmission of a massage depends on the size of the message and the bandwidth of the channel. Message size approximation 640 bit source data, additional bit channel coding approximating equal =160bit.Transmission Time =800 bit/250000 bit=3.2 millisecond so the total delays equal 5 millisecond.

6.10 Communication System Testing and Experiment

When connect the potentiometer in oscilloscope with external power supply (not from ARDUINO) can show the signal as shown Figure 6.17 and 6.18 with high noise can minimize the noise when take the power from ARDUINO. After connecting the potentiometers with ARDUINO and transmit the data through the XBee, these values are read from the computer through interface port as shown Figure 6.19.

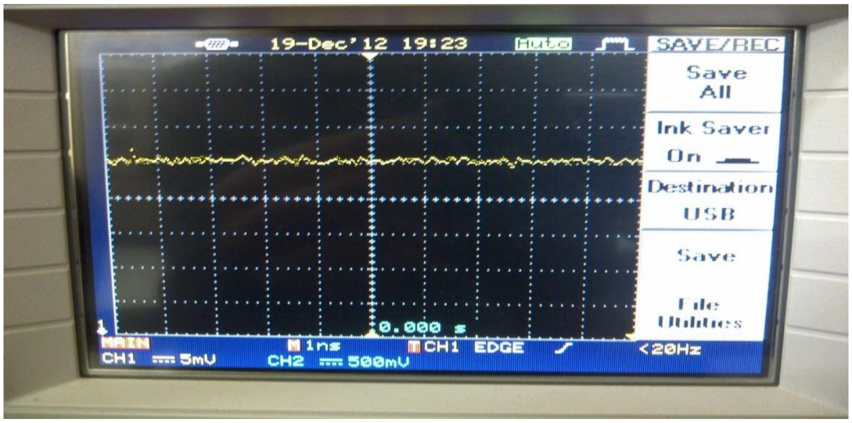


Figure 6.17: The signal of Potentiometer with external power supply

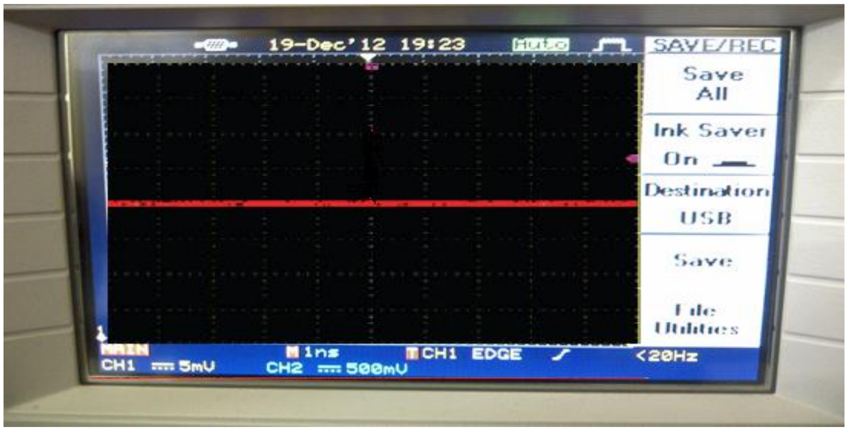
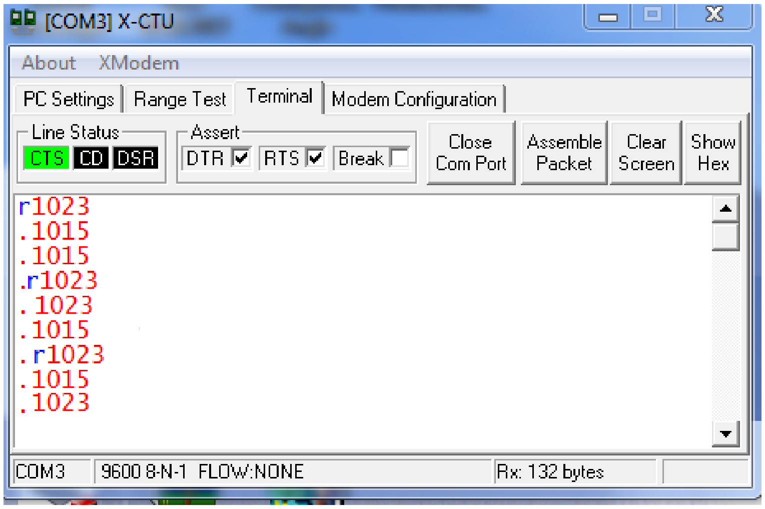


Figure 6.18: The signal of potentiometer with ARDUINO power supply



```
r1023
.1015
.1015
.r1023
. 1023
.1015
. r1023
.1015
.1023
```

COM3 | 9600 8-N-1 FLOW:NONE | Rx: 132 bytes

Figure 6.19: The Potentiometers values in serial interface

CHAPTER 7

Software System

7.1 Introduction

The importance of virtual rehabilitation arises due to the lack of motivation in the traditional rehabilitation with physiotherapy, where the patient feels bored as a direct result of monotony and repetition in therapeutic exercises. However, in order to get the motivation and enhance the rehabilitation process, an interesting computer game will be placed in front of the patient, which works in parallel with the movement of the patient's limbs to get the patient motivated to do the therapeutic exercises correctly and joyously.

Every therapeutic session consists of playing different computer games which are designed on the bases of therapeutic protocols in order to get correct results by correct exercises. However, one of these games concerned with measuring the therapeutic level, and evaluate the patient's progress, where this computer game will be played at the start of every therapeutic session. Furthermore, those games can help to improve the patient's mental linking and thinking if the games require the usage of patient's cognitive skills.

On the other hand, the therapeutic process needs to store and retrieve the patient information, and his/her therapeutic progress, which can be achieved by a suitable data management system. Moreover, the software system must control the signals that come from the sensors, and represent it on the LCD Screen, and send the suitable signals to control the load, because all of the project's parts must work simultaneously in real time, to achieve the synchronization.

The software system is one of the major systems in this project. It is connected to other systems by a wireless module that sends the outputs and receives the inputs serially from a controller that reads the sensor's values, and sends the control signals to the breaks system. The software system is divided into four integrated subsystems as shown in Figure 7.1.

The first part of the software system is the Entrance Graphical User Interface (E-GUI), it is responsible for allowing the system's user to create, edit, delete, display, and select any patient's profile to start the therapeutic session. Moreover, the system's user able to change the game options, and devices configuration.

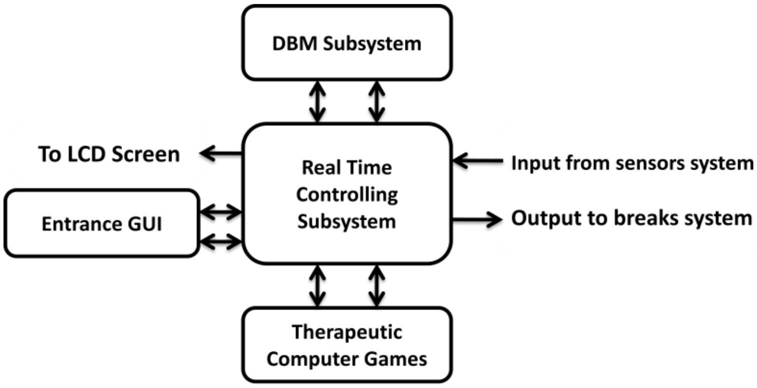

DBM Subsystem
To LCD Screen
Real Time Controlling Subsystem
Input from sensors system
Output to breaks system
Entrance GUI
Therapeutic Computer Games

Figure 7.1: Block Diagram for Software System

Another subsystem is the Therapeutic Computer Games Software (TCGS) it is a collection of different games. In fact, those games are based on the therapeutic protocols, yet it is designed to achieve the motivation and kill the boredom to make the therapeutic exercises more effective and enjoyable. In addition, each game has multilevel stages, where the playing difficulty changes from easy level to hard. Finally, those games can help to improve the patient's mental linking and thinking if the games require the usage of patient's cognitive skills.

The third subsystem is the database management subsystem (DBMS) which achieves the therapeutic progress tracking by progress reports and multi-user system properties further. It is responsible for recording and retrieving patient’s information and the treatment progress data for each patient.

The last software system part is the Real Time Controlling Subsystem Software (RTCSS), which represents the core of the whole project. RTCSS is the software interface between the device systems; it is read the inputs that come from sensors system, and output the control signals to the load. Moreover, RTCSS is the software interface between internal software subsystems to get the parallel work integration, by transceiving the parameters between the internal software subsystems.

7.2 Entrance Graphical User Interface (E-GUI)

A system's user (patient or physiotherapist or system's administrator) will be allowed to access with patient's profiles, therapeutic games options, and system's devices configuration by graphical screen that easy to use. A system's user can use the computer keyboard and mouse directly to enter the user interface screens or deal with onscreen components.

As shown in the following Use-Case diagram for the patient and physiotherapy permissions in Figure 7.2. A patient can select his/her profile to start playing exercises game in his/her therapeutic session. In addition, A patient able to display a simple progress reports that represents his/her therapeutic case. However, a physiotherapy has the ability to create, edit and delete any patient profile and display advanced and detailed progress reports for any patient yet. Furthermore, a physiotherapy can able to enter special sessions for special cases of patients, and change whole games' options like: time, difficulty, and motivation options.

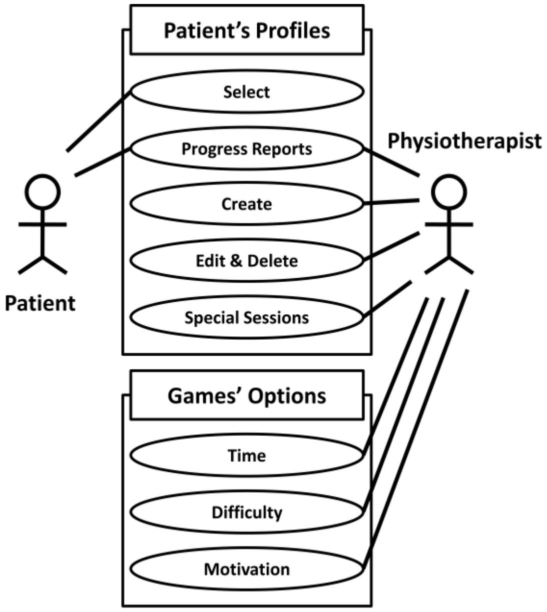


Figure 7.2: Use-Case Diagram #1

A system's administrator has the permissions to control the configuration of the system's parts that relates to the software system. A system's administrator can able to change the workspace size and resolution on the LCD Screen, and change the inputs and outputs ranges that comes serially from wireless module. Moreover, A system's administrator has the permission to set the device constants and parameters as rod's lengths, and constant angles that required in the end-effector's position calculations. Yet he can change the images, audio and database files directories, and select the correct serial port (com port) that currently opening for sending and receiving the data. As shown in Use-Case diagram in Figure 7.3.

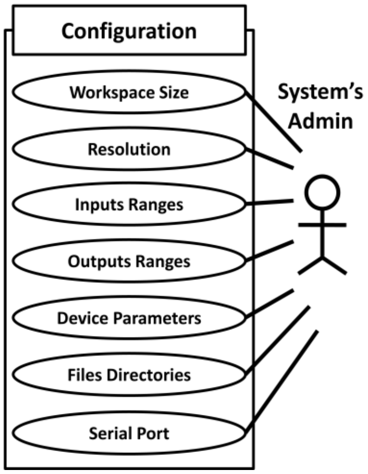


Figure 7.3: Use-Case Diagram #2

7.3 Therapeutic Computer Games Software (TCGS)

Therapeutic Computer Games part is the software part that consists of four different games, which are based on the therapeutic protocols in order to make the therapeutic exercises more effective and enjoyable. So, the motivation factor for the patient is one of the most important factors in these games design. Furthermore, TCGS should be designed to achieve eye-hand coordination which is the ability to coordinate the information received through the patient's eyes to control and move his/her arm.

Reaching targets, following paths, and object manipulation are some categories of therapeutic exercises games [38]. However, reaching targets and following paths are the chosen game's categories in therapeutic computer games software.

Some previous studies in virtual rehabilitation use different forms of reaching targets games, and for example:

1. Eight or sixteen targets placed on the circumference of a circle, where the patient shall move his arm and shoulder to reach those targets respectively by a circular movement similar to the motion of a clock [13].
2. Some targets with different colors appear on the screen in different positions, and the patient stretches his arm in a specified distance and orientation to reach those targets [39].
3. Many objects both in 2D, and 3D environments are shown on the screen, and the patient extend his/her arm in order to reach these targets [40].

Furthermore, some studies in virtual rehabilitation use different forms of following paths games, and for example:

1. Maze, LeLe Hand Writing Pattern and Complex Labyrinth, are three games in pattern, where the subject is required to move the stylus along and follow the specified trajectory while holding the stylus on the X-Y plane [41].
2. The arm's movement for the patient required to be along a circular trajectory, both clockwise and counterclockwise, where they are constrained by means of an impedance controller provided by the exoskeleton [38].

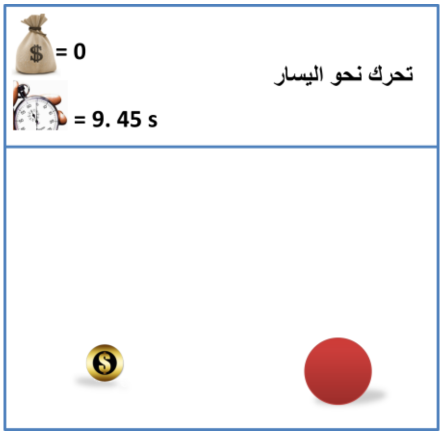
Collect Money Game (CMG) and Falling Parachutes Game (FPG) which are two reaching targets games, and Steady Hand Game (SHG) and Memorize Path Game (MPG) which are two following paths games. Those four are the chosen games in therapeutic computer games software. However, those games will be designed to get effective and correct therapeutic exercises and to get more motivation for the patient.

7.3.1 Collect Money Game (CMG)

Collect Money Game, is the first chosen computer game, which is one of reaching targets games that is simple to play for the patient, and fast to calculate the evaluation parameters by placing the targets in different places on the workspace.

A white space screen with a small red circle that represents the zero point position will be shown on the screen. As progressing in time a dollar sign icon is added to the screen with an Arabic[*] text command that asked the user to move his/her arm to reach the target. The red circle moves with the arm movement to catch the dollar sign icon, as shown in Figure 7.4.

After reaching a target, a new one will be added in another place, as shown in Figure 7.5. The same thing will be repeated until the game's time finished. Then the game's results will be displayed on the screen, and moved to the database file through DBMS.



= 0

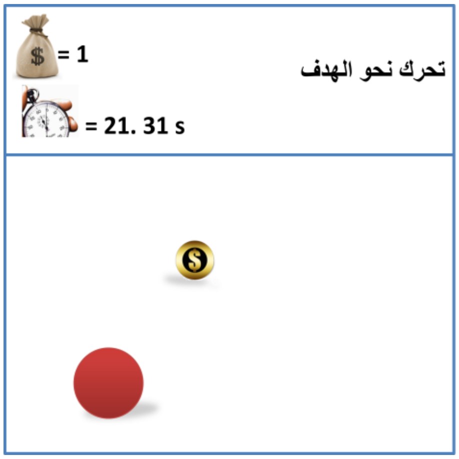
تحرك نحو اليسار

= 9. 45 s

Figure 7.4: CMG Snapshot #1

7.3.1.1 Positioning of Targets in CMG

The simplest exercise is just performing the same movement repeatedly by reach the same task every time, which is not the optimal for retaining learning over time. In addition, the second methodology of practice is that introducing a fixed pattern of movements by reaching the same sequence of targets in the same acquisition session, where that may improve performance in that particular task but not transfer to any activities of daily living activates [42]. Contextual interference sequence of presentation which is the random movements for reaching random targets in random positions, where that leads to better performance, and helps more in motor learning and stroke recovery for daily living activates [43]. Moreover random schedule promotes considering each movement as a problem to be solved [42].



= 1

تحرك نحو الهدف

= 21. 31 s

Figure 7.5: CMG Snapshot #2

[*]: Arabic language is the mother language for patients.

7.3.1.2 Anatomy for (A to B to C) Movements

Suppose that, the patient arm is placed on position A, Target 1 placed on position B and Target 2 that hidden will be placed on position C as shown in Figure 7.6. However, a patient's arm movement from position A to position B to reach Target 1 on x-y plane without knowing the position of Target 2 contains [44]:

1. Movements for the elbow and shoulder joints
2. Controlling for the muscles.
3. Eye-hand coordination with feedback.
4. Cognitive training.
5. Movement's plasticity improvement.

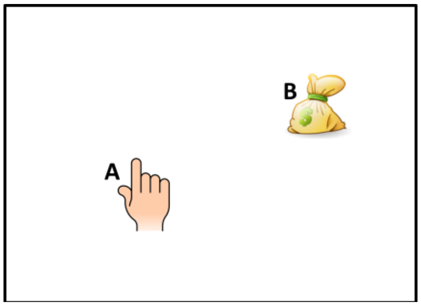


Figure 7.6: A to B Movement

Moreover, a movement form position B to reach Target2 that placed at the same time of reaching Target1 as shown in Figure 7.7 contains more:

1. Movement controlling.
2. Reaction training.
3. Each movement as a problem to be solve [42].

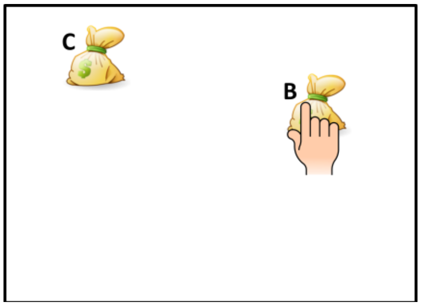


Figure 7.7: B to C Movement

7.3.2 Steady Hand Game (SHG)

Steady Hand Game, is the second chosen computer game, which is one of following paths games that requires some concentration skills to follow the allowable path of movement.

Different shapes of bars for different game's levels, and a ring circle with stick that describes the patient's arm position will be shown on the screen, and a dollar sign icon which describes the target placed at the end position of the bar. in the maze. The patient will move his/her joints to get a movement for the stick on the screen to follow the correct path that leads to the target through the bar with the least possible number of collisions, as shown in Figure 7.8, which describes this snapshot of Steady Hand Game. In addition, A patient will move his/her arm through the bar in order to lead to the target, and the bar thickness starts to be wider as progress of navigating throw the allowable path.

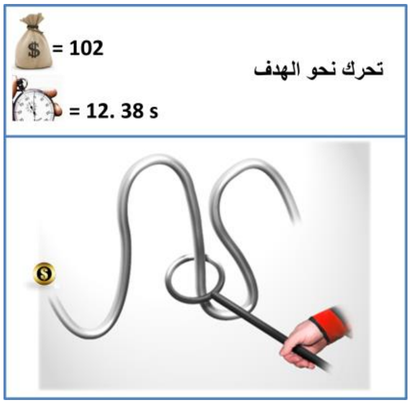


Figure 7.8: SHG Snapshot #1

7.3.2.1 Anatomy for Steady Hand Game Movements

A patient's arm movement from one position to another navigating a bar on x-y plane without make collisions with the bar edges contains:

1. Movements for the elbow and shoulder joints
2. Controlling for the muscles.
3. Eye-hand coordination with feedback.
4. Concentration Skills
5. Cognitive training.
6. Movement's plasticity improvement.

Steady Hand Game can be enhanced more to get:

7. Memory Skills.

7.3.3 Falling Parachutes Game (FPG)

Falling Parachutes Game, is the third chosen game one game. Which is the second game between two reaching targets games, which similar based on the same basics of the Collect Money Game (CMG). Where a number of parachutes (Targets) falling from the sky in random positions and times, and the patient (User) must move his/her arm in order to reach the falling parachutes and shoot them by a some rocket to make parachutes exploded in the sky before the explosion on the earth. See Figure 7.9.

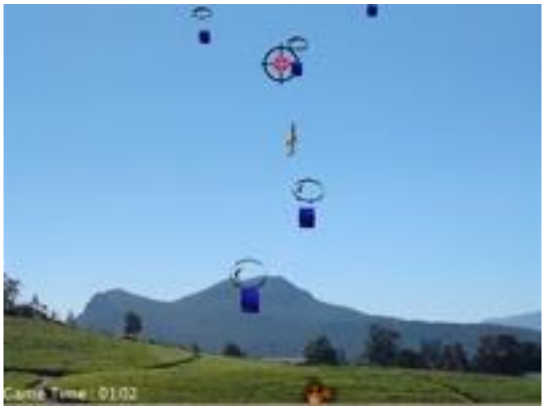


Figure 7.9: FPG Snapshot #1

Falling Parachutes Game similar to the collect money game in the positioning of targets methodology, because both in the same category of therapeutic computer games, where the game starts for some fixed time, and the parachutes starts to fall down from the sky from different locations.

However, this game is better motivated than the collect money game for the child patient's, because it takes the action and war impression. In addition, this type of games needs more concentration and effects more in the stimulating of the concentration skills, so it may make some positive effect on the whole cognitive effects.

7.3.4 Memorize Path Game (MPG)

Memorize Path Game, the fourth game in therapeutic exercise, which is one game of the following paths games which similar based on the same basics of the Steady Hand Game SHG. Where a patient (User) must memorize some sequence of places and then move following the moving path of this sequence in the same order that sequence showed at the beginning of the game. However, this game may help to improve the memory skills for the stroke patients. See Figure 7.10.

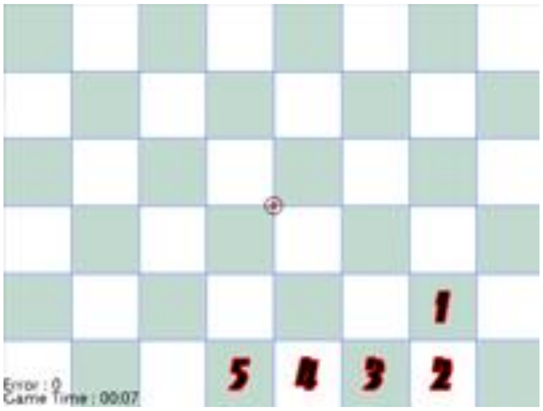

Error : 0
Game Time : 00:07

Figure 7.10: MPG Snapshot #1

7.3.5 Chosen Therapeutic Session Description

Stroke patients are typically seen for one or two half-hour sessions per day [45]. However, as a conclusion from a discussion with the physiotherapist Dr. Akram Amro [44], one session per a day with thirty minutes, where the patient can choose between four games, where he can play the second or the fourth game for three times for example approximately in seven minutes. Moreover, the physiotherapist in the first and third game can control the game time before it starts, and he/she must control the exercise session time manually.

7.3.6 Evaluation Methodology

Section C of Rivermead Motor Assessment is a therapist tool that is used in order to evaluate the patient's therapeutic case by asking the patient to do fifteen simple arm exercises, and evaluate each movements independently [46]. For example, clinical measures questionnaires, psychological measures questionnaires and usability measures questionnaire [39] which will be stored on the patient's profile yet. The second methodology is the progress reports that records the therapeutic case's parameters while the patient playing the therapeutic games such as elbow and shoulder speed of movement [47]. And another evaluation methodology (Third Methodology) for the patient's therapeutic case is the games' results and the progress of these results [39]. For example in Steady Hand Game (SHG) the performance evaluated by another progress report that records the number of collisions and the percentage of valid points, number of points located on the ideal path and completion time [41].

7.3.7 Game's Motivation Discussion

Motivation factor for the patient is one of the most important factors in the therapeutic computer games design, which can be enhanced by adding more and more games and exercises, and more updates for a game design itself.

It is important for the patient that the performed tasks are fun and meaningful in order to get his/her attention and desire to the therapeutic exercise. However, lack of motivation has been identified as a major cause of failure to neuro-rehabilitation process [48].

7.3.8 Cognitive Effects

TCGS, requires some cognitive skills in order to improve the patient's cognitive aspects. That arises due to the need of full assessment of the stroke patient's cognitive strengths and weaknesses when undergoing rehabilitation [49], that in order to use the suitable methodologies to improve the patient's cognition.

Cognitive abilities after a stroke could weaken in more than one aspect, may be in attention and concentration, or in memory, or in perception, or in the mental capacity, and also in other aspects like spatial awareness as neglect, and the executive functioning, and dysphasia and impairment of language [50].

However, one of physiological theories describes the cognitive processes by: "attention, perception and memory are identified as different cognitive activities that are involved in identifiable aspects of cognitive behavior" [51]. Those three aspects will be the main direct aspects of this study. But, the mental capacity, spatial awareness, executive functioning, and impairment of language will be excluded in TCGS design.

Cognitive processes include attention, mentally focusing on some stimulus; perception, interpreting and analyzing and organizing sensory information to yield meaningful information and store them in the memory which the storage facilities and retrieval processes of cognition [52].

A CMG focuses on the usage of the attention and reaction cognitive skills by the way. In addition, the SHG requires the usage of the attention and concentration skills in order to avoid the collisions with the bar's edges, and requires some of perception skills that are used to choose the correct path and through the whole bar to get the target.

7.4 Database Management Subsystem (DBMS)

The Entity-Relationship-Model for the software system DBMS consists of eight different entities, and each entity has some attributes, and there are some relations between entities each others. However, the next relational model describes the DBMS for the software system. See Figure 7.11.

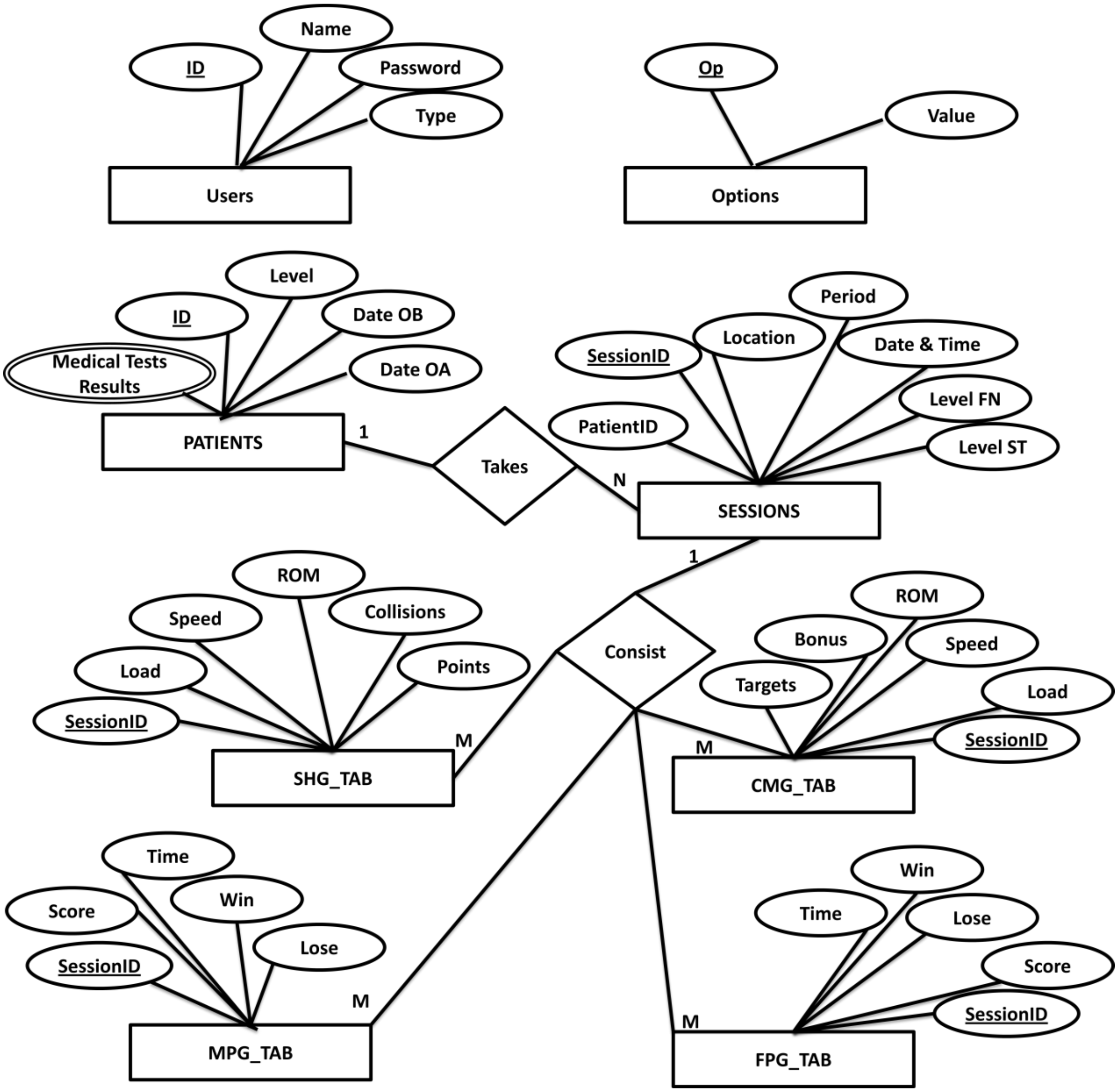


Figure 7.11: Entity-Relationship Model for DBMS

Those eight entities describe the whole DBMS System which stores the data into some different tables. A patient has some attributes that describes his/her therapeutic profile, he/she may make a session in a day, where each session can consist of four types of (CMG, SHG, FPG and MPG). However, another two entities that describes the user itself which is maybe patient or physiotherapist or administrator, and another entity is the option entity where any option has one operator and one value.

The patient's information, sessions' descriptions and exercises' data stored on a database file by the Database Management Subsystem Software. In order to achieve multiuser system and therapeutic progress tracing. Microsoft SQL Server Databases is one of the popular database systems, it is fast as required while the main purpose of the database management subsystem is to provide quick access to the data [53]. In addition, Microsoft DBAs run more mission critical databases, when compared to Oracle DBAs [54].

PATIENTS table stores information about the condition of the patient such as patient's ID, name, date of birth, date of accident, and stores the results of the medical tests before starting the virtual rehabilitation process.

The SESSIONS table contains information about a therapeutic session such as start date and time, location, period, and the start level and finished level evaluation before the session itself. And a new entry in this table will be created for every day and every session where the patient login the system and starts to play games.

CMG_TAB, SHG_TAB FPG_TAB and MPG_TAB store the played games' results and parameters, such as number of reached targets and bonus targets in CMG. In addition, the number of collisions and collected points in SHG. Moreover, stores and retrieves the maximum and minimum angles and the movement speed and maximum load that describes the muscle strength. And stores the number of win and lose points in FPG, and stores the total completion time and the total error time in MPG.

OPTIONS table have fixed number of rows, where each row describe some global option for the whole system, for example the COMPORT name and address stored in the value part for the COMPORT operation (OP).

Finally, the USERS table stores three types of users, each with his/her privileges and type, where each user have a username and user password and he/she can login into his/her profile on the system and access the recourses that he/she allowed to use and access them. However, the three types of users are PATIENT, PHYSIOTHERAPIST and ADMINISTRATOR.

7.5 Real Time Controlling Subsystem Software (RTCSS)

This part of software system works as the core of the whole project, which represents the software interface between external systems and internal subsystems. It is responsible to send the control signals to the break system, read the values from sensors system and sending and receiving parameters between E-GUI, DBMS, and TCGS each other.

RTCSS consists of six subparts as shown in Figure 7.12. Those subparts works simultaneously in real time to achieve the synchronization feature that gives the computer game the ability to be enjoyable and motivated.

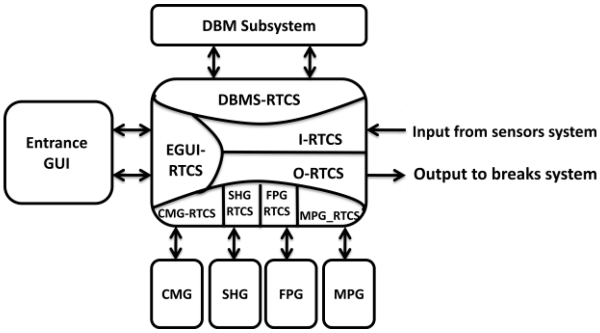


Figure 7.12: RTCSS Block Diagram

The first subpart of RTCSS is the Inputs Real-Time Controlling Software (I-RTCS). It is responsible to read and analyze the input values that come from the sensors system, in order to determine the location of patient's arm, in order to send the coordination of the patient's arm to the working computer games.

Another subpart is the Outputs Real-Time Controlling Software (O-RTCS), which responsible to send the suitable control signals to the break system. That controls the incensing and decreasing of the technical arm resistance as needs in the computer games both in entrance game and treatment computer game.

Database Management Subsystem Controller (DBMS-RTCS) is the third subpart of RTCS, which sends SQL Queries to DBMS, and retrieve the data that stored in it, in order to get a continues patient's treatment profile. The forth subpart of RTCSS is the Entrance Graphical User Interface Real-Time Controlling Software (ESG-RTCS), which sending and receiving the parameters from the DBMS and the Graphical User Interface on the LCD Screen.

And the last four subparts, are the Therapeutic Computer Games Software Real-Time Controlling Software (for CMG-RTCS, FPG-RTCS, MPG-RTCS and SHG-RTCS). They are determine the location of new targets that will be added on the games, and read location of the patient arm depending on the sensors system inputs from I-RTCS. Moreover, they are read the load value that required in the game and sends the suitable control signals to the break system through O-RTCS. In addition they are record and retrieve the information and data to and from the DBMS through DBMS-RTCS.

CHAPTER 8

Software System Implementation

8.1 Introduction

The next stage after analyzing and designing the software system is to implement the whole software system using the appropriate environment, and the suited programming language and programming tools and frameworks. Which all of that described in the previous chapter.

This chapter contains some topics about the implementation part for the software system, starts with the programming language and the implementation technique that used in the implementation. In addition, the implementation for the DBMS and the normalized model for the stored data. Moreover, some topics for the methodology to map between the screen space and device space, and between game's levels and device configuration.

Furthermore, this chapter contains a topic on serial communication implementation, and another section for evaluation methodologies implementation. In addition, it contains a section on project's forms and windows beside the user's permissions that represents the graphical user interface. Moreover, it is contains a section for some of implementation challenges that solved while implementation stage.

8.2 Spiral Development Model

Software System implementation methodology in this project is the Spiral Development Process, where the whole system's implementation divided into seven consecutive phases those are built on each other, begins at small nucleolus and starts to be bigger and bigger, and ends with full integrated system that represents the whole system software. Figure 8.1 see the used spiral lifecycle development process.

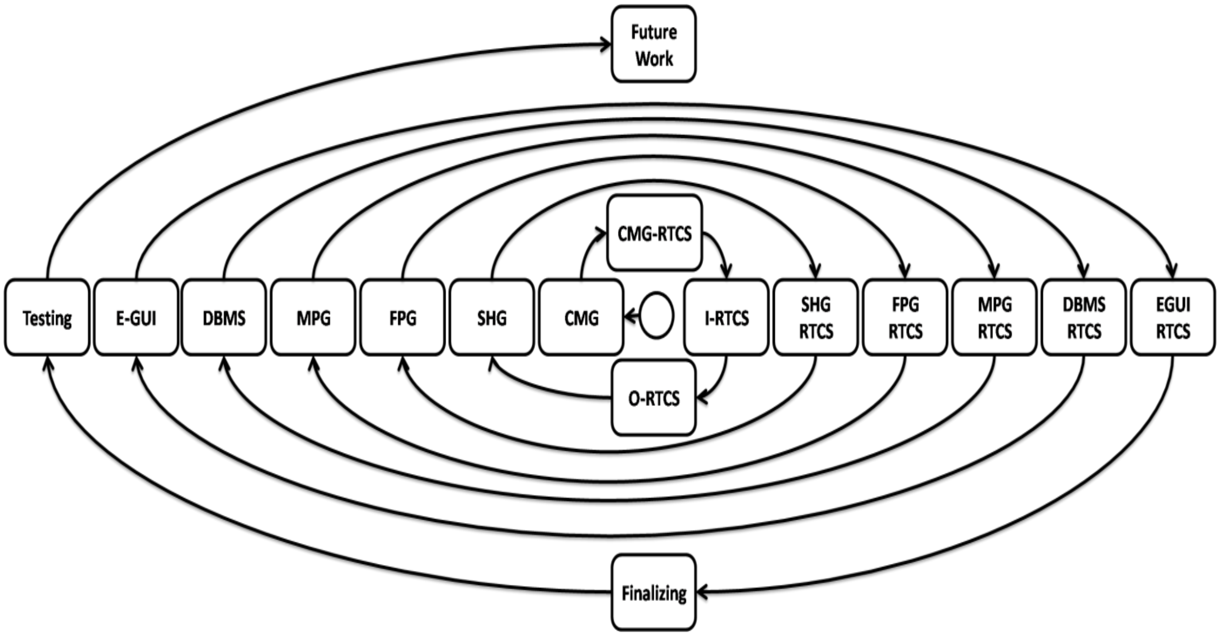


Figure 8.1: Spiral development model

Software System consists of fourteen subparts (EGUI, CMG, SHG, FPG, MPG, DBMS, I-RTCS, O-RTCS, EGUI-RTCS, CMG-RTCS, SHG-RTCS, FPG-RTCS, MPG-RTCS and DBMS-RTCS), as shown in Figure (7.14) in the previous chapter (Software System Chapter). Moreover, spiral development model get one of bottom-up approach advantages which that the implementation starts from small blocks, and go up, where the implementation process can be tested after each phase to verify that it in the correct way.

Initial phase consists of the construction and implementation and first therapeutic computer game (CMG), and the implementation of CMG-RTCS which controls the placement of targets in CMG. Moreover, it includes the implementation of the Inputs-RTCS and Output-RTCS, to create an initial integrated system that can considers as the nucleolus of Software System.

Phase II tasks describe as: implementing the Steady Hand Game, yet, implementation of the Real-Time Controller for the SHG, and add those two parts to the Software System nucleolus, that in order to extend the system by adding another one of the Therapeutic Computer Games.

Phase III consists of implementation of the Falling Parachutes Game, and the implementation of the real-time controller system for this game, and it contains adding those two parts to the total project nucleolus, and trying to run all of the three games together with other system's parts (I-RTCS and O-RTCS).

The fourth phase consists of implementation for the fourth game which is Memorize Path Game, and it contains the implementation of the real time controller for this game too, and it contains the process of adding the total worked part in this phase to the whole system's nucleolus.

Fifth phase consists of more additions to extend the Software System, which includes the implementation of the Database Management System using Microsoft SQL Server and the implementation of the DBMS Real-Time Controlling Software.

Sixth phase consists of implementation of the entrance menus and graphical user interface (E-GUI) that allows the system’s user to create, delete, update and select a patient profile. It is consists Entrance GUI RTCS which sending and receiving the parameters from the DBMS and EGUI-RTCS.

In the final phase, the final touches will be added, and the Software System will started the testing and evaluation, where the integrated project will be used by some patients with some physiotherapists, in order to repair the system’s bugs.

8.3 Implementation Scenario

As describes before in section 8.1, the implementation process divide into seven phases. However, the next seven sections will represent all those seven phases, by representing some related issues and some important notes about each phase implementation. Moreover, each phase description contains some figures that show the whole software system after finishing each phase (result of each phase)

During the implementation stage for the software system, there are many issues in code writing and graphical user interface design and implementation. However, totally the following seven subsections section talks about 21 of these issues and challenges.

8.4 Implementation of Phase 1

The first phase consists of four parts, which are: the implementation of the Collect Money Game, the implementation of real-time controlling for Collect Money Game, the implementation of input real-time controlling subsystem and the implementation of output real-time controlling subsystem yet.

The next subsections represent some important issues in the implementation of the first implementation phase, which are:

1. Change the resolution of the XNA screen.
2. Mapping between device space and screen space.
3. Serial communication implementation.
4. Display a text in 3D world.
5. Make multi screen game.
6. Usage of C# forms in XNA project.
7. Drawing 3D models and change the camera position.
8. Problem of collision detection between 3D models.
9. Make an unmovable C# forms.

8.4.1 Change the Resolution of the XNA Screen

The default resolution in XNA 4.0 is (800 pixel x 480 pixel). But, the game requires a changeable resolution that suits with the screen width and height. So, in order to change of the default resolution there is a need to change width and height parameters in the width and height variables in the graphics adapters, and those three lines of code in the (Initialize) function give the programmer the ability to access those variables directly without any problem :

```
graphics.PreferredBackBufferHeight = 600;
graphics.PreferredBackBufferWidth = 800;
graphics.ApplyChanges();
```

And in order to determine the resolution of the primary and secondary to make some conditions screen that connected with computer, the following code can be help to display the working area :

```
MessageBox.Show(Screen.PrimaryScreen.WorkingArea.Width + "");
MessageBox.Show(Screen.PrimaryScreen.WorkingArea.Height + "");
```

8.4.2 Mapping between Device Space and Screen Space

It is required after each sample reading that a converting from device workspace that shown in and the game's window space on screen. The device workspace is 800 millimeter width ranged between (-400 mm, 400 mm), and 400 millimeter height ranged between (350 mm-750 mm), where the zero point away from the workspace origin point of 550 mm on y-axis. However, the game's window is 800 pixel width, and 600 pixel height, and the zero point at the top left corner of the screen. See Figure 8.2. Suppose that $(\grave{x}, \grave{y})$ is the onscreen point, then :

$$\grave{x} = x + 400$$
$$\grave{y} = 600 - ((y - 350) * 1.5)$$

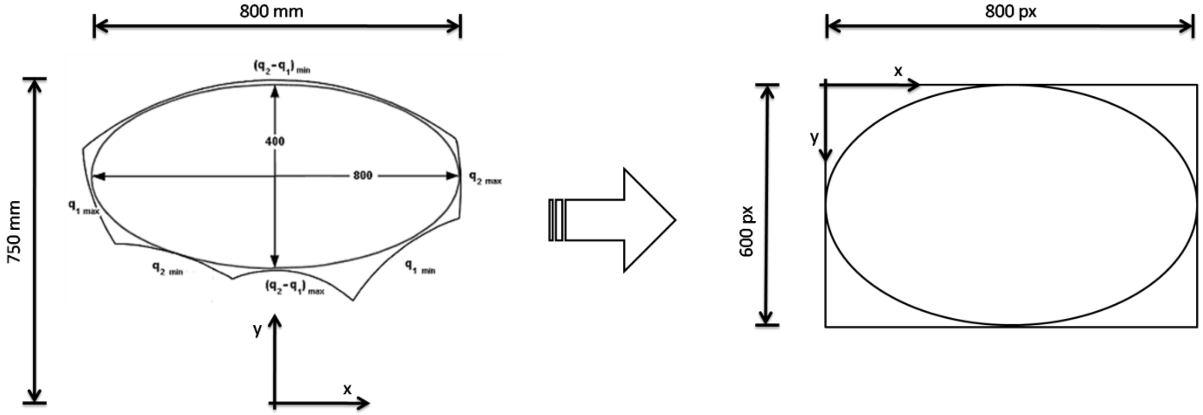


Figure 8.2: Device and screen workspaces

However, this code used for this mapping:

```
S.Write("r");
x3 = int.Parse(S.ReadLine());
x2 = int.Parse(S.ReadLine());
double Q2 = -0.2434 * x3 + 203.9;
double Q3 = 0.2765 * x2 - 27.45;
double Q4 = 36.5, L1 = 55, L2 = 18.7, L3 = 28.3;

double X = L1 * Math.Cos((Q2 / 180.0) * Math.PI)
         + L3 * Math.Cos(((Q2 + Q3 + Q4) / 180.0) * Math.PI)
         + L2 * Math.Cos(((Q2 + Q3) / 180.0) * Math.PI);
double Y = L1 * Math.Sin((Q2 / 180.0) * Math.PI)
         + L3 * Math.Sin(((Q2 + Q3 + Q4) / 180.0) * Math.PI)
         + L2 * Math.Sin(((Q2 + Q3) / 180.0) * Math.PI);

HandPosX = (X - 1) * -1.1;
HandPosY = (Y - 53) * -1;
```

8.4.3 Serial Communication Implementation

A serial communication session will be opened between software system and external controller (ARDUINO), that in order to receive inputs that come from sensors systems, and send outputs to break system, through XBee RF module. Serial connection established using COM port with a USB cable on RS232 standards.

Input and Output Real-Time Controlling Software can deal with reading and writing by C# code that below:

```
using System;
using System.Collections.Generic;
using System.IO.Ports;
namespace VirtualRehabilitation
{
    class SerialInterface
    {
        SerialPort SP
        int Serial_Rate = 9600
        public SerialInterface(String COM_Port
        {
            SP = new SerialPort(COM_Port, Serial_Rate
            SP.Open();
        }

        public void SerialRead(ref int Theta1, ref int Theta2,
                                   ref int Theta3)
        {
            SP.Write("r
            string Line = SP.ReadLine();
            int Delimiter1 = Line.IndexOf(":");
            int Delimiter2 = Line.LastIndexOf(":");
            Theta1 = int.Parse(Line.Substring(0, Delimiter1
            Theta2 = int.Parse(Line.Substring(Delimiter1 + 1,
                     Delimiter2)); //
            Theta3 = int.Parse(Line.Substring(Delimiter2));
            return;
        }

        public void SerialWrite(int Load1, int Load2)
        {
            SP.Write("w");
            SP.WriteLine(Load1 + ":" + Load2
        }
    }
}
```

8.4.4 Display a Text in 3D World

It is not allowed and not logical to draw a 2D text in 3D XNA world. So, in order to write a text, it must be a 3D text with three dimensions, that can be generated by draw the 2D text and extrude it, and then fill it with some color.

Nuclex XNA Tools is a free software tools that integrated with XNA framework to get advanced libraries and functions that help the programmers in order to improve their own game programming skills. Furthermore, Nuclex have some integrated functions that render the 3D text on the screen after extrude it. The following code shows subpart of the displaying of 3D text in 3D environment in the XNA games after making the integration between XNA framework with the Nuclex Tools :

```
SpriteBatch spriteBatch;
TextBatch textBatch;
VectorFont arialvector;

protected override void LoadContent()
{
    spriteBatch = new SpriteBatch(GraphicsDevice);
    textBatch = new TextBatch(GraphicsDevice);
    .
    .
    .
}

protected override void Draw(GameTime gameTime)
{
    .
    .
    .
    textBatch.Begin();
    string txt3d1 = "Hello";
    Text helloWorldText1 = arialvector.Fill(txt3d1);
    helloWorldText1 = arialvector.Extrude(txt3d1);
    textBatch.DrawText(helloWorldText1,
                   Matrix.CreateTranslation(
                   new Vector3(0, 25, 0)),
                   new Color(255, 255, 255, 255) );
    textBatch.End();
    .
    .
    .
}
```

8.4.5 Make Multi Screen Game

A XNA project provides only one screen for game programming, and in order to make multi screen game in Visual Studio environment there is a need for some static and global variable that represents a virtual screen id, where a project that contains more than one game, and some screens for user login and options must be a multi screen project.

Suppose that SCREENID is an integer, static, and global variable that represents some label for game screens, where a change from screen to another needs only to change the SCREENID variable.

In order to deal with multi screen project a change will occurred to the Draw function that contains the code of drawing models and objects on the screen. The following code shows the changes on the drawing function :

```
protected override void Draw(GameTime gameTime)
{

    GraphicsDevice.Clear(Color.CornflowerBlue);
    spriteBatch.Begin(SpriteSortMode.Immediate,
    BlendState.NonPremultiplied);

    if (screenId == 0)
    {
        // Some Code
    }
    else
    if (screenId == 1)
    {
        // Some Code
    }
    else
    if (screenId == 2)
    {
        // Some Code
    }
    else
    .
    .
    .

    spriteBatch.End();
    base.Draw(gameTime);
}
```

8.4.6 Finalizing of Phase 1

See Figure 8.3, Figure 8.4, where those snapshots token after finishing the implementation of the phase 1, and run the CMG and test it for many times.

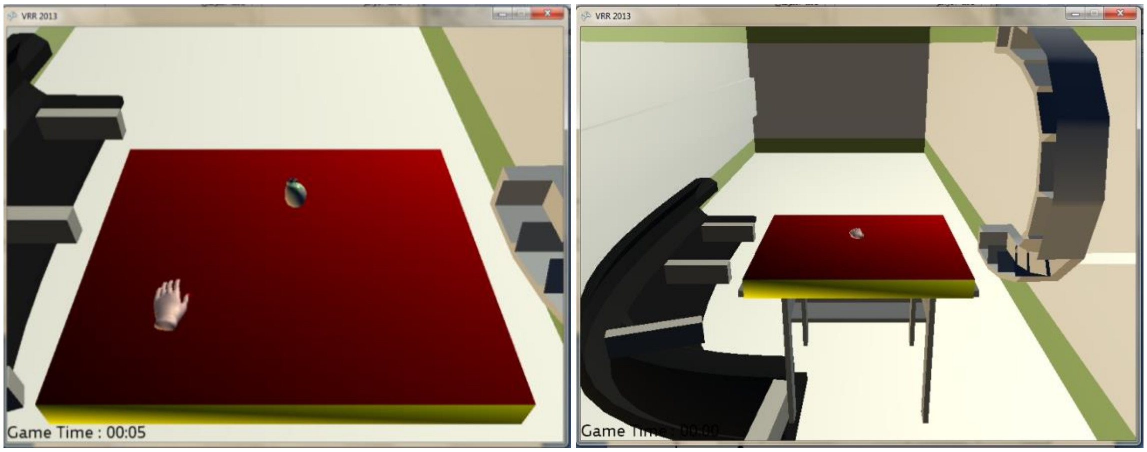


Figure 8.3: Collect Money Game Snapshots #1 #2

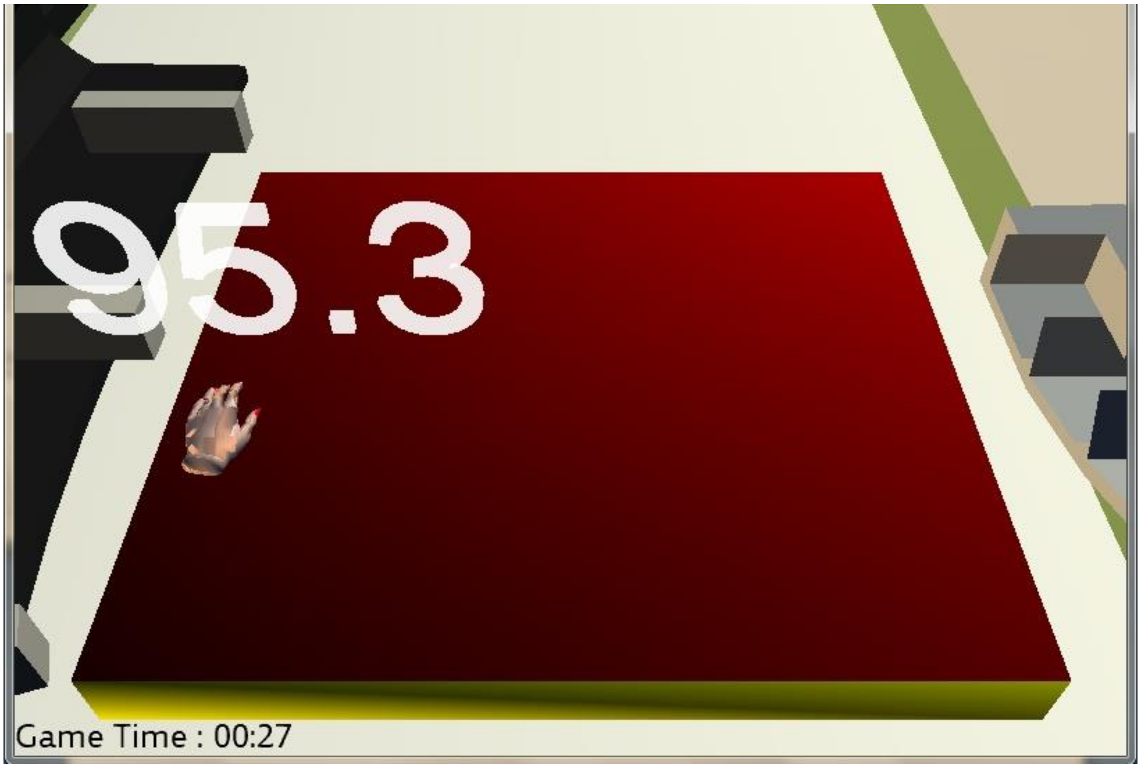


Figure 8.4: Collect Money Game Snapshot #3

8.5 Implementation of Phase 2

The second phase consists of two parts only, which are: the implementation of the Steady Hand Game and the implementation of real-time controlling for steady hand game. Furthermore, adding those two parts to the previous software system implementation phase that described before.

The next subsections represent some important issues in the implementation of the second implementation phase, which are:

1. Problem of drawing two 3D models over each other.
2. Rendering 3D sphere objects.
3. Drawing 2D background in 3D environment.
4. Problem of generating random path in SHG.

8.5.1 Problem of Drawing Two 3D Models Over Each Other

In 3D environment programming in XNA, there are a problem of rendering 3D models over each other, where the XNA renders the first model, then renders the second model on the same screen variables where the rendering of the second model shown above the first one, But there is an intersection area between both, where only the subsection of the second model shown on the screen.

And in order to try to solve this problem, a multi pass algorithm with alternative order of object rendering will be used, where the first model will be drawn first in the first pass, and the second model will be drawn first in the second pass, and then render the first and then the second and so on. Using the following code:

```
if (Pass++ % 2 == 0)
{
    DrawModel(Model1, World1, view1, projection1);
    DrawModel(Model2, World2, view2, projection2);
}
else
{
    DrawModel(Model2, World2, view2, projection2);
    DrawModel(Model1, World1, view1, projection1);
}
```

However, this solution not solves the problem but only it shows the first object and second object alternatively.

8.5.2 Rendering 3D Sphere Objects

It easy to create an 3D sphere in XNA, but there is no ready function that renders the sphere in order to show it on the screen. So, there is a need for building a function to make that:

```
static VertexBuffer vertBuffer;
static BasicEffect effect;
static int sphereResolution;
public static void InitializeGraphics(GraphicsDevice
                       graphicsDevice, int sphereResolution)
{
    Game1.sphereResolution = sphereResolution;
    effect = new BasicEffect(graphicsDevice);
    effect.LightingEnabled = false;
    effect.VertexColorEnabled = false;
    VertexPositionColor[] verts = new
              VertexPositionColor[(sphereResolution + 1) * 3];
    int index = 0;
    float step = MathHelper.TwoPi / (float)sphereResolution;

    //create the loop on the XY plane first
    for (float a = 0f; a <= MathHelper.TwoPi; a += step)
    {
        verts[index++] = new VertexPositionColor(
                    new Vector3((float)Math.Cos(a),
                    (float)Math.Sin(a), 0f), Color.White);
    }

    //next on the XZ plane
    for (float a = 0f; a <= MathHelper.TwoPi; a += step)
    {
        verts[index++] = new VertexPositionColor(
                    new Vector3((float)Math.Cos(a), 0f,
                    (float)Math.Sin(a)), Color.White);
    }

    //finally on the YZ plane
    for (float a = 0f; a <= MathHelper.TwoPi; a += step)
    {
        verts[index++] = new VertexPositionColor(
                    new Vector3(0f, (float)Math.Cos(a),
                    (float)Math.Sin(a)), Color.White);
    }
    vertBuffer = new VertexBuffer(graphicsDevice,
    typeof(VertexPositionColor), verts.Length, BufferUsage.None);
    vertBuffer.SetData(verts);
}
```

```
public static void Render(BoundingSphere sphere, GraphicsDevice
    graphicsDevice, Matrix view, Matrix projection, Color color)
{
    if (vertBuffer == null)
    InitializeGraphics(graphicsDevice, 30);
    graphicsDevice.SetVertexBuffer(vertBuffer);
    effect.World = Matrix.CreateScale(sphere.Radius) *
                   Matrix.CreateTranslation(sphere.Center);
    effect.View = view;
    effect.Projection = projection;
    effect.DiffuseColor = color.ToVector3();

    foreach (EffectPass pass in effect.CurrentTechnique.Passes)
    {
        pass.Apply();

        //render each circle individually
        graphicsDevice.DrawPrimitives(PrimitiveType.LineStrip,
                                      0,sphereResolution);
        graphicsDevice.DrawPrimitives(PrimitiveType.LineStrip,
                      sphereResolution + 1, sphereResolution);
        graphicsDevice.DrawPrimitives(PrimitiveType.LineStrip,
                      (sphereResolution + 1) *2, sphereResolution);
    }
}
```

8.5.3 Drawing 2D Background in 3D Environment

As known in XNA, a programmer can only make his/her computer game on 2D environment or on 3D environment, so it is not allowed to use a 2D components and textures in 3D environment game. But sometimes there is a need to use a 2D background and put it bellow all of the 3D models and components.

In order to solve this problem, you must open and close the 2D sprits batches before starting the 3D environment programming, like this:

```
spriteBatch.Begin(SpriteSortMode.Deferred, BlendState.AlphaBlend);
.
.
spriteBatch.Draw(game2_Back, rectBack01, Color.White);
.
.
spriteBatch.End();
```

8.5.4 Finalizing of Phase 2

See Figure 8.5, Figure 8.6, where those snapshots token after finishing the implementation of the phase 2, and run the SHG and test it for many times.

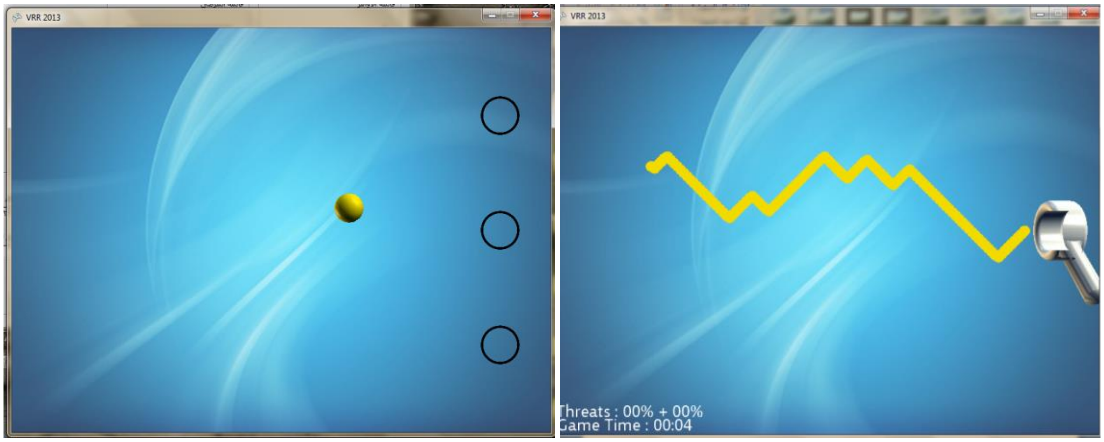


Figure 8.5: Steady Hand Game Snapshots #1 #2

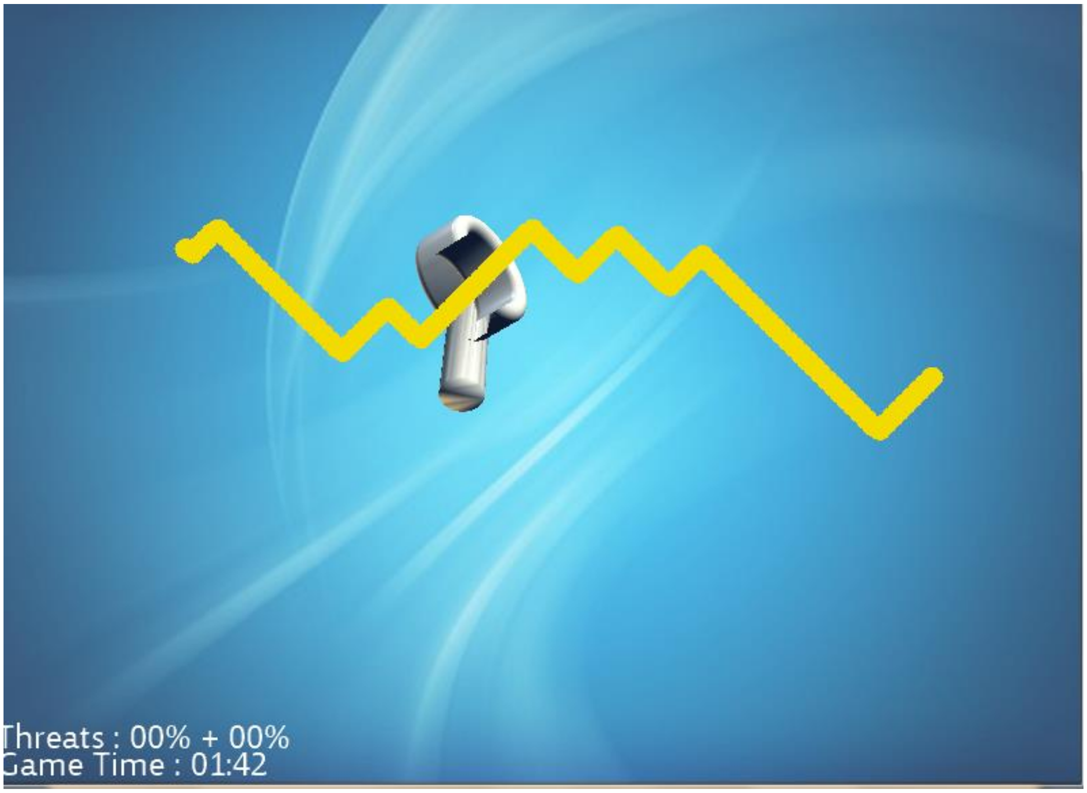


Figure 8.6: Steady Hand Game Snapshot #3

8.6 Implementation of Phase 3

The third phase consists of two parts too, which are: the implementation of the Falling Parachutes Game and the implementation of real-time controlling for Falling Parachutes Game, and add this game to the whole system.

The next subsections represent two important issues in the implementation of the third implementation phase, which are:

1.Converting 3D point in 3D environment into 2D point in 2D environment.

2. Making the explosion effect in FPG.

8.6.1 Making the Explosion Effect in FPG

In the falling parachutes game FPG, there is a need to make the explosion effect when the rocket make a collision with the falling parachute, where both models will be hidden and the collision effect only will be shown.

There is a way to make this effect using animated sprites by hiding and showing a series of pictures in a fast time. The reservation and loading the explosion effect by this way as bellow :

```
const int game3Size = 20;
classes.Explosion[] Explosions = new classes.Explosion[game3Size];
Texture2D t2dExplosionSheet;
Random generator;
protected override void LoadContent()
{
.
.
.
    generator = new Random();
    t2dExplosionSheet =
                Content.Load<Texture2D>("Anime\\Explosions");
    for (int i = 0; i < game3Size; i++)
    {
        Explosions[i] = new classes.Explosion(t2dExplosionSheet,
                         0, generator.Next(8) * 64, 64, 64, 16);
    }
.
.
.
}
```

And in order to initiate and start the drawing and moving for the explosion effect, the programmer must use the 'Active' function as bellow:

```
// Explosion Initialize
Explosions[i].Activate(X_Position, Y_Position,
                        new Vector2(X_Target, Y_Target),
                        Speed, Offset);
// Explosion Sound
game3Sound_But.Play();
```

However, in the drawing area and drawing functions, the programmer must put some piece of code in order to draw the explosion after initialize it:

```
for (int i = 0; i < game3Size; i++)
    Explosions[i].Draw(spriteBatch, true);
```

And this code for the other parts of the explosion class:

```
class Explosion
{
    static int iMapWidth = 1920;
    AnimatedSprite asSprite;     int iX = 0;    int iY = -100;
    bool bActive = true;     int iBackgroundOffset = 0;
    Vector2 v2motion = new Vector2(0f, 0f);
    float fSpeed = 1f;
    public int X
    {
        get { return iX; }        set { iX = value; }
    }
    public int Y
    {
        get { return iY; }        set { iY = value; }
    }
    public bool IsActive
    {
        get { return bActive; }
    }
    public int Offset
    {
        get { return iBackgroundOffset; }
        set { iBackgroundOffset = value; }
    }
    public float Speed
    {
        get { return fSpeed; }      set { fSpeed = value; }
    }
```

```
        public Vector2 Motion
        {
            get { return v2motion; } set { v2motion = value; }
        }

        public Explosion(Texture2D texture,
                    int X, int Y, int W, int H, int Frames)
        {
            asSprite = new AnimatedSprite(texture, X, Y, W, H, Frames);
            asSprite.FrameLength = 0.05f;
        }
        public void Activate(int x, int y, Vector2 motion,
                        float speed, int offset)
        {
            iX = x;
            iY = y;
            v2motion = motion;
            fSpeed = speed;
            iBackgroundOffset = offset;
            asSprite.Frame = 0;
            bActive = true;
        }

        private int GetDrawX()
        {
            int X = iX - iBackgroundOffset;
            if (X > iMapWidth)
                X -= iMapWidth;
            if (X < 0)
                X += iMapWidth;

            return X;
        }

        public void Update(GameTime gametime, int iOffset)
        {
            if (bActive)
            {
                iBackgroundOffset = iOffset;
                iX += (int)((float)v2motion.X * fSpeed);
                iY += (int)((float)v2motion.Y * fSpeed);
                asSprite.Update(gametime);
                if (asSprite.Frame >= 15)
                {
                    bActive = false;
                }
            }
        }
```

```
    public void Draw(SpriteBatch sb, bool bAbsolute)
    {
        if (bActive)
        {
            if (!bAbsolute)
                asSprite.Draw(sb, GetDrawX(), iY, false);
            else
                asSprite.Draw(sb, iX, iY, false);
        }
    }
}
```

8.6.2 Converting 3D Point in 3D Environment into 2D Point in 2D Environment

In falling parachutes game FPG, there is a need to convert the collision point in 3D environment to the corresponding point in the 2D environment in order to represent the collision event using some 2D texture (Explosion Texture).

It is hard and takes a lot of equations and processor time to determine the actual point after conversion. So, a function that separate the screen into subparts can give us an approximate result easily and quickly. The function bellow do that:

```
void convert3Dto2D(float X, float Y, ref int RX, ref int RY)
{
    float YY1;
    float XRange1;

    YY1 = (-0.00015565f * Y * Y * Y)
        + (0.038365f * Y * Y)
        + (-5.3341f * Y)
        + 300.05f;

    XRange1 = (-0.00000019885f * YY1 * YY1 * YY1)
        + (0.00030357f * YY1 * YY1)
        + (-0.2019f * YY1)
        + 93.48f;

    XRange1 *= 2;
    int XX1;
    XX1 = 400 + (int)(X * (800f / XRange1));

    RX = XX1;
    RY = (int)YY1;
}
```

8.6.3 Finalizing of Phase 3

See Figure 8.7, Figure 8.8, where those snapshots token after finishing the implementation of the phase 3, and run the FPG and test it for many times.

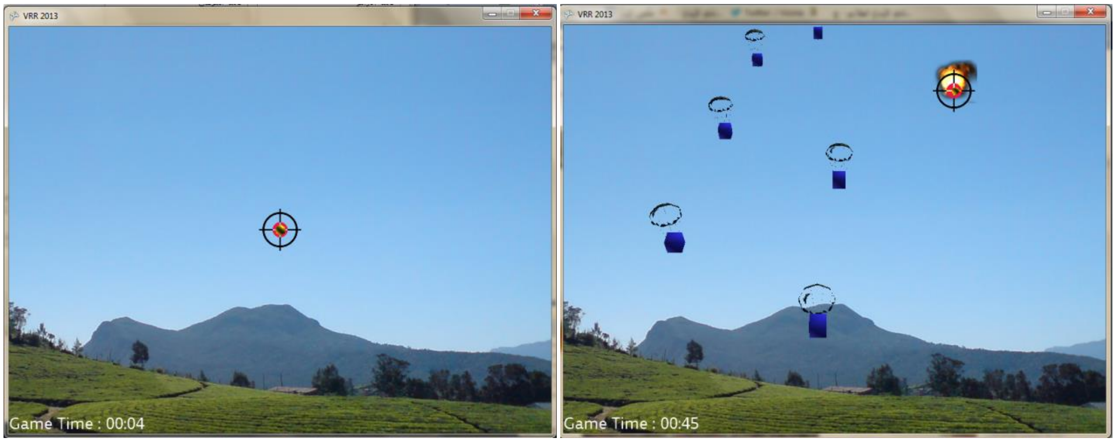


Figure 8.7: Falling Parachutes Game Snapshots #1 #2

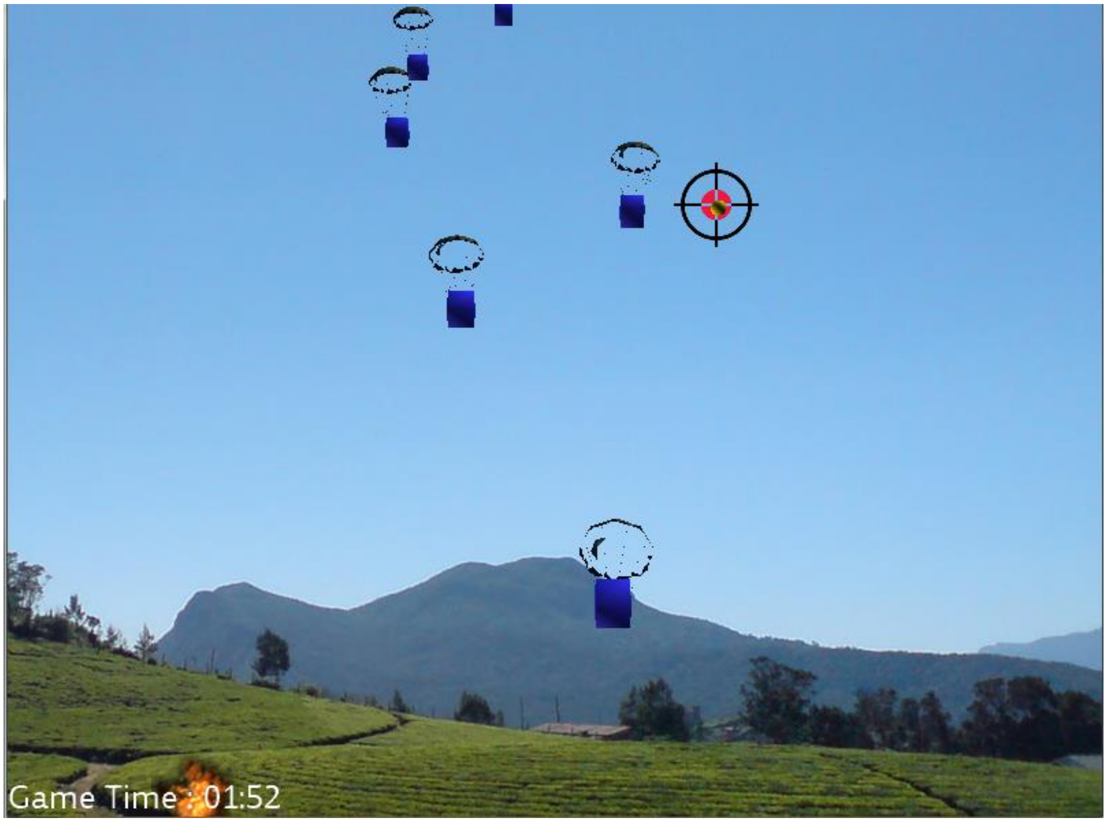
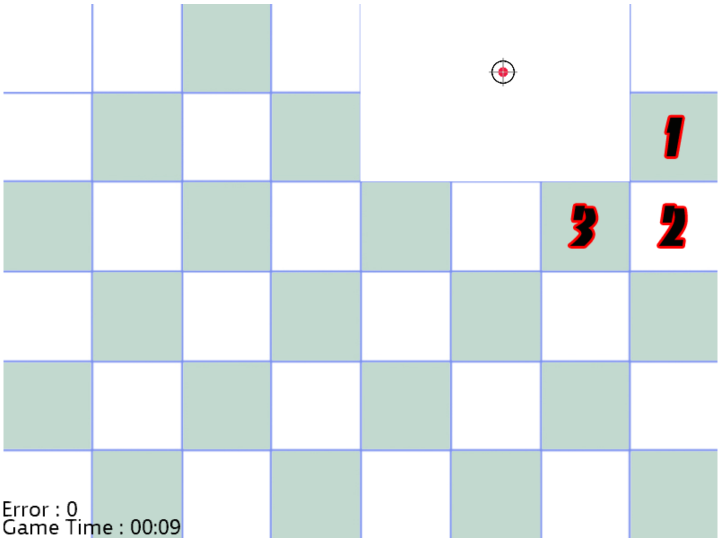


Figure 8.8: Falling Parachutes Game Snapshot #3

8.7 Implementation of Phase 4

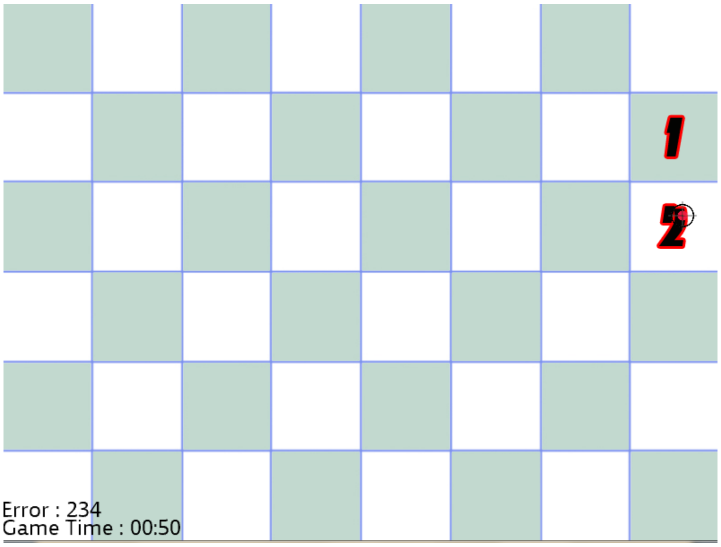
The fourth phase consists of two parts too, which are: the implementation of the Memorize Path Game and the implementation of real-time controlling for Memorize Path Game, and add this game to the whole system.

See Figure 8.9, Figure 8.10, where those snapshots token after finishing the implementation of the phase 4, and run the MPG and test it for many times.



Error : 0
Game Time : 00:09

Figure 8.9: Memorize Path Game Snapshots #1



Error : 234
Game Time : 00:50

Figure 8.10: Memorize Path Game Snapshots #2

8.8 Implementation of Phase 5

The fifth phase consists of building the whole database management system DBMS, and add it to the whole system nucleolus that built in the previous four phases.

After building the database itself on SQL Server, there is a need to connect it with the C# project using the standard libraries that connect both parts. This part of code on the C# side make that:

```
public partial class RTCSS : Form
{
    public static SqlConnection myConnection;
    public static SqlCommand myCommand;
    public static SqlDataReader myReader;

    public RTCSS()
    {
        InitializeComponent();

        myConnection = new SqlConnection("DataSource=.\\SQLEXPRESS;
        AttachDbFilename=|DataDirectory|\\DB1.mdf;
        Integrated Security=True;
        Connect Timeout=30;User
        Instance=True");
        .
        .
        myConnection.Open();
        myCommand = new SqlCommand("", myConnection);
        .
        .
        // Options !!
        frms.RTCSS.myCommand.CommandText = "select * from options";
        frms.RTCSS.myReader = frms.RTCSS.myCommand.ExecuteReader();
        while (frms.RTCSS.myReader.Read())
            {
            .
            .
            }
        frms.RTCSS.myReader.Close();
        .
        .
    }
.
.
}
```

8.8.1 Normalized Logical Design for DBMS

After adding more tables for the adding games, and more tables for game's options, and using both ER-to-Relational Mapping Algorithm and Normalization Algorithm [55], the results for conversion (Entity-Relationship Model for DBMS) that shown in Figure 7.14, from conceptual model to normalized logical model shown below, in Figure 8.11.

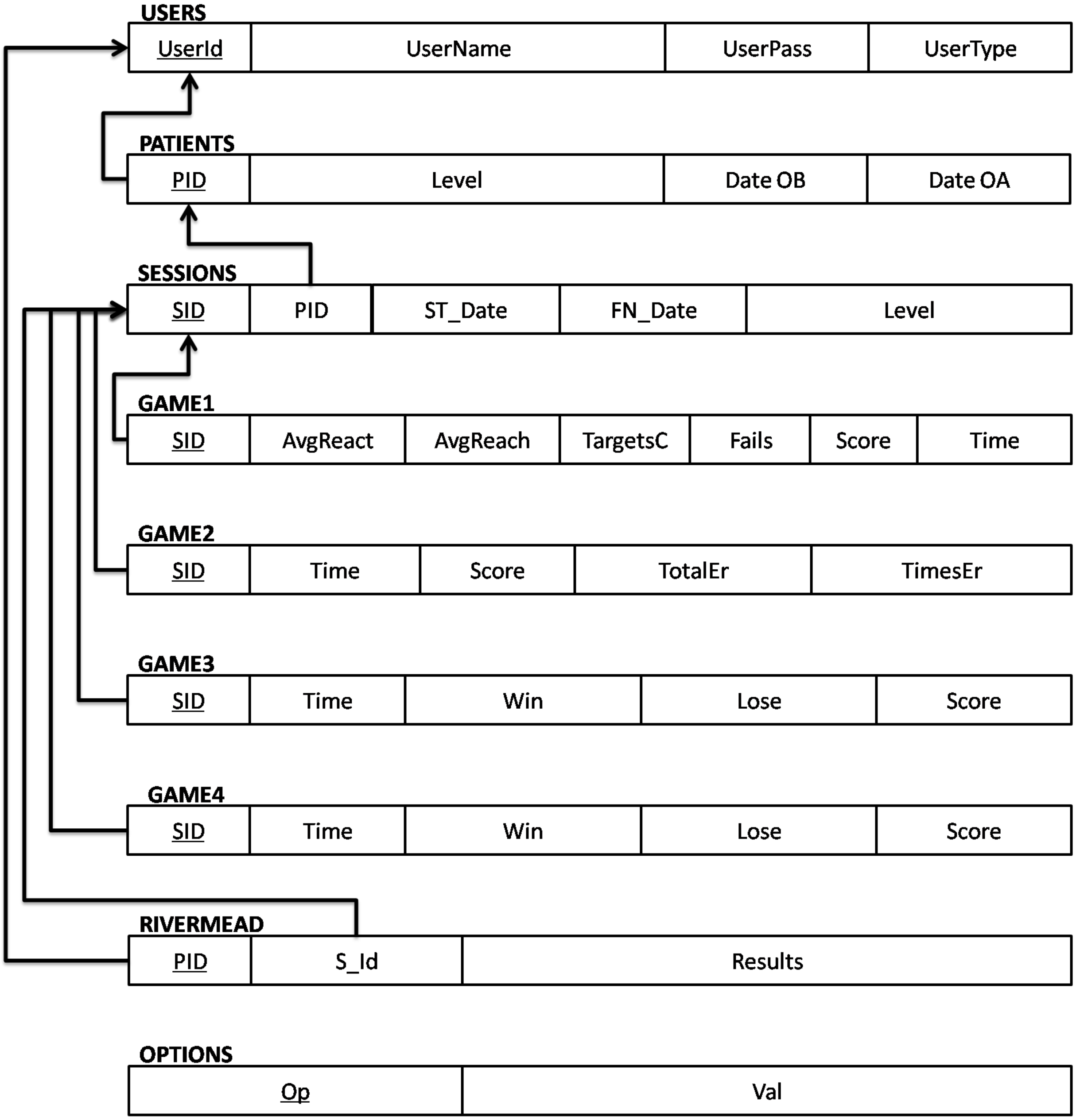


Figure 8.11: Normalized logical design for DBMS

8.9 Implementation of Phase 6

The sixth phase consists of making the graphical user interface and integrate it with the whole system's parts. Which consists of:

1. Challenge of Fade In - Fade Out Effects Using Alpha Blending: and this topic for improving the graphical user interface using some special functions and effects as fading out and fading in.
2. Project Windows and User's Permissions: where this topic list all of the project windows and forms and describe them
3.Primary-Secondary Screen Programming: where this topic talks about a new feature in this project which is using of two screens in display, where the primary screen for the physiotherapist or for the administrator and the other one for the patient. Where the patient cannot see anything in the physiotherapist side, and maybe he do not know that the (Virtual Rehabilitation Robot) have a computer in some hidden place.

8.9.1 Challenge of Fade In - Fade Out Effects Using Alpha Blending

In order to improve the graphical user interface effects, the usage fade in and fade out effects gives more motivation and more beautiful menus and screens. The fade in or fade out effects can be generated using Alpha Blending tools in XNA.

The first step to use the Alpha Blending tools it is initialize and begin the sprite batch that used to draw 2D backgrounds and 2D textures and other 2D components in XNA, and that can be done by this line of code:

```
spriteBatch.Begin(SpriteSortMode.Deferred, BlendState.AlphaBlend);
```

And in order to draw the backgrounds and textures after opening the sprite batch with the fade in effect, use some variable that starts from 0 to 256, and draw the 2D component assigning the alpha value to the variable value. See this code:

```
ApplicationTimer ++;

int mAlphaValue1 = ApplicationTimer > 256 ? 256 : ApplicationTimer;
spriteBatch.Draw(screen_Background, rectBack01,
                    new Color(255, 255, 255, mAlphaValue1));
```

However, in order to make the fade out effect you can only change the counter variable starting from 256 to 0.

8.9.2 Project Windows and User's Permissions

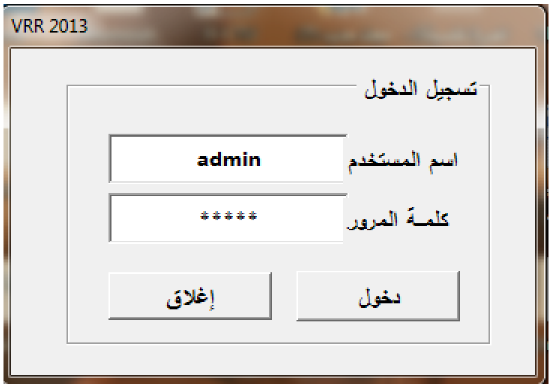
VRR2013 is the chosen name for the software application that represents the whole software system parts, which implemented by C# language on the Visual Studio 2010 environment with XNA 4.0 framework.

VRR2013 consists of 12 windows and forms which represent the graphical user interface that allow all users with all types of users to interact with the other system parts. VRR2013 consists of Login Form, Main Menu Window, and 9 C# forms for three types of users (Patient, Physiotherapist, Administrator).

8.9.2.1 Login Form

The login form is the startup window for VRR2013, where the user can enter his/her name and password, as shown in Figure 8.12. In order to enter the system and interact with other windows with his/her own Permissions.

After filling the textboxes and click the entrance button, a search query will executed in order to find the username and his/her password matching, and then to retrieve the user variables that stored in the database (i.e. User Type, User Id).


VRR 2013

تسجيل الدخول

اسم المستخدم: admin

كلمـة المرور: *****

Figure 8.12: Login Form

8.9.2.2 Main Menu Window

After entering the login form, a menu window will be shown which contains some choices that depend on the user's type and user's permissions;

I.e. patient's menu window contains four choices: new sessions, edit patient's data, show patient's parameters and log out, as shown in Figure 8.13.

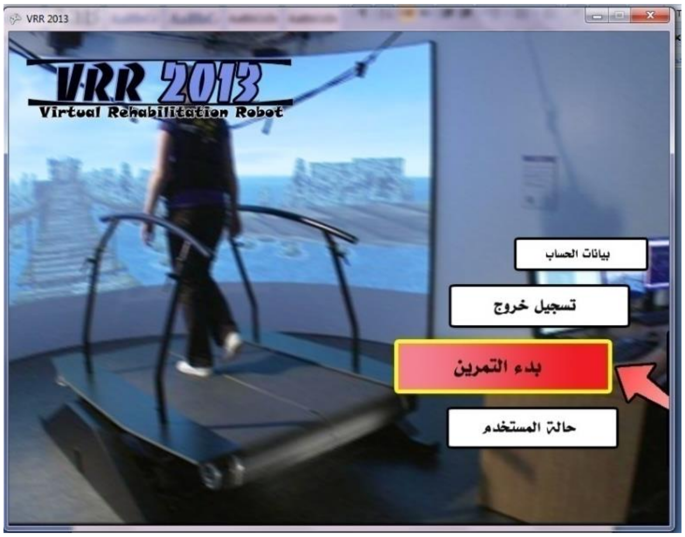
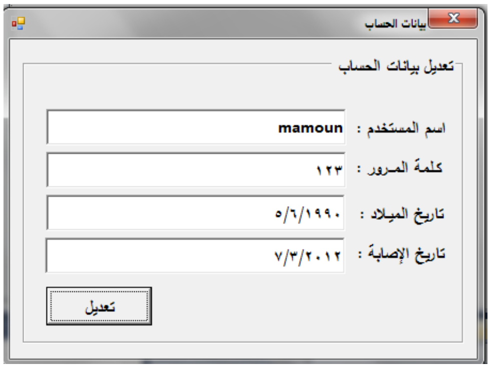


Figure 8.13: Main Menu Screenshot

8.9.2.3 Patient's Permissions Windows

A patient has the permissions and the ability to create and start a new therapeutic session, show and display the patient's profile and some therapeutic parameters, and edit the patient's name, password, date of birth and date of accident.

A patient can change his/her username both in Arabic or English languages, and can create or change his/her password, and edit both his/her date of birth and date of accident. Show Figure 8.14.



بيانات الحساب

تعديل بيانات الحساب

اسم المستخدم : mamoun

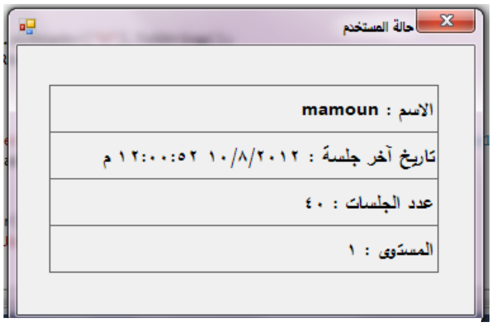
كلمة المـرور : ١٢٣

تاريخ الميـلاد : ٥/٦/١٩٩٠

تاريخ الإصابة : ٧/٣/٢٠١٢

Figure 8.14: 1st Patient's Form

After login in the system with his/her username and password, the patient can display his/her status parameters like the date of the latest exercise session on the VRR2013 system, and he/she can show the total number of sessions that have been finished on the system. Moreover, a patient can show his/her therapeutic level from one to ten. Look to the patient's status form in Figure 8.15.



حالة المستخدم

الاسم : mamoun

تاريخ آخر جلسة : ١٠/٨/٢٠١٢ ١٢:٠٠:٥٢ م

عدد الجلسات : ٤٠

المستوى : ١

Figure 8.15: 2st Patient's Form

The patient also can start a therapeutic exercise session by choosing the 'Start Session' choice, where a menu of four games will be displayed, and the patient using the keyboard arrows can choose the game that he/she want to play it first to do the exercise session. Look to the Figure 8.16.

In start session form, a patient only allowed to play a game between four games for a number of times that selected by the system administrator, for example the patient is allowed to play the second game for three times in the same session, where he/she cannot play this game for the forth.

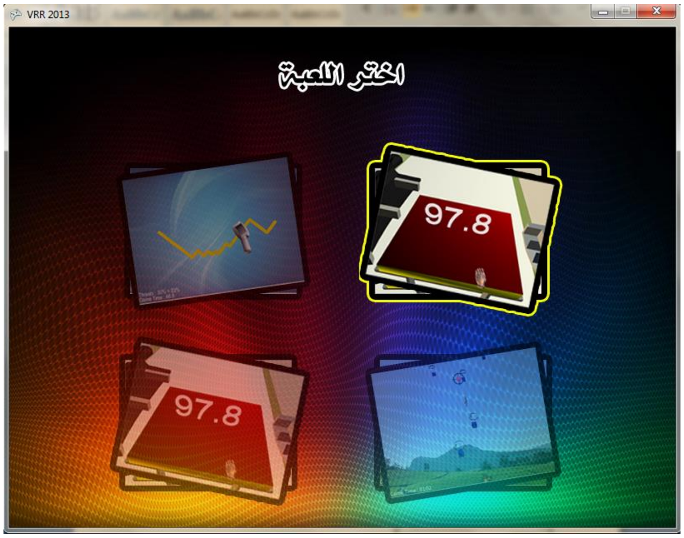
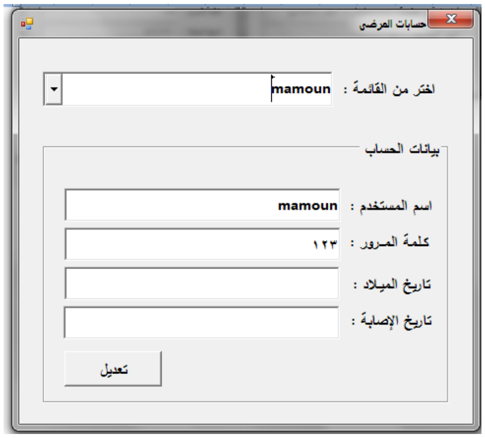


Figure 8.16: Start Session Form

8.9.2.4 Physiotherapist's Permissions Windows

A physiotherapist has four different windows, that represent his/her permissions on the software system. The first screen allows the physiotherapist to add a new patient profile, and it allows to edit any exist patient's profile, show Figure 8.17.

Moreover, a physiotherapist can edit his/her profile parameters by the second form. In addition, he/she can present the therapeutic reports for any patient using the third form. Figure 8.18 represents both forms.



اختر من القائمة : mamoun

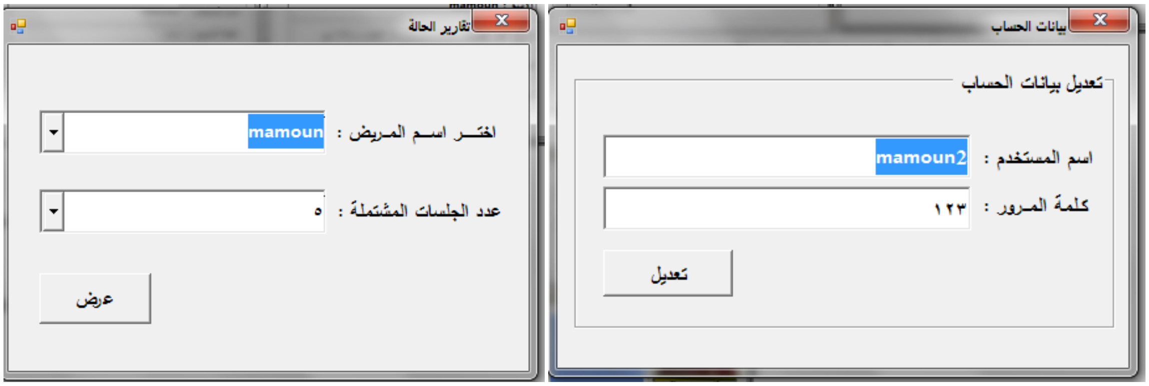
اسم المستخدم : mamoun

كلمة المـرور : ١٢٣

تاريخ الميـلاد :

تاريخ الإصابة :

Figure 8.17: Patients Accounts



اختـــر اسـم المـريض : mamoun

عدد الجلسات المشتملة : ٥

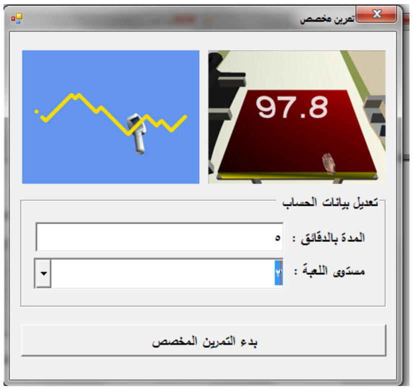
اسم المستخدم : mamoun2

كلمة المـرور : ١٢٣

Figure 8.18: Some Physiotherapist Forms

Furthermore, the forth form in the physiotherapist forms, allows the physiotherapist to run a special session of games by detecting the game itself, and detect the game's time, and the playing level. A physiotherapist can choose a game from two games, and choosing the playing level from ten levels, further he/she can choose the game's time in minutes. Show Figure 8.19.

A physiotherapist must choose the game's time, and playing level then click on the game's logo in order to start the selected game.



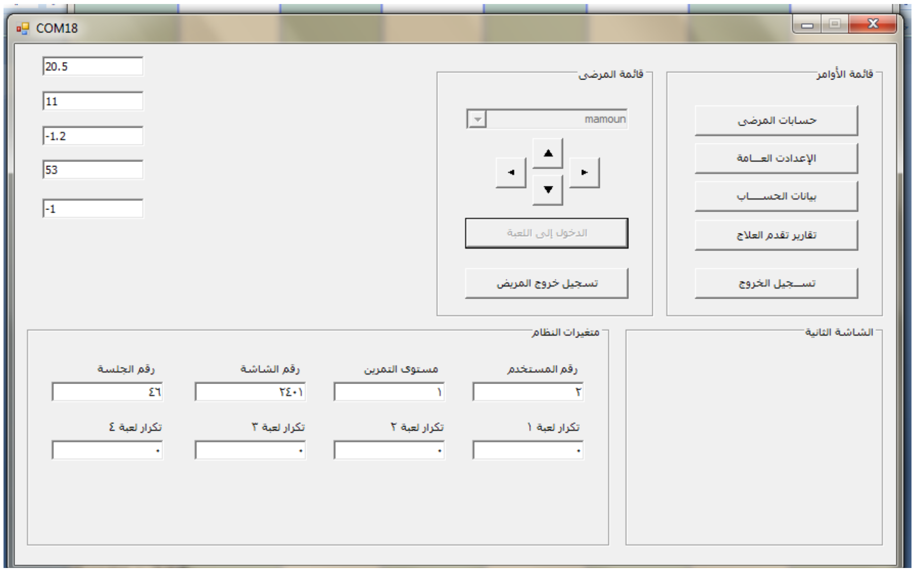
المدة بالدقائق : ٥

مستوى اللعبة : ٢

Figure 8.19: Special Session

8.9.3 Primary-Secondary Screen Programming

A new feature in this project which is using two screens for the display, where the primary screen is for the physiotherapist or for the administrator and the other one is for the patient. Where the patient cannot see anything in the physiotherapist side, and maybe he do not know that the Virtual Rehabilitation Robot have a computer in some hidden place, Figure 8.20 shows the primary screen and Figure 8.21 shows the secondary screen.



20.5
11
-1.2
53
-1

mamoun

رقم المستخدم	مستوى التمرين	رقم الشاشة	رقم الجلسة
٢	١	٢٤٠١	٤٦
تكرار لعبة ١	تكرار لعبة ٢	تكرار لعبة ٣	تكرار لعبة ٤
٠	٠	٠	٠

Figure 8.20: Primary Screen Snapshot

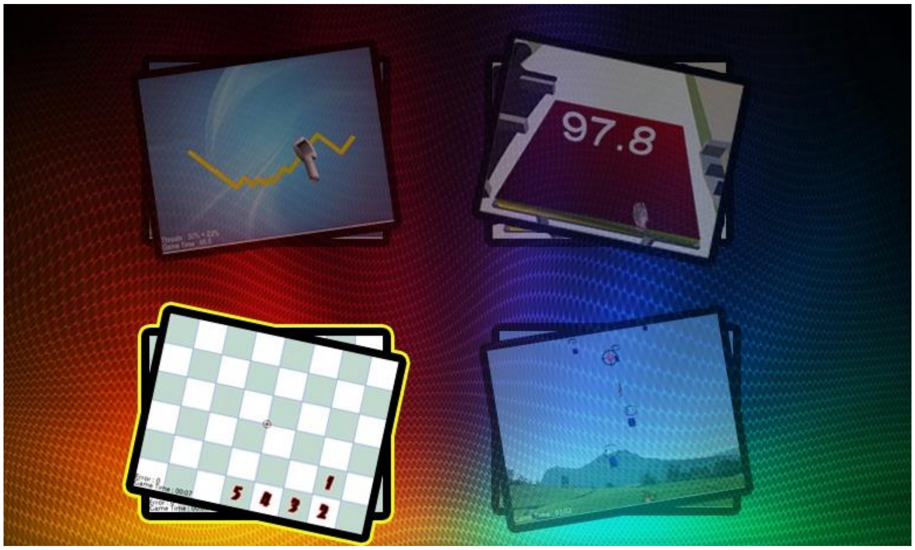


Figure 8.21: Secondary Screen Snapshot

8.10 Implementation of Phase 7

The seventh (last) phase consists of finalizing the whole system and test it. And while this phase there are some suggested ideas have been added, and there are some problems have been solved, such as:

1. Problem of auto-reaching target in CMG.
2.Usage of real stopwatch functions.
3.Moving out of allowed path in SHG.
4.Reaction time detection and calculation in CMG.

The seventh phase also contains the integration of the all project's parts and test the whole project and evaluate it. Moreover, the next four subsections describes the four problems that listed before.

8.10.1 Problem of Auto-Reaching Target in CMG

After making some tests on the collect money game (CMG), a problem has been detected, where some targets generated randomly on the same place that the user hand's position on the workspace, where the collision between the user's hand and the target happens automatically without any movement on the user's side.

So, there is a need to check the random position of the next target before place it on the screen, and ignore any random position that impacted with the hand position on the screen workspace. However, this code do that:

```
bool CorrectRandomValues = false;
while (!CorrectRandomValues)
{
    TargetPosX = generator.Next(-35, 35);
    TargetPosY = generator.Next(-30, 30);
    if (!new BoundingSphere(newVector3(
     (float)frms.RTCSS.HandPosX,
     (float)frms.RTCSS.HandPosY, 0f),
     4.5f).Intersects(new BoundingSphere(
    newVector3((float)TargetPosX,
     (float)TargetPosY, 0f), 2.5f)) )
        CorrectRandomValues = true;
}
```

8.10.2 Moving Out of Allowed Path in SHG

In steady hand game SHG, there is a need to ignore any movement of the user's arm out of the allowed path:

```
if (!game2Strt)
{
    game2X = frms.RTCSS.HandPosX;
    game2Y = frms.RTCSS.HandPosY;
}

gameOver2_1 = Math.Abs(game2DX);
gameOver2_2 = Math.Abs(game2Y  - frms.RTCSS.HandPosY);

if (game2Strt && !m && !isCol)
{
    gm2r.Add(new game2R(gm2Times,
                        stopwatch.ElapsedMilliseconds, 0));
    MediaPlayer.Play(BuzzGame2);
    MediaPlayer.Volume = 1;
    isCol = true;
    smallstopwatch.Reset();
    smallstopwatch.Start();
}

else if (!m)
{
    if (gameOver2_1 >= 4.0 || gameOver2_2 >= 20.0)
        gameOver = true;
    if (upOrDown)
    {
        if ((game2Y - game2DY) - frms.RTCSS.HandPosY > 0.1)
        {
            game2Y = frms.RTCSS.HandPosY - game2DY;
            m = true;
        }
        else
        {
            game2DY = game2Y - frms.RTCSS.HandPosY;
            game2DX = game2X - frms.RTCSS.HandPosX;
        }
    }
    else
    {
        if ((game2Y - game2DY) - frms.RTCSS.HandPosY < -0.1)
        {
            game2Y = frms.RTCSS.HandPosY + game2DY;
            m = true;
        }
        else
        {
            game2DY = game2Y - frms.RTCSS.HandPosY;
            game2DX = game2X - frms.RTCSS.HandPosX;
        }
    }
}
if (m)
{
    game2Y = frms.RTCSS.HandPosY + game2DY;
    if (isCol)
    {
        game2DX = game2X - frms.RTCSS.HandPosX;
        game2DY = game2Y - frms.RTCSS.HandPosY;
        gm2r[gm2Times].total = smallstopwatch.ElapsedMilliseconds;
        LostPoints += (int)gm2r[gm2Times].total;
        gm2Times++;
        MediaPlayer.Play(game2Song);
        MediaPlayer.Volume = 0.05f;
        isCol = false;
        smallstopwatch.Reset();
    }
    game2X = frms.RTCSS.HandPosX + game2DX;
}
```

8.10.3 Usage of Real Stopwatch Functions

There is a need to end some games after some part of time, and there is a need to calculate some score for some game that depends on the game time. So, there is a need to a way that a programmer in XNA can display and control the game time.

A XNA project has a global variable for whole project time. So, a programmer must create other timers that depend on the computer time, where he/she can control and display those timers when he/she needs.

A Stopwatch object is an C# object that gives the programmer all of that, where he/she can start, stop, resume, reset, restart and display the time in milliseconds, seconds and minutes. Where the declaration of this object needs this line of code:

```
System.Diagnostics.Stopwatch s = new System.Diagnostics.Stopwatch();
```

8.10.4 Reaction Time Detection and Calculation

In collect money game CMG, there is a need to calculate the reaction time of the patient's arm movement as one of the evaluation parameters that depends on the patient's case.

The reaction time is the time from the target is shown on the screen to the time that the patient start to move his/her arm in order to catch and reach the target. This value can be calculated for each target in the CMG, and finally the software system can store the average value of those reaction times

The following code for the start time of positioning a target on the screen:

```
.
.
.
else if (PlayingTimer++ == 200)
{
    strt = stopwatch.ElapsedMilliseconds;
    isReaction = true;
    reactionX = frms.RTCSS.HandPosX;
    reactionY = frms.RTCSS.HandPosY;
    reactionTime = 0;
    calcTheTime = true;
}
.
.
.
```

And the following code is for the end time where the system records the reaction time assuming that X is the threshold value of movement:

```
.
.
.
if (isReaction && (Math.Abs(reactionX - frms.RTCSS.HandPosX) > X
        || Math.Abs(reactionY - frms.RTCSS.HandPosY) > X))
{
    isReaction = false;
    reactionTime = stopwatch.ElapsedMilliseconds;
}
.
.
.
```

CHAPTER 9

Experimental Results

9.1 Introduction

This chapter provides the experimental results for the mechanical system, the sensors and interfacing system, the load system and the computer system as separate parts. After that, the whole device is tested as a complete system.

9.2 Mechanical System

The theoretical calculations are done in chapter four. These calculations determine the dimensions of the mechanical device and the chosen material. They consider also the strength of the structure and the smoothness of the motion without a large deflection or joints locking.

However, these calculations need to be checked on real life. So, this section shows the mechanical experiments on the device related to the strength, deflection, bearing, configuration, flexibility and finally, the workspace test.

9.2.1 Strength Test

This section is devoted for the strength of mechanical device. It is important to guarantee that the device is rigid and will not fail or destroy below the rated loads. The device is built based on the same specifications for the theoretical results.

The experiment is done by a normal person whom uses the device and performs the exercises. The experimental results of the mechanical device proof that theoretical calculations are correct and that the device can withstand with normal operation that is designed for. In addition, the total weight of the parallelogram is not large, even when the experiments are in the vertical position where the patient carries most of the end-effector weight.

9.2.2 Deflection Test

The importance of the deflection test is to guarantee the device from joints locking that happens due to the joint movements. The theoretical deflection calculations consider the parallelogram with its dimensions and the used material as a good with a very small deflection that does not considered in real life.

The experiment is done by a normal person who drives the manipulator from one position to another performing the exercises. The result is that no measurable deflection is found at the end-effector.

9.2.3 Bearings Test

This section is needed to test the smoothness of the device. The theoretical calculations that are done in section (4.2.3) chose the required bearings to make the device movement smooth. The experiment is done by moving the end-effector from one position to another while checking the smoothness of the trajectory. The result shows that the friction is very small and that anyone can move it like moving a small mass with tiers on a horizontal plane without the feeling of the end-effector weight.

9.2.4 Configuration Test

The mechanical device is built based on many features. One of them is the configuration for changing the position (horizontal, vertical and inclined position). The other feature is the flexibility which allows the mechanical device to be suitable for each patient size and height.

The theoretical calculations show the specifications of the configuration joint and the vertical shaft respectively. The experimental test shows that the device can be moved and locked by the loosen/tighten of the configuration joint respectively. The experiments on the vertical and the configuration shafts are done in an easy way. Furthermore, the device is suitable for all user sizes and satisfies the three positions (horizontal, vertical and inclined).

9.2.5 Workspace Test

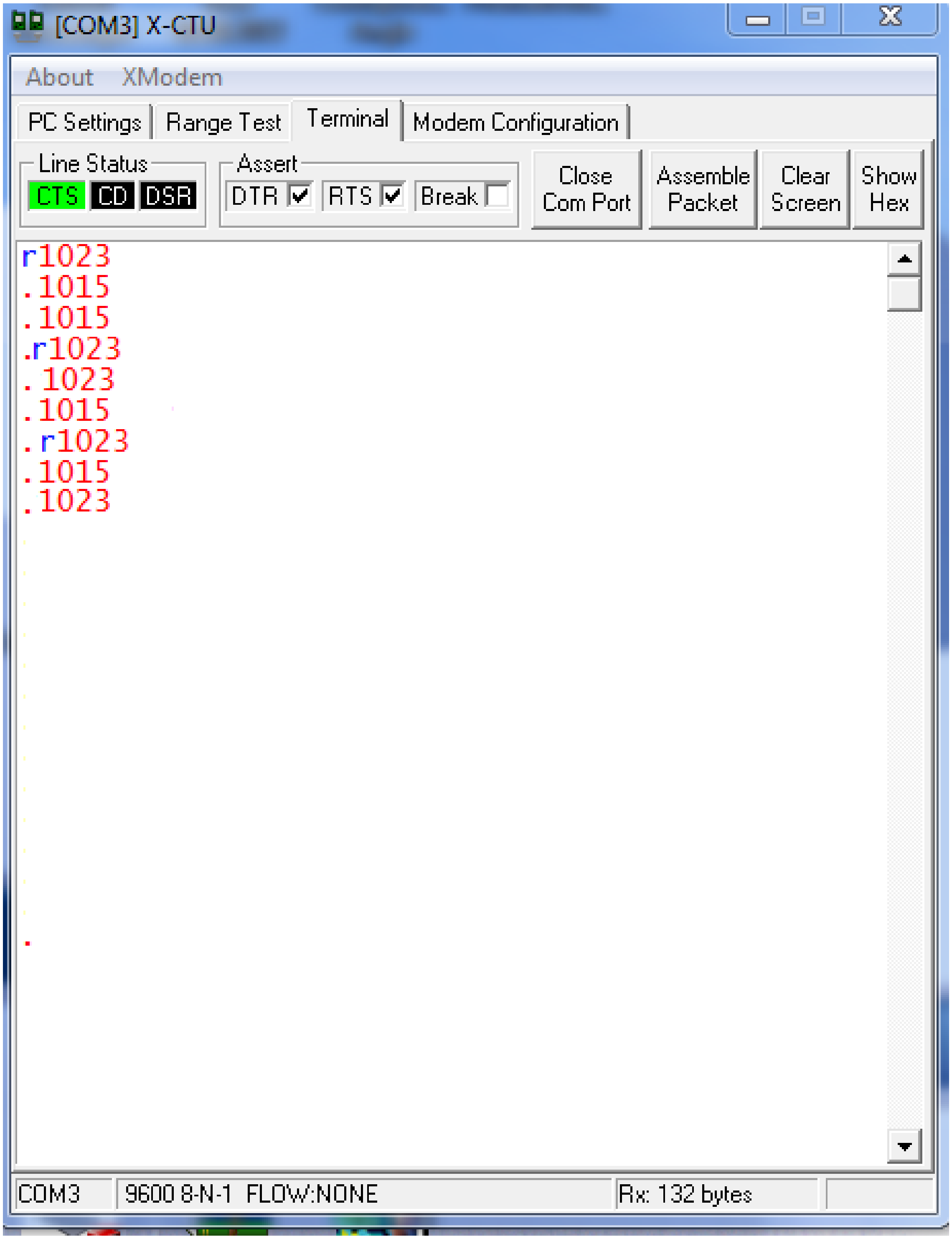
The workspace for the manipulator is 400 mm X 800 mm based on the open source Braccio di Ferro design which is suitable for most patients. The workspace test is important to know if the required workspace is achieved or not. After measuring the maximum values for the workspace of the parallelogram it is proofed that the required workspace is accomplished. Furthermore, the manipulator in this mechanical design can satisfy a larger workspace 500 mm X 900 mm.

9.2.6 Damper Test

This test is devoted for the smoothness of the damper and the ability to change the load by hand. The test is done by rotating the screw of the variable orifice damper by hand and notice the change in the load.

9.3 Communication System Testing

The interfacing between the real workspace and the computer must be symmetrical and synchronized. After connecting the three potentiometers with ARDUINO and transmit their values through the XBee, these values are read from the computer through interface port as shown Figure 9.1. When the patient moves the end-effector from one point to another, the curser on the screen moves in the same direction.



DTR ☑ RTS ☑ Break □

```
r1023
.1015
.1015
.r1023
. 1023
.1015
. r1023
.1015
.1023
.
```

COM3 | 9600 8-N-1 FLOW:NONE | Rx: 132 bytes

Figure 9.1: The interfacing port for reading the three potentiometers

9.4 Software System Testing

Using the black-box testing methodology, the software system had been tested. Black-box is a method of software testing that tests the functionality of an application as opposed to its internal structures or workings. This method of test can be applied to all levels of software testing: unit, integration, system and acceptance. It typically comprises most if not all testing at higher levels, but can also dominate unit testing as well.

Software system testing depends on the spiral development model which divided the software system into seven phases as described in CHP8. Those seven phases are seven programming units.

Each phase tested with black-box testing methodology as unit test, and each added phase on previous phase tested as unit test, and so on, that from the first unit to the last unit. After that the whole software system tested as integrated system (integration test). That means the testing process comes after finishing a phase and before starting the next phase.

A programmer says that the testing result is good results if he/she can transient from step to another, where starting the third three for example means that the second phase got good results in testing, and so on. However, all of that is one of the most advantages of the spiral development model. Figure 9.2 show that the four games after running successfully.

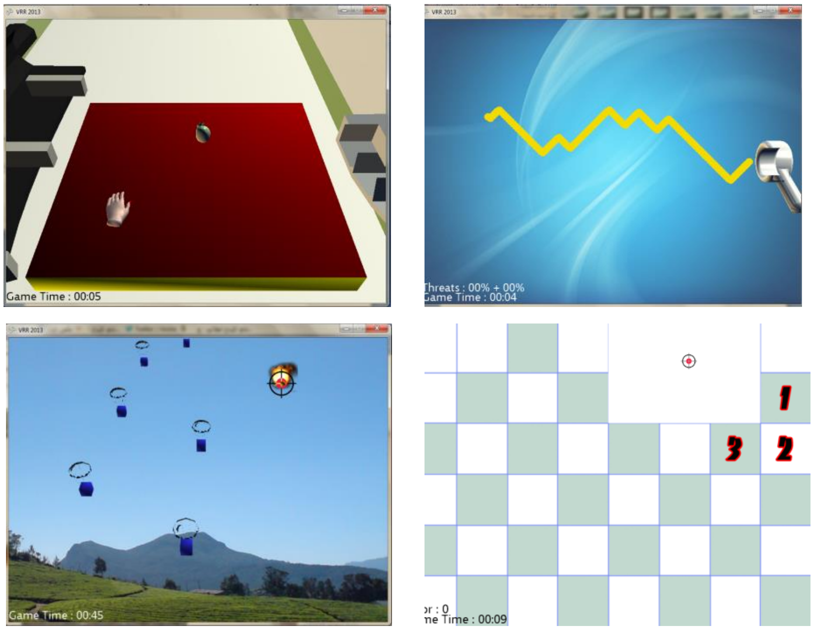
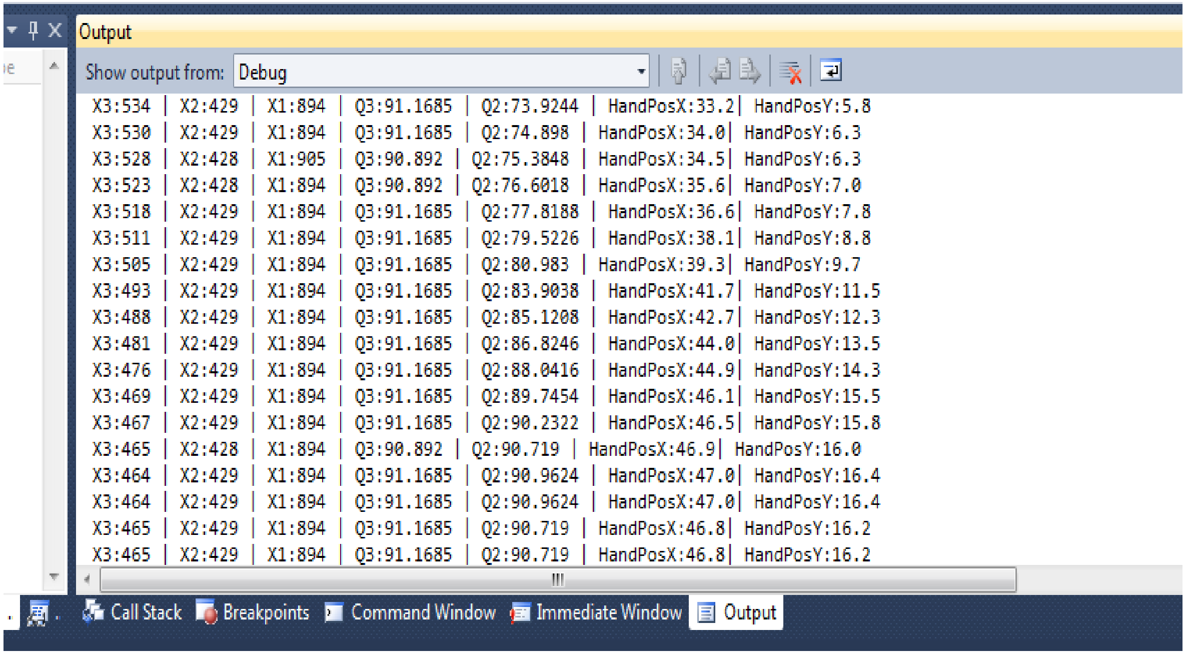


Figure 9.2: Four therapeutic computer games

9.5 Testing of input reading

There is a need to test the input reading from the communication system in order to ensure that the reading inputs are correct. The testing process contains reading sample inputs that describes the sensors readings in voltage, then convert those values from voltage values into position angles. Moreover, it contains moving the end-effecter on the device space in order to calibrate device space with screen space.

While testing the inputs from the communication system throw the Input-RTCS module, the following sample of output had been generated, as shown in Figure 9.3.



```
X3:534 | X2:429 | X1:894 | Q3:91.1685 | Q2:73.9244 | HandPosX:33.2| HandPosY:5.8
X3:530 | X2:429 | X1:894 | Q3:91.1685 | Q2:74.898 | HandPosX:34.0| HandPosY:6.3
X3:528 | X2:428 | X1:905 | Q3:90.892 | Q2:75.3848 | HandPosX:34.5| HandPosY:6.3
X3:523 | X2:428 | X1:894 | Q3:90.892 | Q2:76.6018 | HandPosX:35.6| HandPosY:7.0
X3:518 | X2:429 | X1:894 | Q3:91.1685 | Q2:77.8188 | HandPosX:36.6| HandPosY:7.8
X3:511 | X2:429 | X1:894 | Q3:91.1685 | Q2:79.5226 | HandPosX:38.1| HandPosY:8.8
X3:505 | X2:429 | X1:894 | Q3:91.1685 | Q2:80.983 | HandPosX:39.3| HandPosY:9.7
X3:493 | X2:429 | X1:894 | Q3:91.1685 | Q2:83.9038 | HandPosX:41.7| HandPosY:11.5
X3:488 | X2:429 | X1:894 | Q3:91.1685 | Q2:85.1208 | HandPosX:42.7| HandPosY:12.3
X3:481 | X2:429 | X1:894 | Q3:91.1685 | Q2:86.8246 | HandPosX:44.0| HandPosY:13.5
X3:476 | X2:429 | X1:894 | Q3:91.1685 | Q2:88.0416 | HandPosX:44.9| HandPosY:14.3
X3:469 | X2:429 | X1:894 | Q3:91.1685 | Q2:89.7454 | HandPosX:46.1| HandPosY:15.5
X3:467 | X2:429 | X1:894 | Q3:91.1685 | Q2:90.2322 | HandPosX:46.5| HandPosY:15.8
X3:465 | X2:428 | X1:894 | Q3:90.892 | Q2:90.719 | HandPosX:46.9| HandPosY:16.0
X3:464 | X2:429 | X1:894 | Q3:91.1685 | Q2:90.9624 | HandPosX:47.0| HandPosY:16.4
X3:464 | X2:429 | X1:894 | Q3:91.1685 | Q2:90.9624 | HandPosX:47.0| HandPosY:16.4
X3:465 | X2:429 | X1:894 | Q3:91.1685 | Q2:90.719 | HandPosX:46.8| HandPosY:16.2
X3:465 | X2:429 | X1:894 | Q3:91.1685 | Q2:90.719 | HandPosX:46.8| HandPosY:16.2
```

Figure 9.3: Sample input from sensors

9.6 Device Test

This section is devoted to the device testing as a complete system. All the previous components are connected with each other to be tested. The device with the final shape is explained in the following figures. These figures consist of three parts:

1) The mechanical structure with the load system, the three sensors and the control box (where the battery, XBee and Arduino exist) as shown in Figure 9.4.

2) The LCD screen for the patient as shown in Figure 9.5.

3) The computer for the therapist as shown in Figure 9.6.

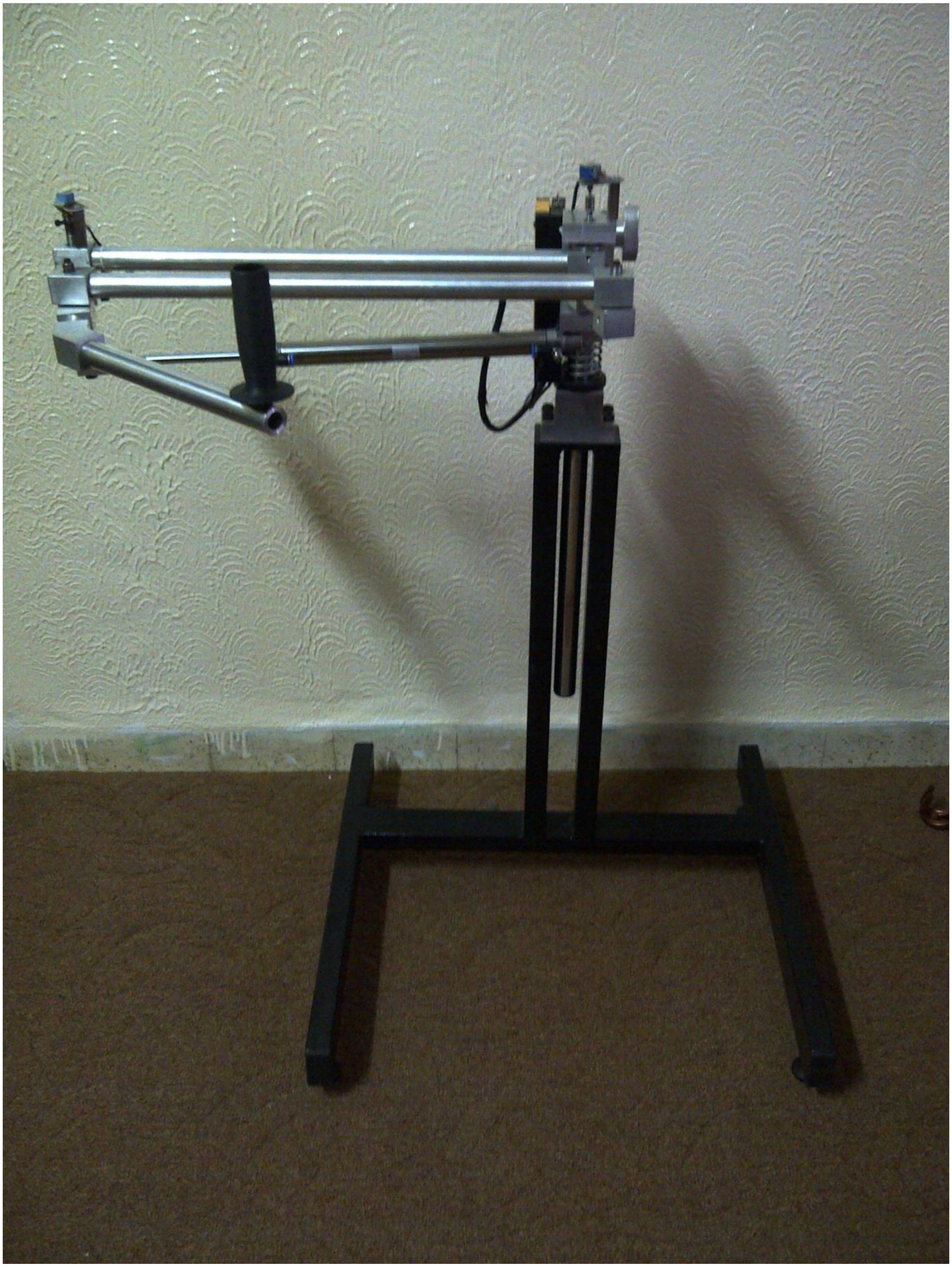


Figure 9.4: The mechanical structure with the load system, the three sensors and the control box

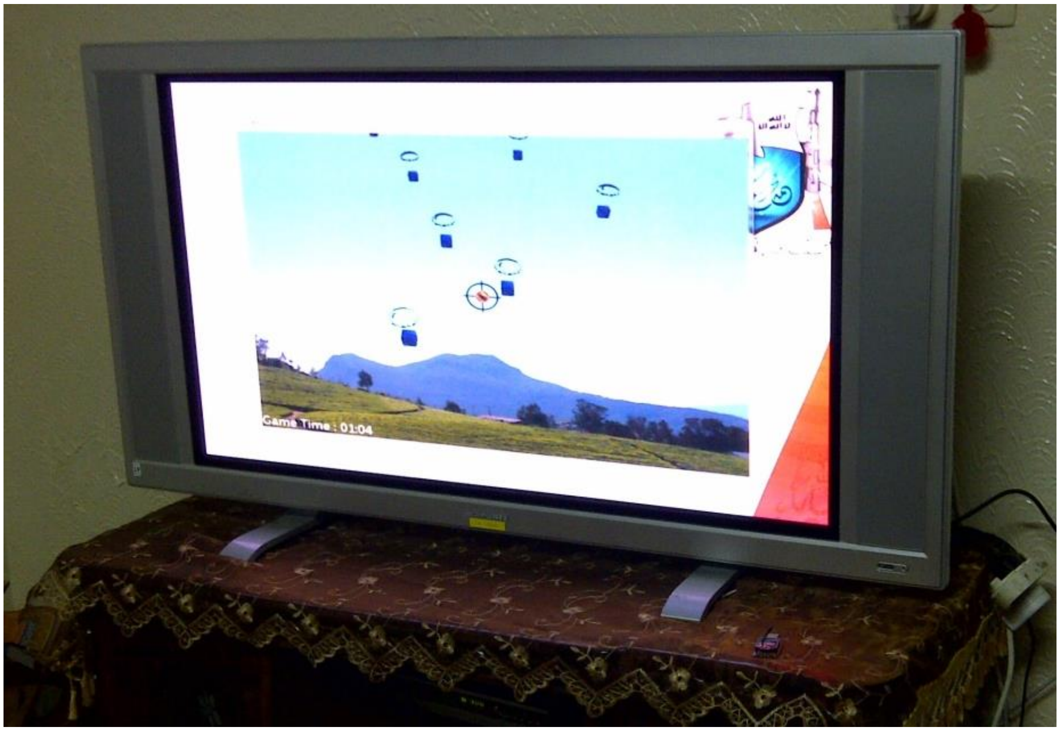


Figure 9.5: LCD screen on the patient side

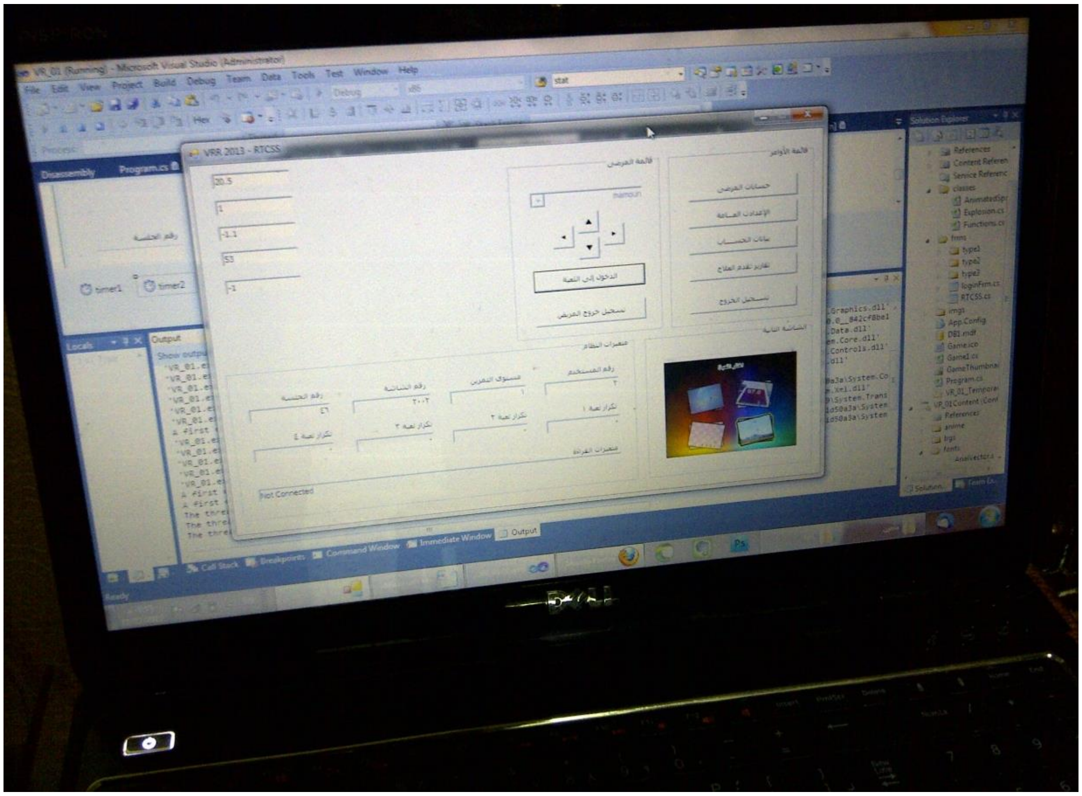


Figure 9.6: The therapist computer

The whole system is tested staring with the portability and flexability of the device after connecting the load system and the interfacing components. The system achieves these two features. The next step is checking the device suitability for the users; the device achieves it easly through the configuration and the height joints. After that, the system is tested for its smoothness in the end-effector motion without the load system. The interfacing is done with the two requiered features (symmetrical and synchronized interfacing) with all the games and the results are acceptable.

CHAPTER 10
Evaluation

10.1 Introduction

This chapter provides the final step for the virtual rehabilitation robot, where physiotherapist Dr. Akram Amro evaluates the whole device as a clinical one with a real stroke patient. The following sections explain the evaluation process which is performed in two sessions with the physiotherapist and the patient about the device.

10.2 First session

This session was held on 5/1/2013 from 12:30 pm to 4:30 pm at the therapist's clinic. The idea with the main features of the device is explained to Dr. Akram. The first impression for him is that he did not expect such a device with a nice shape. The detailed test begins as if a patient is using the device. The start window on the therapist's computer appears asking for the administrator's user name and password. After entering these data the main window is opened to let the therapist creates accounts for each patient with a password and determines the specifications for the session.

These specifications include the chosen game, the period for each one, the previous reports showing the improvement of the patient and an option for split the screen into two parts. These two parts are: one displays the game only on a separate LCD for example for the patient and the second is the computer's (or laptop) screen which displays the game with other options for the session control and games selection.

The therapist tests the workspace and the flexibility of the mechanical part by moving the end-effector in the horizontal plane using both the left and the right hands. The next step is changing the height of the device to be suitable if the user is sitting on a chair, lying on a bed or even if he is standing. Dr. Akram suggests replacing the fixed-bases of the device with wheels with stopper to increase the portability of the device in the clinic. The reason behind choosing the fixed-bases over the wheels with stopper is to be sure that the device is safe all the time and that no one will forget to use the stopper while the patient is using the device which may expose a patient's life in danger.

The next step is increasing the load with the two proposed methods either by hand or by a stepper motor. It is decided to install the motor for the automatic load control. Dr. Akram preferred the manual control with the ability to change the load to a suitable range, so he can test it with his own hands and change it as he likes. Another important thing is that if the patient becomes tired of using the device and he want to continue the exercise, he can change it while he is near the device without the need to use the therapist's computer to change it by himself or ask the therapist to do that.

The load is set at the maximum and the therapist moves the end-effector. He is asked to notice if the load difference and if it is small compared to the required one. The answer is that the load is enough for a patient having stroke since the mission is improving the hand's motion and coordination under different loads and not bodybuilder. Based on the therapist requirements, using a motor for the load control is not an option anymore. From an engineering point of view, using manual control saves energy and makes the battery lasts for a longer period.

The four games are tested while moving the end-effector. Dr. Akram thinks that these games are suitable in motivating patients. In addition, they allow for the completion of cognitive tasks and coordinated motions. All the games with their ability to increase the difficulty of levels by increasing the speed and the number of objects with the random positions that are not predictive are very useful to kill the bored of the traditional rehabilitation exercise.

The final test is the configuration test especially for the inclined plane. One of the main targets of the project is studying if the inclined plane is suitable for stroke patients or not. The surprise is that this plane is the most useful for performing exercises almost similar to activities of daily living like eating or shaving. The vertical plane is tested also which shows that it is useful because the patient has to lift the total weight of the device while performing the exercise so it is useful for muscle strength.

10.3 Second session

This session was held on 6/1/2013 from 3:45 pm to 5:50 pm at the therapist's clinic. It aims to know the first impression of the patient after seeing the device and his/her opinion about the device. A female patient (NF, 56 years old) is met and asked to try the device. This patient agrees to use the device and gives information about her situation while recording a short video showing her uses the device under direct supervision of Dr. Akram. She is laying on the bed because she cannot feel comfort on a chair.

She is asked about her first impression of seeing the device and if it scary or not. The answer is that the device is user friendly and not a scary device. Dr. Akram tests her concentration and coordination by let her follows his finger in different positions. This test is performed before deciding which configuration is suitable for her. Dr. Akram decides that she needs the coordination test and this means that the suitable configuration is the horizontal plane.

The patient starts with Collecting Money game in two parts one with a load and the other without. The second game is Falling Parachutes. The third one is Memorizing Path and the final game is Steady Hand Game. While she faced difficulties in the first game until she get used of it, she find these games interesting especially the second and the third games.

The only thing that should be taken into account for future tests is to put the LCD screen for the patient on a high place (for example hanging it on the wall) which is not possible to do because of the lack of time.

The final evaluation of Dr. Akram Amro is found in the next section where it is written by Dr. Akram himself.

10.4 Final evaluation of Dr. Akram Amro

The hand rehabilitation device

After testing the upper extremity rehabilitation device, I think it is a great contribute to the rehabilitation of stroke patients in Palestine, that could be replicated for further research and use in physiotherapy department at Palestine and even worldwide

The resistance in the device would contribute to more challenging performance from strength point of view, the manual resistance that would help in a constant resistance that represents a certain load held by hand. The games that had been developed by the researchers contribute to the process of rehabilitation in different perspectives.

Memorize Path Game participates in enhancing the cognitive abilities of the stroke patients, a major problem that usually is seen after stroke, together with that, the horizontal movements in this game facilitates the control of upper extremity in stroke rehabilitation

Falling Parachutes Game would definitely improve the eye hand coordination and the sequencing process in a functional task. Which plays a pre request ability toward performance of independent functional activities in everyday living tasks.

The inclination challenges both the strength of the upper extremity as it is against gravity, and enhances hand control in eccentric contraction, that helps in control of hand movement deceleration with the gravity direction, and at the same time represents a “part task” of many functional activities like eating, shaving, teeth brushing that requires those angles of movements, and this combination of shoulder abduction, elbow flexion, and wrist control

CHAPTER 11

Conclusion and Future Works

11.1 Introduction

This chapter provides the conclusion of the whole device as a complete system. In addition, the future works are found here**.**

11.2 Conclusion

The device is proved to be suitable for stroke patients with either left or right hand impairment. It is suitable also for the patient if he/she is sitting on a chair, lying on a bed or even standing. It is useful for the completion of cognitive tasks and allows for performing coordinated motions in the horizontal and inclined planes. It is useful for muscles strength especially in the inclined and the vertical planes. The novel feature of the inclined plane is found to be useful for performing activities similar to that of daily living. This device is ready for real clinical testing and for evaluating its effectiveness in physiotherapy following well designed.

11.3 Future Works

The following four tasks are suggested as future works:

1. Testing and evaluating the device with real patients.
2. Compare the improvement of these patients with other patients whom do not use the device.
3. Adding one or two buttons at the end-effector to increase the challenge for the patients (i.e. do some actions that require more than a movement).
4. Modify the device at the end-effector to allow for object grasping for example which is important in performing the Activities of Daily Living (ADL).

- References

[1] Physiotherapist Samih Dweik, March, 9, 2012.

[2] Cerebrovascular accident definition, 2011, Definition of Cerebrovascular accident, medterms.com.
http://www.medterms.com/script/main/art.asp?articlekey=2676

[3] World Health Organization, 2008, The 10 leading causes of death by broad income group, who.int
http://www.who.int/mediacentre/factsheets/fs310/en/index.html

[4] Physiotherapist Dr. Akram Amro, March, 11, 2012.

[5] Stroke-rehab, Hemiplegia and Hemiparesis, Stroke-rehab.com
http://www.stroke-rehab.com/hemiplegia.html

[6] Holden, M. K., "Virtual environments for motor rehabilitation", 2005.

[7] A. L. Betker, T. Szturm and Z. Moussavi, " Development of an Interactive Motivating Tool for Rehabilitation Movements", University of Manitoba, Manitoba, Canada, 2005.

[8] Matthew J. D. Taylor, et al, "Activity-promoting gaming systems in exercise and rehabilitation", Colchester, Essex, UK, 2011.

[9] G. Burdea, "Virtual Rehabilitation- Benefits and Challenges", CAIP Center, Rutgers University, 2003.

[10] Physiotherapist Dr. Ali Abu-Ghazala, February, 1-2, 2012.

[11] Physiotherapist Munther Oweiwi, February, 7-9, 2012.

[12] M. Casadio, V. Sanguineti, P. G. Morasso, V. Arrichiello, "Braccio di Ferro: A new haptic workstation for neuromotor rehabilitation", Technology and Health Care 14:123–142, 2006.

[13] S. Mazzoleni, et al," Robot-aided therapy on the upper limb of subacute and chronic stroke patients: a biomechanical approach", Scuola Superiore Sant'Anna and Auxilium Vitae Rehabilitation Centre, 2011.

[14] A. Basteris, et al, "A tailored exercise of manipulation of virtual tools to treat upper limb impairment in Multiple Sclerosis".

[15] Z. Wright, L. Patton, V. Ravichandran, “Startle reduces recall of a recently learned internal model”, 2011.

[16] V. Squeri, J. Zenzeri, P. Morasso, A. Basteris, “Integrating proprioceptive assessment with proprioceptive training of stroke patients”, 2011.

[17] V. Squeri, et al, “Bilateral robot therapy based on haptics and reinforcement learning: feasibility study of a new concept for treatment of patients after stroke”, 2006.

[18] D. Piovesan, M. Casadio, F. A. Mussa-Ivaldi and P.G Morasso,”Multi-joint arm stiffness during movements following stroke: implications for robot therapy”, IEEE, 2011.

[19] P. Lam, "A haptic-robotic platform for upper-limb reaching stroke therapy: Preliminary design and evaluation results", 2008.

[20] Ecma International Organization, "C# Language Specification", 4th Edition, June 2006, ECMA-334.

[21] Jin-Shyan Lee, Yu-Wei Su, and Chung-Chou Shen, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi", Taiwan, 2007.

[22] Mark Langdon Sivak, "Design Of Virtual Environments For Rehabilitation Devices", Master of Science in Mechanical Engineering, Northeastern University - Boston, August 2009.

[23] E. Guglielmino, et al, (2008), "Semi-active Suspension Control Improved Vehicle Ride and Road Friendliness", London: Springer.

[24] FLUDICON Company, 2012, *The Smart PID System from FLUDICON* http://www.fludicon.com/industrial/products/

[25] Richard G. Budynass and J. keith Nisbett, "Mechanical Engineering Design ", Ninth Edition, New York: McGraw-Hill.

[26] F. Tinmark (2007), "Analysis of elite golfers’ kinematic sequence in full and partial shots", MSc, The Swedish school of Sport and Health Sciences (GIH).

[27] B. A. Forouzan, Data Communications and Networking, 2009.

[28] S. S. Riaz, "The role of ZigBee technology in future data communication system", India, 2007

[29] A.Lonza,"RF management application using ZigBee network", Germany, 2010

[30] C. Yawut and S. Kilaso, "A Wireless Sensor Network for Weather and Disaster Alarm Systems", Thailand, 2011

[31] S. A. Mitilineos, "Indoor Localization with Wireless Sensor Networks", Greece, 2010.

[32] Digi International, 2009
www.digi.com

[33] R. Faludi, Wireless Sensor Network, First Edition, December, 2010

[34] B. Kinney, "Consulting ZigBee Technology: Wireless Control that Simply Works", Germany, 2008

[35] Marije A.J. Baalman, "Sense/Stage - Low Cost, Open Source Wireless", Concordia University/2009

[36] G. Thonet Patrick A. Jacquin P. Colle, "ZigBee – WiFi Coexistence", Schneider Electric Innovation, Department Grenoble, France, April 2008

[37] P. Wotton, "Providing reliable sensing and control using ZigBee wireless networks", July 2006

[38] A. Frisoli, et al, "Robot-mediated arm rehabilitation in Virtual Environments for chronic stroke patients: a clinical study", Pasadena, CA, USA, May 19-23, 2008.

[39] Y. Jung, S.-C.Yeh, J. Stewart, *Tailoring Virtual Reality Technology for Stroke Rehabilitation: A Human Factors Design*, The USC-UT Consortium for the Interdisciplinary Study of Neuro plasticity and Stroke Rehabilitation, 2006.

[40] J. C. Perry, J. A. Ruiz-Ruano, and T. Keller, "Tele-rehabilitation: toward a cost-efficient platform for post-stroke neuro-rehabilitation", Mikeletegi Pasealekua and Parque Tecnologico in Spain, 2011.

[41] N. Pernalete, et al, "Development of an Evaluation Function for Eye-Hand Coordination Robotic Therapy", California State Polytechnic University, Pomona and Western Michigan University, Kalamazoo USA, June 29 - July 1, 2011.

[42] John W. Krakauer, "Motor learning: its relevance to stroke recovery and neuro rehabilitation", Stroke and Critical Care Division, Department of Neurology, Columbia University College of Physicians and Surgeons, New York NY, USA, 2006.

[43] Shea JB, Morgan JB, "Contextual interference effects on the acquisition, retention, and transfer of a motor skill". J Exp Psychol: Hum Learn Mem 1979; 5:179–187.

[44] Meeting with physiotherapist Dr. Akram Amro, Mon 30-4-2012,5:00 PM to 5:45 PM.

[45] A. Alamri, et al, "Haptic Virtual Rehabilitation Exercises for Post-stroke Diagnosis", 2007.

[46] Meeting with physiotherapist Dr. Akram Amro, Mon 30-4-2012, 5:00 PM to 5:45 PM.

[47] K. Morrow, C. Docan, G. Burdea, A. Merians, ″Low-cost Virtual Rehabilitation of the Hand for Patients Post-Stroke", The State University of New Jersey, 2006.

[48] R. Loureiro, et al, "Using haptics technology to deliver motivational therapies in stroke patients: Concepts and initial pilot studies", In Proceedings of Euro Haptics, p 1-6, 2001.

[49] Scottish Intercollegiate Guidelines Network, "Management of patients with stroke: Rehabilitation, prevention and management of complications, and discharge planning", 2010.

[50] Royal College of Physicians Intercollegiate Stroke Working Party, "National clinical guidelines for stroke". London: Royal College of Physicians, 2008.

[51] Elizabeth A. Styles, ″Attention, perception and memory: an integrated introduction", Psychology Press, February 2005.

[52] Kathleen M. Galotti, "Cognitive Physiology In and Out of the Laboratory", Carleton College, 4th edition.

[53] D. Jack, et al, "Virtual reality-enhanced stroke rehabilitation", IEEE Transactions on Neurological Systems and Rehabilitation Engineering, 9, 308-318, 2001.

[54] An Alinean White Paper, ″Source: Microsoft SQL Server and Oracle Database: A Comparative Study on Total Cost of Administration (TCA)", January, 2010, http://www.alinean.com/PDFs/Microsoft_SQL_Server_and_OracleAlinean_TCA_Study_2010.pdf

[55] R. Elmasri and S. B. Navathe, ″Fundamentals of DATABASE SYSTEMS", CHP7+CHP10, Fifth Edition, Greg Tobin, 2007.