Portland State University

PDXScholar

Dissertations and Theses

Dissertations and Theses

Summer 7-13-2016

Vision-Based Motion for a Humanoid Robot

Khalid Abdullah Alkhulayfi
Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds

Part of the [Artificial Intelligence and Robotics Commons](#), and the [Robotics Commons](#)

Let us know how access to this document benefits you.

Recommended Citation

Alkhulayfi, Khalid Abdullah, "Vision-Based Motion for a Humanoid Robot" (2016). *Dissertations and Theses*. Paper 3176.

<https://doi.org/10.15760/etd.3173>

This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

Vision-Based Motion for a Humanoid Robot

by

Khalid Abdullah Alkhulayfi

A thesis submitted in partial fulfilment of the
requirements for the degree of

Master of Science
in
Electrical and Computer Engineering

Thesis Committee:
Marek Perkowski, Chair
Christof Teuscher
John M. Acken

Portland State University
2016

Abstract

The overall objective of this thesis is to build an integrated, inexpensive, human-sized humanoid robot from scratch that looks and behaves like a human. More specifically, my goal is to build an android robot called Marie Curie robot that can act like a human actor in the Portland Cyber Theater in the play Quantum Debate with a known script of every robot behavior. In order to achieve this goal, the humanoid robot need to has degrees of freedom (DOF) similar to human DOFs. Each part of the Curie robot was built to achieve the goal of building a complete humanoid robot. The important additional constraints of this project were: 1) to build the robot from available components, 2) to minimize costs, and 3) to be simple enough that the design can be replicated by non-experts, so they can create robot theaters worldwide. Furthermore, the robot appears lifelike because it executes two main behaviors like a human being. The first behavior is tracking where the humanoid robot uses a tracking algorithm to follow a human being. In other words, the tracking algorithm allows the robot to control its neck using the information taken from the vision system to look at the nearest human face. In addition, the robot uses the same vision system to track labeled objects. The second behavior is grasping where the inverse kinematics (IK) is calculated so the robot can move its hand to a specific coordinate in the surrounding space. IK gives the robot the ability to move its end-effector (hand) closer to how humans move their hands.

Acknowledgments

I am especially grateful for my thesis advisor Professor Marek Perkowski who supported me from my first college term. He has been the inspirational person pushing me toward achieving my dreams. I always got his support when I ran into trouble or needed help with my research or writing. I would also like to thank members of my thesis committee, Dr. John M. Acken and Dr. Christof Teuscher, for their critical remarks that helped me improve the writing of the thesis and give more precise answers to the fundamental questions of this research.

I would also like to thank my mother who raised me to be successful and thankful to those who helped me in creative and professional life. I am now using the discipline she instilled in me. I am also grateful for the time and effort she put into finding the best partner for me. My wife joined my travels in this life and shared every moment. She is always where I need her to be, helping and supporting me, without fatigue or boredom.

I thank God every day for putting people like them in my life. May God bless them all and keep them in a good health and wealth for the rest of their life

Table of Contents:

Abstract	i
Acknowledgments.....	ii
List of Tables	v
List of Figures	vi
1. Introduction	1
2. Background and Related work	12
3. Building Marie Curie.....	18
3.1 Body Proportions.....	18
3.2 Building the Leg.....	20
3.2.1 Torque Calculation.....	24
3.3 Building the Body	25
3.3.1 Bending Motion	29
3.3.2 Rotating Motion	49
3.4 Building the Left Arm and Body Modification.....	54
3.5 Building the Neck.....	61
4. Using the Interactive Genetic Algorithm to Evolve Motion of Leg.....	63
4.1 The Steps to Implement the Algorithm in the Program	64
4.2 Chromosomes.....	65
4.3 Fitness Function	65
5. Using a Genetic Algorithm to Develop Manipulation Behavior for an Arm	67
5.1 Closed-loop Control System	68
5.2 Feedback Circuit	68
5.3 The Steps to Implement the GA Algorithm in the Program to Control the Cup Motion	71
5.4 Fitness Function for the GA	71
6. Tracking.....	72
6.1 Related Work – background for humanoid tracking.....	73
6.2 Robot Tracking Scene	74
6.3 Tracking Algorithm.....	74

6.4	Using Kinect for Face Tracking	75
6.5	Using Kinect for AForge Glyphs Tracking.....	78
6.6	Evaluation of Tracking.....	81
6.7	Results of the Evaluation of Tracking.....	86
7.	Grasping.....	92
7.1	Related Work – Background on android robot grasping.....	92
7.2	Robot Grasping Scene.....	94
7.3	Using Inverse Kinematics	94
7.3.1	Geometric Method	100
7.4	Reachable Coordinates.....	103
7.5	Evaluation of Grasping.....	109
8.	Conclusion	113
8.1	Contribution	121
	Bibliography	123
	Appendix A: The Power Supply	125
	Appendix B: Pololu Maestro Servo Controller.....	127
	Appendix C: Hardware Controlling Program	129
	Appendix D: Main Application	131
	Appendix E: The System Design.....	135
	Appendix F: The Supplemental Files	137

List of Tables

Table 3-1 A snap shot of the calculation for both gears	36
Table 6-1 Experiment one for the robot.....	86
Table 6-2 Experiment two for the robot	86
Table 6-3 Experiment one for the human	87
Table 6-4 Experiment two for the human	87
Table 7-1 The look-up table for the elbow angles	106
Table 7-2 The look-up table for the shoulder angles	107

List of Figures

Figure 1-1 Marie Curie robot.....	3
Figure 2-1 The InMoov robot	13
Figure 2-2 The Poppy robot.....	14
Figure 2-3 The PR2 robot	15
Figure 2-4 The Atlas robot.....	16
Figure 2-5 The RoboThespian robot.....	17
Figure 3-1 Body proportion calculation (Zarins, 2011).....	19
Figure 3-2 The leg kinematics	21
Figure 3-3 The hip	21
Figure 3-4 The knee design.....	22
Figure 3-5 The ankle	23
Figure 3-6 Explanation of Torque formula for leg design (Nave, 2009).....	24
Figure 3-7 The mannequin – side view.....	26
Figure 3-8 The mannequin – front view	26
Figure 3-9 The skeleton - from inside the body.....	27
Figure 3-10 Upper body with the skeleton - from the side	28
Figure 3-11 Upper body with the skeleton - from the front.....	28
Figure 3-12 The Monster Torque (Thomas, 1994)	30
Figure 3-13 The universal joints	31
Figure 3-14 The pipe with the universal joint (straight)	31
Figure 3-15 The pipe with the universal joint (bended)	32
Figure 3-16 Bending the body forward (the upper part).....	32
Figure 3-17 Bending the body forward (full body)	33
Figure 3-18 The space between the two wooden dowels	34
Figure 3-19 gear train (Norton, 1992).....	38
Figure 3-20 Compound gear train (Norton, 1992).....	38
Figure 3-21 The Monster Torque designed by the author (top view).....	39
Figure 3-22 The Monster Torque (side view).....	40
Figure 3-23 Shaft channel	41
Figure 3-24 The shaft of the driven gear	42
Figure 3-25 The points where the two gears contact (Norton, 1992)	43
Figure 3-26 The servo gears - from inside.....	44
Figure 3-27 The different types of gear splines (Thomas, 1994)	44
Figure 3-28 The plastic piece that holds the gear to the pipe	45
Figure 3-29 The plastic piece that holds the gear to the pipe – from a different angle	46
Figure 3-30 The support for the device.....	46
Figure 3-31 The long threaded rod at the center of the body.....	47
Figure 3-32 The long threaded rod at the center of the bending point	47
Figure 3-33 The device with the long threaded rods and the plastic piece.....	48

Figure 3-34 Measuring the body weight.....	49
Figure 3-35 A was the servo gear and B was the gear that was fixed to the wood under the body (Norton, 1992).....	50
Figure 3-36 The driven gear under the body	51
Figure 3-37 The Lazy Susan	52
Figure 3-38 The wood base (top view)	53
Figure 3-39 The two holes under the wood base	53
Figure 3-40 The space between the body and the wood base.....	54
Figure 3-41 The Marie Curie robot after adding the new body and new left arm. This design still lacked good female body proportions, so I used the InMoov design and 3D printing.....	56
Figure 3-42 InMoov left arm (Langevin, 2012).....	57
Figure 3-43 InMoov arm attached to the torso (Langevin, 2012).....	58
Figure 3-44 The bicep of the left arm	59
Figure 3-45 Curie robot after adding InMoov left arm.....	60
Figure 3-46 InMoov robot neck (Langevin, 2012)	62
Figure 4-1 The Circuit that is used in the feedback of the algorithm	66
Figure 5-1 Closed-loop control system to control the liquid levels while manipulating the cup.....	68
Figure 5-2 The cup from the inside. The leveling wires are attached to the side of the cup while the ground is attached to the bottom of the cup.	69
Figure 5-3 Feedback circuit that is used to get the level of the liquid in the cup	70
Figure 5-4 Snapshot of the GA code running to test each of the four wires in the cup....	70
Figure 6-1 The robot's sight of view in the red rectangle in the middle of the picture	75
Figure 6-2 The Kinect sensor (Zeng, 2012).....	76
Figure 6-3 The Kinect sensor without the cover (Zeng, 2012).....	77
Figure 6-4 Internal design of the Kinect (Raheja, Chaudhary, & Singal, 2011)	77
Figure 6-5 A group of AForge glyphs (Kirillov, 2006).....	79
Figure 6-6 Two robots moving around (Redlarski, Ambroziak, & Palkowski, 2013)	79
Figure 6-7 AForge glyph on the ground (Kirillov, 2006).....	80
Figure 6-8 The Digital Angle Gauge	82
Figure 6-9 The protractor.....	83
Figure 6-10 An overview of the experiment area where the five points are placed according to the segmentation of the neck rotation angle.....	84
Figure 6-11 The movement of the person in experiment one	85
Figure 6-12 The movement of the person in experiment two.....	85
Figure 6-13 Experiment one analysis	88
Figure 6-14 The vertical gear surrounded by the blue rectangle is responsible for moving the head up and down	89

Figure 6-15 The regular gear surrounded by the blue rectangle is responsible for moving the head left and right	89
Figure 6-16 Experiment two analysis	90
Figure 7-1 DOFs in the shoulder and elbow (Abdullah, Tarry, Datta, Mittal, & Abderrahim, 2007).....	95
Figure 7-2 FK explanation (Wyeth & Milford, 2011)	96
Figure 7-3 IK explanation (Wyeth & Milford, 2011)	97
Figure 7-4 The distance between the shoulder and the elbow	99
Figure 7-5 The distance between the elbow and the end-effector	99
Figure 7-6 Geometric of two DOFs robot.....	100
Figure 7-7 Drawing the geometric of the two DOFs for the robot	101
Figure 7-8 Triangle	102
Figure 7-9 Measuring the angle of the shoulder	104
Figure 7-10 Measuring the angle of the elbow	105
Figure 7-11 The calculation for the reachable coordinates by the arm, and the snapshot shows the area that the arm can move in where the blue and green cells intersect	108
Figure 7-12 The angle of a human shoulder (American Academy of Orthopedic Surgeons, 2011)	110
Figure 7-13 Human elbow angle (American Academy of Orthopedic Surgeons, 2011)	111
Figure 8-1 Nora shaking the robot's hand	121

1. Introduction

Robot Theater allows for the demonstration of various robot capabilities in a way that is fun, interesting and edifying. For this reason, the Portland Cyber Theater was established to demonstrate various types of robots built by students in PSU's Electrical and Computer Engineering Department. Because building a humanoid robot that performs as a universal actor with all possible motions and behaviors is still beyond the reach of even top research institutions, such as Disney Research, the idea is to build specialized humanoid robots that are able to perform certain selected behaviors for the plays that they participate in. For instance, the Marie Curie Robot has to perform specific behaviors of human Marie Curie from a theatrical script about. The theater demonstrates how the robots use various innovative hardware and software technologies that allow them to execute a multitude of tasks and individual behaviors. The theater is an exercise in robot system integration, and it is well-known that integration is the most difficult aspect of humanoid robot design. In addition, the robot theater can demonstrate how different robots can interact with each other and with humans and how information from different sources can be integrated to make the robots reason, control, perceive and respond to the state of the environment around them. Thus, not only is a robot an integrated system, but the group of robots is a robot system of a higher order, and each robot should be designed for integration into this system. Some of the theater robots are android robots that are intended to behave like real humans. For instance, the theater robots can perform human-like motions, track a person or a robot, grasp an object, shake hands, etc. All of these human-like actions improve the quality of theater performances

and make watching it more interesting and satisfying. However, adding human-like actions to the robot requires a lot of knowledge, about both hardware and software. However, by integrating the right theory with the adequate technology, the theater designers will be able to create the intended behaviors for a group of robots.

The work presented in this thesis contributes to the Portland Cyber Theater. This theater has several robots and stage props. The design presented in this thesis for Marie Curie robot is the foundation design that allows the robot to perform the together with Einstein and Schrodinger Cat robots. Marie Curie was a Nobel Prize laurate in physics and chemistry who worked in the first part of the twentieth century. She is one of the main characters in two plays at the Portland Cyber Theater. Since she is a chemist, her performance in the theater should be related to her knowledge and skills in chemistry. She will perform a sequence of tasks related to chemical experiments in order to demonstrate how well her body is designed and how the motions of the body can be controlled. For the purpose of this thesis, the Marie Curie robot will execute tasks related to natural autonomous motions like greetings, drum playing, as well as vision-related behaviors such as tracking and grasping. However, the original Curie robot that I started with did not have a neck. In addition, the arms and the body were not functioning in the right way and they missed some important degrees of freedom. The body parts were not proportional and the grasping behaviors were very shaky. Figure 1-1 shows a picture of the Curie robot before my involvement in this project.

Figure 1-1 Marie Curie robot

This thesis presents my work done on the Marie Curie robot. My work includes work on the body, legs, neck and arm of the Curie robot that, when moving together, will be able to perform moving, tracking and grasping tasks. Thus, the presented work relates to mechanical, electrical and software design of an integrated theatrical humanoid human-size robot. These kinds of robots are very rare. They consist of a few robots in Disneyland, Disneyworld and the RoboThespian robots that are sold to various theaters across the world. The commercial appeal of these robots has meant very little has been published about their design because companies do not want to share information. For most of them there are no publications at all. The abovementioned robots cost between \$8500 and \$450,000 and are built by large teams. They require complicated processing methods and advanced machining tools. For instance, the research on the Honda robot started in the 1970s and was done in a leading company with hundreds of engineers.

There are no low-cost and high-quality android robots in the world that are able to compete with the robot requirements formulated in this thesis.

Although in many ways this research is “basic engineering” with components of mechanical, kinematics and software design, it is nevertheless a pioneering effort in the area of theatrical robot design. Some aspects, such as tracking, also involve “engineering science.” The emphasis is not on going deeply into the theory of one aspect but to create an integrated prototype that is useful for its next users.

The purpose of the tracking behavior is to make the Curie robot track the incoming person with its head. In other words, the Curie robot should be able to look at the person’s face and track him or her while this person is moving around. The other purpose of the tracking behavior is to make the Curie robot recognize and track objects. The goal of the grasping behavior is to make the Curie robot move its arm to the position of the object and grasp it with its hand. The embedded system design and the mechanical design need to work together to achieve the best performance for the scenes of the play Quantum Debate (Perkowski & Dhawan, Quantum Consciousness, 2014).

The mechanical design of the robot is one of the most important initial aspects of building an android robot. It sets the limits of the robot’s extremities and relates to many issues that are handled by the embedded system. Building an android robot is much more difficult than building a humanoid robot because a humanoid robot only needs a head and two arms, but an android robot should resemble a human in all body proportions. In previous designs PSU ECE students tried to build good limbs and a head with realistic

behaviors and robustness, meaning ideally no vibrating motions and the ability to repeat motions. This goal was not achieved using the capabilities of our shop, which is equipped only with simple tools like drills and saws. Aluminum, wood and plastic were used, and no 3D printing technology were used.

Therefore, Prof. Perkowski and I decided that the design of an open-source robot called InMoov, designed by Gael Langevin (Langevin, 2012), should be used to get the best design of some of the body parts. My work, as well as the work of other students proved that this was a very good decision, and future robots in the Intelligent Robotics Lab will all use at least some 3D-printed components, such as fingers, palms and forearms. This is the first MS thesis at PSU that relates to the new 3D printing technology for robotics and the first that integrates complete mechanical design with various software algorithms for motion.

The 3D-printed parts allow for easy construction of the complex components of the human-like body, such as fingers and internal components such as gears that control shoulders. However, Prof. Perkowski and I decided to build a robot that is not a direct replica of InMoov. Instead we built a different body and head for Marie Curie, and added legs. While InMoov is only a fixed torso that cannot move and has no legs, my robot is a complete humanoid robot that sits. The Curie robot uses its upper body and lower body for various behaviors such as playing drums, playing other instruments, kicking the non-obedient Schrodinger Cat robot. In addition, the Curie robot uses the whole body for

complex motions that were not possible for the original InMoov, nor for the previous version of the Curie robot.

The overall objective of building the Curie robot was to create a realistic humanoid robot that can act like a human actor playing Marie Curie in the Portland Cyber Theater. In addition, there are some sub goals that determine the reasons for building each part of the Curie robot and how each part needs to be designed. The goal of building the leg was to give the Curie robot the ability to do some entertaining behaviors such as playing a drum or kicking the non-obedient Schrodinger Cat robot. Moreover, the main objective for building the new body for the Curie robot was to give the robot a body that is robust enough to carry the neck, head, and arms. Furthermore, the body should be able to bend forward and rotate. The goal of the bending motion is to give the Curie robot the ability to look at the Schrodinger Cat robot and to the other robots that are located on the ground, as required by the Quantum Debate script. The goal of adding rotating motion is to allow the Curie robot to interact with its surroundings during the robot theater performance. She rotates to Bohr, Einstein and Newton who stand near her and talks with each of them in turn. Furthermore, the main objective of building a new arm is to replace the old arm of the Curie robot with a more realistic arm that has all the degrees of freedom (DOFs) functioning properly. The new arm should also restore good upper body proportions. Finally, the goal of adding a neck to the Curie robot is to make it track things with its head. This is the third-generation arm design and the second-generation neck design. Both take into account unsuccessful tests of the previous versions. The behavior of all body components should be robust, where robust means: 1) ability to carry weight

and torque, 2) precision of manipulation, as demonstrated in IK, 3) repeatability of motion, 4) force sufficient to make the motion possible or interesting, such as kicking the drum, and 5) speed and acceleration necessary to make the motion human-like. All these requirements are difficult to satisfy together, so some trade-offs are necessary and several experiments were done with different variants.

In addition to innovative mechanical design, the embedded system design is another important aspect of building a humanoid robot. The presented design, which is explained in appendix E, includes a Kinect sensor for vision and an Arduino Uno with a Pololu Servo Controller for moving the arms, legs, neck and torso. The mechanical and electrical control subsystems are combined using the algorithms created by me for tracking behavior and the IK for grasping behavior. Observe that grasping and tracking behaviors are easier on an industrial robot than on an android robot. The Marie Curie robot is the first android robot in the Intelligent Robotics (IR) laboratory that has these abilities. Although several previous projects achieved goals of grasping and tracking, it was never done with such accuracy and speed as achieved here, and never on android robots in the IR laboratory.

Tracking a person's face is done using an algorithm and software created by me that ensures the robot is looking at the person's face all the time (All software was written by me from scratch, as my goal was easy portability). My algorithm takes the skeleton information from the Kinect sensor and tries to keep the face in the center of the image. It executes this action using the two DOF in the neck. However, the grasping behavior uses

the calculations for the IK to reach a known object's position within the reachable space of the hand. The arm is going to take a cup and pour its content into another cup. In addition, the arm can shake the cup to mix the content and it should be able to grasp more than one object.

To evaluate the tracking algorithm, the robot tracking ability is compared to human tracking. The experiments with the algorithm include two ways showing how fast and how natural the robot can track a human face using the algorithm. Substituting the robot with a human and running the same experiments demonstrates the difference between the tracking algorithm and human tracking. By comparing the results from testing both the robot tracking algorithm and human tracking, I can analyze how close the algorithm tracks a human face compared to a normal human. Similarly, the grasping behavior of the robot arm is evaluated and compared to human grasping. Since the main reason of building the android robot here is to create a robot that looks and behaves like a human, it is reasonable to compare the robot arm to a human arm. This comparison allows us to evaluate the arm design and how well it grasps objects compared to human grasping.

The Interactive Genetic Algorithm (IGA) is an algorithm that can find the solution to a not well-specified problem. It uses the human observation of leg movement to generate a better leg movement that is closer to the desired solution. The goal of using IGA was to add animation to the leg for some fun and interesting movements. The problem was that the robot should kick the drum. However, the solution was not precisely

defined, and we just wanted to have the user observe and agree that this is a sufficiently good drumming action or the leg is close to kicking the drum in an interesting way. The algorithm successfully achieves the goal by using feedback from a human.

The Genetic Algorithm (GA) has the same principle of IGA except it requires no human interaction. The goal of using GA is to add the behavior of balancing the cup used in the robot theater. The innovation of my algorithm here is that the feedback is not from software, but from contact sensors that measure levels of the liquid from different points. The algorithm successfully achieves the goal.

The work in this thesis is important in building an android robot that looks and acts similar to humans. The work shows a way to build an android robot whose parts are close to that of a human and has exactly the same number of DOFs. For instance, there are five DOFs in the leg, two DOFs in the body, five DOFs in the arm, and two DOFs in the neck of the Marie Curie robot. All these DOFs satisfy the tasks requirements for the robot theater. In addition, they are built with low cost materials. Moreover, the new tracking algorithm makes the robot lifelike. It also makes the robot able to collect data that can be used in future grasping behaviors. Thus, it allows the Curie robot to identify the surrounding objects and their positions. Furthermore, the IK make the robot able to move its hand to given positions like a human being. All these specifications and abilities satisfy the tasks requirements of the robot theater.

There are many advantages that the work in this thesis provides compared to other android robots. Other android robots are created in a way that makes them hard to copy

and expensive, or they would not be able to perform in the Portland Cyber Theater. However, the Curie robot was built with low cost materials that can help humanoid robot enthusiasts to start building their own inexpensive robots. One of the new things that this approach provides is the ability to design and build a robot leg that is fast, cheap, replicable, and mimics human leg behavior except for walking. In addition, this approach shows a way to design and build a robot body that is robust, able to bend and to rotate. The bending motion was done using a low cost monster torque device, which is shown in chapter three. Furthermore, this approach shows the Curie robot ability to mimic human tracking of human faces and labeled objects using an affordable, good quality depth sensor placed on the robot's head.

The work in this thesis can be adopted by many enthusiasts when building humanoid robots for the following reasons. My approach to building the whole robot was to give the high school students who work with Prof. Perkowski the ability to replicate the robot in their high schools. Moreover, making a low cost monster torque device is a low cost solution to magnify the force of the motors. This device solves some old problems that needed powerful motors. For instance, universities with low budgets that need better robot motions can use the proposed Monster Torque Device design. Furthermore, tracking human faces and labeled objects using an affordable depth sensor. The most important reason is that the robot I built will be the foundation of several robots in the IR Laboratory. Nevertheless, the low cost materials used to build the robot, which total around \$1,000, are another reason for this work to be replicated in the future.

Replicating this robot might take between three to five months to build by a teenage team.

In conclusion, the goals of the thesis's research were initially formulated as follows:

1. Design a humanoid, android robot with natural body proportions to play Marie Curie in the robot theater. The robot should be sitting on a desk and should have movable hands, legs, neck, head and body.
2. The robot should move similarly to humans, should be able to track humans and objects and grasp objects.
3. The robot should be inexpensive, at most \$2,000, and the construction should be easy enough to be reproduced by undergraduate students and high school students by using the description created in this work.
4. The robot should be programmed in C# from scratch. It should not use complex software packages such as Orange, Open-source Computer Vision (OpenCV), or Robot Operating System (ROS) which are difficult to use for non-experienced programmers.
5. The system should be complete and integrated so that the next users and designers can program on top of the software developed in this thesis. The first users will be PSU students from Intelligent Robotics classes.

2. Background and Related work

A humanoid robot is a robot whose body and extremities were built to look like a human. In general, a humanoid robot should at least have arms and a head in order to be called a humanoid robot. Android robots are more similar to humans than humanoid robots. They have facial gestures, hands, fingers, legs, etc. that makes them look similar to a human. In fact, the robot shape can be optimized according to the functions that are required from the robot. In other words, the robot can look different according to the tasks that the robot was built for. It is almost impossible to build a general purpose robot that can act like a human in the real world. For that reason, many robots were built to perform only a limited number of tasks or they could be improved to do more tasks (Kajita, Hirukawa, Harada, & Yokoi, 2005).

One of the robots built to be an android robot is the InMoov robot. InMoov, an open-source, 3D-printed, life-size robot, was designed and built by Gael Langevin, a French sculptor and designer. He started building InMoov in January 2012 with the help from the InMoov community. The designer's goal was to build the first open-source, 3D-printed, life-size robot that can be built by anyone who has a 3D printer and is interested in humanoid robots. Therefore, InMoov is a great reference to build The Curie robot. InMoov has a very advance design of the neck and arm. The total cost of the robot is around \$900 (Electronic Products, 2016). The robot has been built by more than 150 developers from around the world. Every one of those developers has their own version and purpose of building it. In general, InMoov was designed with a head, two arms, and a torso. The design has five DOFs in each hand, two DOFs in the neck, and two DOFs in

the stomach. The original robot does not have legs, but the designer was able to make the robot move with a mobile base (Langevin, 2012). Figure 2-1 shows a picture of InMoov.

Figure 2-1 The InMoov robot

Poppy is another humanoid robot designed and built to be a 3D printable robot. It was built in in 2012 in the Flowers laboratory at Inria Bordeaux Sud-Ouest. It was designed to be an open-source robot, but it cost around \$8,500. Unlike InMoov, Poppy is not a life-size robot and it is more expensive. However, Poppy has many ideas and designs that can be used to add more DOF to our robot. The way of Building the body of this robot is a great inspiration for may designs for the Curie robot's leg. Poppy's

assembly takes two to three days, and is easier than assembling InMoov, or so I have found from personal experience (3D printer and 3D printing news, 2011). Figure 2-2 shows a picture for Poppy.

Figure 2-2 The Poppy robot

There is another humanoid robot called PR2 that was built by the Willow Garage company. The goal of this robot was to provide a platform that can be used by researchers and developers to test their inventions. The robot is not an android robot that looks similar to humans. However, it is a humanoid robot that has a head, two arms, and a mobile base. Moreover, PR2 uses its embedded vision system with its two hands to

track and grasp items. Therefore, PR2 can give us a lot of behavior ideas that a humanoid robot can do. In addition, PR2 has a unique head that has the vision system which is similar to our approach of the tracking system. The robot's hands are grippers that allow developers to write simpler code to grasp items than writing a code for a robot with a human hand. The robot has no bending motion, and the only way the robot can look down is by using its neck. PR2 is very expensive. Its price is \$285,000 with one arm or \$400,000 with two arms (Willow Garage, 2008). Figure 2-3 shows PR2.

Figure 2-3 The PR2 robot

Atlas is a more advance humanoid robot that can walk almost like a human. It was built by a company called Boston Dynamics. They designed this robot to have 28 DOFs

with a very strong body. The way that Boston Dynamics add strength to the body is very interesting. They add many support pipes to the robot that does not change the robot's shape. For this reason, our robot has many support martials that was inspired from Atlas. Furthermore, it has its own vision system that can see obstacles while the robot is walking. In addition, the robot can use its hands to manipulate tools designed to be used by humans. However, this robot is still a humanoid robot that can do many tasks like a human but it is not an android robot that can play Marie Curie in a theater. Atlas does not have any human face or human neck that would make him look similar to a human. It is designed for the military rather than public use (BostonDynamics, 1992). Figure 2-4 shows a photo of Atlas.

Figure 2-4 The Atlas robot

One of the very few android robots designed to perform in a theater is the RoboThespian. It is an android robot that looks similar to a human and can perform like a

human in the theater. However, RoboThespian cannot walk since its feet are fixed to its base. The Curie robot is a non-walking robot too, but able to do more behaviors than RoboThespian would do if it was used in the Portland Cyber Theater. Since The Curie robot was built to act in the Portland Cyber theater, it has to be able to track and grasp objects. According to Prof. Perkowski's experience with RoboThespian, the robot cannot grasp items as a normal human can do. Also, RoboThespian has no feedback, no sensors, no camera and no interaction with its surroundings. Animation of arms is very good but animation of the head is poor. It cannot bend and rotate its body. The robot is very expensive, costing around \$57,000 dollars (Engineered Arts Ltd, 2004). Figure 2-5 shows a photo of RoboThespian.

Figure 2-5 The RoboThespian robot

3. Building Marie Curie

The main objective of building the Curie robot was to design and program a realistic android robot that can act like a human in the Portland Cyber Theater, with other secondary objectives as listed in the previous section. Building this type of robots takes a long time. When I started working, the Curie robot had two arms that were not working properly and it was very hard to redesign the arms from scratch. Therefore, Prof. Perkowski and I made a strategic decision to stop using brackets and aluminum pipes and use 3D printing. We chose the InMoov design. The new upper part was attached to our previous Curie robot body that was modified to accept the new parts from InMoov. With help from a group of students we were able to complete the building of the body, the left arm, and the neck of the Curie robot. Finally, I achieved the main goal of building the Curie robot and all the sub goals of each part of the Curie robot.

3.1 Body Proportions

In order to give Marie Curie the best android look, she had to have the right body proportions. I was able to find a calculator that finds body proportions for a given human height, gender, and age (Zarins, 2011). Figure 3-1 shows the calculation results.

Figure 3-1 Body proportion calculation (Zarins, 2011)

The robot's body proportions are based on the height of the face. Since the height of the Curie robot head is 20 cm, I have to take that length as the base value to calculate the other parts. In other words, the size of the other parts of the body such as the arms, legs, and body proportioned in relation to the head. Because the whole robot height should be eight head units tall (Zarins, 2011), the Curie robot is 160 cm tall. The other way to come up with the robot's proportions is to get the desired length of the robot and enter this into the calculator. The calculator then divides that number by eight head units. Therefore, one can calculate the height of the head unit. However, since the head of the Curie robot was built by students from previous classes, I used its head height in my calculations. All other parts of the robot's body were designed from scratch by me. Only the head remained unchanged from the old design since it was working properly.

3.2 Building the Leg

The goal of building the leg was to give the Curie robot the ability to do some entertaining behaviors like play a drum or kick the non-obedient Schrodinger Cat robot. This design of the leg was the second attempt at building an android robot leg in our lab. Because the leg is hidden from audience's eyes by the long dress of Marie Curie, the exact shape of the leg was not a problem. Thus, we only need to focus on the speed, acceleration, and general proportions, as evidenced by kicking behaviors. The Curie robot is a theatrical stationary robot that does need legs designed for walking. Building a leg for a non-walking robot that has to kick a drum, requires building a leg with a fast enough movement to generate a strong and realistic sound.

The Curie leg has five DOFs, two in the hip, one in the knee, and two in the heel. The servos in the hip allow the thigh to move up-down and left-right, and the two servos in the heel allow the leg to move up-down and left-right also. Figure 3-2 shows the kinematics for each joint.

Figure 3-2 The leg kinematics

The movement between XY axes allows the whole leg to rotate to the left or to the right. There is also the movement between the XZ axes that allows the whole leg to move up and down. Figure 3-3 shows the two servos that move the leg left or right and up or down.

Figure 3-3 The hip

The movement of the knee in the XZ plane allows the lower part of the leg to move up and down. The leg design uses a C bracket and ball bearings in the back of the servo to lift the leg, as shown in Figure 3-4.

Figure 3-4 The knee design

The movement of the ankle in the XZ plane allows the foot to move up and down. Furthermore, there is the ankle movement in the XY plane that allows the foot to rotate to the right and to the left. Moreover, the servo that controls the movement of the foot in the XY plane is connected to the closed end of the small C bracket that is fixed to the XZ plane servo horn and a ball bearing from the back. The body of the XZ plane servo is holding the leg as shown in Figure 3-5.

Figure 3-5 The ankle

The tradeoff for the robot leg was that my particular design has a low robustness. Robustness here is the strength at every joint. Since the Curie robot is a non-walking robot, low robustness is not a problem. For instance, if you look at Figure 3-5 the horizontal servo has only one connection point with the C bracket. In the case of this particular leg, shaking behavior or lack of precision is OK. This connection can make the leg shaky and break if the leg kicks something very strongly, but makes the leg cheaper and able to move faster. The part of the leg where robustness is important is the knee. There should be at least two connection points between the lower and the upper part of the leg. This leg design costs \$380, which includes \$280 for eight big servos, \$40 for four small servos, and around \$60 for other materials.

3.2.1 Torque Calculation

As I was not able to find a book about non-walking android leg design, I adapted methods from literature about industrial arms (Nave, 2009). Torque (τ) is the factor that causes the rotation. In this leg design the torque calculation is very important and is calculated using the equation below:

$$\text{Torque } (\tau) = \text{Force } (F) * \text{Radius } (r) * \sin \theta$$

In this equation F is the weight that the servo carries, and r is the length of the thigh or the leg. θ is the rotation angle of the thigh or the leg. Figure 3-6 demonstrates this equation.

Figure 3-6 Explanation of Torque formula for leg design (Nave, 2009)

If the torque applied by the weight is equal to the torque of the servo, then the leg will not move. Therefore, the torque applied by the weight needs to be smaller than the torque of the servo, which is hard to achieve in general. However, I found that C brackets can make radius r shorter, so less torque from the servo can make the leg move.

3.3 Building the Body

For me, building a robot body is one of the hardest yet most interesting parts of the mechanical design part of this thesis. While there are papers about arm design, very little, if anything, has been published about body design or leg design for non-walking robots. The main objective is to build a body for the Curie robot that is robust enough to carry the neck, the head, and the two arms. In addition, the body should be able to bend forward and rotate to the left and to the right. Despite much time that Prof. Perkowski and I spent on searching the internet, we were not able to find a single paper on how to design the robot body with bending ability. All known robots have no DOF allowing them to bend in the way required by the play script.

Body design starts with choosing the right body shape. Since the Marie Curie robot is an android robot, using a mannequin is a great and cheap way to build the robot body. Fortunately, the mannequin we found was exactly 60 cm high which is the right proportion for the robot body. Additionally, the mannequin was separated into three main parts that made working with the body easier. However, it did not have strong support (Skeleton) inside the body that would hold the arms, neck, and head. Figure 3-7 and Figure 3-8 show the original mannequin before any modifications.

Figure 3-7 The mannequin – side view

Figure 3-8 The mannequin – front view

Having a skeleton inside the body is important because the body must carry the weight of the arms, neck, and head. Since the skeleton needed a base, the design started with a base cut from thick plywood. Then the skeleton was built from three pipes

attached to the base of the body. Next, two holes were drilled in the top of the three pipes to insert the wooden dowels that hold the arms. Figure 3-9, Figure 3-10, and Figure 3-11 show the final shape of the skeleton.

Figure 3-9 The skeleton - from inside the body

Figure 3-10 Upper body with the skeleton - from the side

Figure 3-11 Upper body with the skeleton - from the front

The tradeoff in building the robot body was that my particular design had to be less robust, so the body could bend forward. Since standard android robots have no bending motion, adding arms and a neck and having a robust design is relatively easy. However, the Curie robot has the bending motion so there should be two body segments,

which decreases robustness. This adjustment makes attaching the head, neck, and arms more difficult. By sacrificing a little of the body's robustness, I was able to add the bending motion and still support the arms, neck and head. Robustness here means the robot's ability to carry its extremities. This design of the leg costs \$140, which includes \$70 for two big servos, and \$70 for other materials.

3.3.1 Bending Motion

I spent a considerable amount of time studying humanoid and android robots via the internet, and Professor Perkowski has personally investigated in detail these robots in Korea, China, Japan, Germany and Poland, especially the theatrical robot RoboThespian (Engineered Arts Ltd, 2004). By observing the motions of many robots built in Japan and Europe, including the famous InMoov, it was found that the bodies are very rigid. This strength may be necessary for the walking robots, but is a big problem for theatrical stationary robots like the Curie robot. The goal of the bending motion is to give the Curie robot the ability to look at the Schrodinger Cat robot and the other robots on the ground. From what I was able to find on the internet and from what I learned from Prof. Perkowski, there is no robot available that has this ability. This design goal became a challenging problem to solve. The problem is difficult because when the body is bent forward it is very hard to pull it backward using any servo available in the IR Laboratory. Powerful motors with high torques are very expensive and, as one can recall, one of main goals for the Curie robot was a low price. This goal is important because this type of robot is intended to be replicable by high schools and hobbyists in countries all over the world who are not as rich as the USA. Therefore, it was decided to start to work on a

device called the Monster Torque that magnifies the force of the servo. The HS-805BB was used in the building of the Monster Torque because it was the most powerful, inexpensive servo in the lab. Figure 3-12 shows the original Monster Torque design.

Figure 3-12 The Monster Torque (Thomas, 1994)

After it was decided that a home-built Monster Torque was the best solution to our problem, I found that commercial devices are very expensive. Therefore, I decided to build it by myself. However, the body should be bendable before the device is built in order to allow for good measurements and calculations of the Monster Torque servo. The bending motion of the body was achieved by adding the universal joints to the three pipes in the skeleton. Figure 3-13 shows the universal joints.

Figure 3-13 The universal joints

Next, the pipes were cut to fit the skeleton. Then the universal joints were inserted and screwed to each pipe. Figure 3-14 and Figure 3-15 show how each pipe bends.

Figure 3-14 The pipe with the universal joint (straight)

Figure 3-15 The pipe with the universal joint (bended)

Putting all three pipes together allows the body to bend forward. Figure 3-16 and Figure 3-17 show how the three joints should look when they are bent forward.

Figure 3-16 Bending the body forward (the upper part)

Figure 3-17 Bending the body forward (full body)

3.3.1.1 Monster Torque Device

The Monster Torque is a device that uses a servo with a gear to increase the servo's torque. Understanding how it works helped me design the Monster Torque to carry as much torque as needed for any particular problem. The first step to make this device is to measure the weight of all the components that are going to be attached to the body. The weight of the two arms is 4 kg. The head weight is about 1.5 kg. Thus, the whole weight that the device needs to carry is 6.5 kg.

The next step is finding the force point that is generated from the weight of the arms. Since two wooden dowels support the arms, the force point of the arms weight is located in between the two wooden dowels as shown in Figure 3-18. The first wooden dowel is 10.6 cm away from the center of the gear, and the second one is 17.5 cm away. Thus, the average of the two distances is 14.05 cm.

Figure 3-18 The space between the two wooden dowels

To calculate the actual torque needed to carry 6.5 kg, some research and experiments were performed. It was concluded that in order for the device to move the whole weight, the weight used in the calculation should be doubled. In this case the device will carry 13 kg without dropping it. In other words, the device is able to move the 6.5 kg very easily without losing speed. The following equation was used to calculate the torque of the device.

$$\text{Torque} = \text{Force} * \text{Distance} * \text{Sin}\theta$$

$$\text{Torque} = 13 \text{ kg} * 14.05 \text{ cm} * \text{Sin } 90 = 182.65 \text{ kg/cm}$$

The device should have 182.65 kg.cm torque in order to move the upper half of the body up and down. The next step was choosing a servo and calculating the gear ratio for the two gears. The most powerful servo in the lab is the HS-805BB that has 24.7

kg.cm at 6 v and 19.8 kg.cm at 4.8 v. The following equation was used to calculate the gear ratio.

$$\text{Device Torque} = \text{Gear ratio} * \text{Servo torque}$$

The device torque figure used was the 182.65 kg.cm and servo torque was 24.7 kg.cm. Solving for gear ratio gives a figure of 7.4. Since the gear on the servo is the one that moves the larger gear it is called the driving gear while the bigger one is called the driven gear. The ratio of the two gears should be no less than 7.4. The ratio is found by dividing the number of teeth of the bigger gear by the number of teeth of the smaller gear.

Since a 120 teeth gear was available in the lab, the driving gear needed at least 16 teeth. However, finding the exact number of teeth on the driving gear that also had the same pitch number, which defines the distance between the gear teeth, of the driven gear was difficult. The pitch for both of them was 32.

By knowing the gear ratio, it is possible to calculate the speed of the device. In fact, the speed of the device depends on the speed of the HS-805BB servo at 6v which was 0.14 sec/60° as shown in the following equation.

$$\text{Device Speed} = \text{Servo speed} * \text{Gear ratio}$$

$$\text{Device Speed at 6 v} = 0.14 \text{ sec}/60^\circ * 7.5 = 1.05 \text{ sec}/60^\circ$$

I created an Excel file that calculated the number of teeth in the big gear, which is the driven gear. Now one needs only to enter the weight that one wants to carry, the distance where the weight is going to be applied from the center of the driven gear, servo

torque and speed at low and high voltages, and the number of teeth in the driving gear (the servo gear). The next table shows the Excel file. Table 3-1 shows a snapshot of the Excel file that calculated the number of teeth for both gears.

Table 3-1 A snap shot of the calculation for both gears

Monster Torque motor calculation		Values	
input using high voltage	The weight that you are trying to move	6.50	kg
	The distance between the center of the gear and the force	14.05	cm
	The torque of the used servo at high voltage	24.70	kg.cm
	The speed of the used servo at high voltage	0.14	sec/60
	The number of teeth in the servo horn	16.00	Tooth
calculations using high voltage	Torque at the distance that the motor should carry	13.00	kg
	The motor torque	182.65	kg.cm
	The gear ratio for the two gears at high voltage	7.39	
	The motor speed for the servo at high voltage	1.04	sec/60
	The number of teeth in the driven gear at the high voltage	118	Tooth
input using low voltage	The weight that you are trying to move	6.50	kg
	The distance between the center of the gear and the force	14.05	cm
	The torque of the used servo at low voltage	19.80	kg.cm
	The speed of the used servo at low voltage	0.19	sec/61
	The number of teeth in the servo horn	16.00	Tooth
calculations using low voltage	Torque at the distance that the motor should carry	13.00	kg
	The motor torque	182.65	kg.cm
	The gear ratio for the two gears at low voltage	9.22	
	The motor speed for the servo at low voltage	1.75	sec/61
	The number of teeth in the driven gear at the low voltage	148	Tooth

In the above table the inputs are in blue. These inputs are the weight that should be lifted, the distance between the center of the gear and the force point, used servo torque and speed, and the number of teeth in the driving gear. The green fields show the outputs that use the abovementioned equations. The only item that is important is the orange field, the number of teeth that the driven gear should have. The first group of blue and green rows represents the calculations for the high voltage input for the servo, and the second group represents the calculations for the low voltage input. The second orange

row should always be higher than the first one since one needs a larger driven gear to get the same torque from the device at a lower voltage.

In some cases the number of teeth in the gear is too high. One may not be able to find it or it may be too high for one's project. In this case one should play with the values of the inputs. One can try to make the load less heavy or decrease the distance between the center of the gear and the load point. Another possible solution is to use a more powerful servo.

The most effective way to correct for the number of teeth is to increase the gear ratio. One can do increase the gear ratio by finding a smaller gear horn for the servo. This will increase the gear ratio which will eventually allow one to have the same torque with smaller gears. Adding an additional gear is a bad idea because the gear used between the driving gear and the driven gear is going to ensure that they rotate in the same direction. However, it is not going to affect the gear ratio. In fact, using more gears means that one attempts to increase the gear friction, which will reduce the power. Using multiple gears called a gear train and is shown in Figure 3-19.

Figure 3-19 gear train (Norton, 1992)

However, if one uses a compound gear train, one can increase the gear ratio significantly. Figure 3-20 shows a compound gear train.

Figure 3-20 Compound gear train (Norton, 1992)

Building the Monster Torque gave me a lot of experience in using the tools in the shop. Using the right tools is important for getting the best result. Figure 3-21 and Figure 3-22 show the final shape of the device. The calculations as explained above are very common in robotics but little is published on them. Therefore, I believe that his Excel table, together with this explanation will be useful for at least future students in the PSU robotic classes.

Figure 3-21 The Monster Torque designed by the author (top view)

Figure 3-22 The Monster Torque (side view)

The steps of making the Monster Torque device is simple. First, one need to make a wide hole for the servo. Then one need to make another wide channel for the gear shaft, so it can move back and forth. Making the channel this wide helps to change the servo in case it is burned. In addition, it increases the tension between the two gears. Figure 3-23 shows the shape of the channel.

Figure 3-23 Shaft channel

After the base of the device was completed, the driven gear was made by inserting a long screw into the gear. Then the long screw need to be fastened by a washer and nut. Next, two nuts and two bearings were used to fasten the gear to the base. Figure 3-24 shows in what order the components were assembled in the shaft.

Figure 3-24 The shaft of the driven gear

Using the bearings was very important in allowing the shaft to move without changing the place of the nuts. Without the bearings the two nuts would change their place, and they might make the shaft loosen or tighten on the base and prevent the gear from rotating.

Another design aspect that might affect the smooth transfer of power from the driving gear to the driven gear is the gear pitch. The two gears must have the same pitch number in order for them to transfer the full power. Sometimes the two gears will not fit if they have different gear pitch number. Figure 3-25 shows the point where the two gears transfer power. The transfer of power between the two gears might not be perfect if one disposes two gears with different pitch numbers.

Figure 3-25 The points where the two gears contact (Norton, 1992)

In the design of the Monster Torque in this thesis the driven gear had 120 teeth and the driving gear had 16 teeth. The ratio of these two gears means that the driving gear rotates 7.5 times to complete one rotation of the driven gear. This number of rotations could only be accomplished by using a continuous rotation servo. To make the servo rotation continuous, the servo of the type that is available in the lab has to be modified. The modification of the servo is done by opening the servo and cutting the piece of plastic that blocks the servo from having a full rotation. In addition, the servo potentiometer has to be adjusted to the center position so the servo brain thinks that the horn is in the middle all the time. Figure 3-26 shows the servo gears from inside.

Figure 3-26 The servo gears - from inside

Before buying the horn gear for the servo, the designer should look at the type of spline gear in the servo. For the HS-805BB the gear type is the 1D Heavy Duty Spline. Figure 3-27 shows the different types of gears.

Figure 3-27 The different types of gear splines (Thomas, 1994)

Integrating the Monster Torque with the body was very complicated. The problem with attaching the driven gear to the pipe was that the gear tended to push forward from

one side and backward from the other side. This movement made the device work very weakly since the two forces were cancelling each other. The solution for this problem was to use two consecutive holes in the gear.

A very strong plastic piece was finally used to hold the gear to the pipe. Another 2 x 4 piece of wood was used to keep the device from moving backward. Then to make the body very rigid, the three pipes were connected with each other at two different points using two long, threaded rods. Figures 3-28 through Figure 3-33 show how the skeleton looked like after the final modifications.

Figure 3-28 The plastic piece that holds the gear to the pipe

Figure 3-29 The plastic piece that holds the gear to the pipe – from a different angle

Figure 3-30 The support for the device

Figure 3-31 The long threaded rod at the center of the body

Figure 3-32 The long threaded rod at the center of the bending point

Figure 3-33 The device with the long threaded rods and the plastic piece

The final weight of the body was 3.55 kg. That includes the skeleton, the base, and the Monster Torque. Figure 3-34 shows what components were included in the final design that resulted in the total weight of the body. In all designs we used inexpensive components that can be purchased at Home Depot, ACE Hardware or similar shops. This shows future students that they do not need expensive components or commercial robot brackets to achieve many tasks in android robotics.

Figure 3-34 Measuring the body weight

3.3.2 Rotating Motion

The goal of the rotating motion is to make the Curie robot interact with its surroundings during the robot theater performance. In particular, Marie Curie talks to Einstein and the Schrodinger Cat, or Bohr and Newton, which are all mobile, wheeled robots that surround her during their interactions. Adding this rotation is a unique requirement of the robot play Quantum Debate. It was built for some non-humanoid robots in robot orchestras, where the drummer played many drums. Adding the rotating motion was one of robot design requirements formulated by Prof. Perkowski.

The rotating motion gives the Curie robot the ability to track the Schrodinger Cat robot and to the other robots on the ground while they are moving around. Since there was a fixed base under the body, it made the rotating motion design very simple. It is similar to how a boat operates. A boat uses a motor to rotate to the left or to the right. By attaching a servo to the base of the body, the body could rotate by using a gear in the

fixed base. Figure 3-35 shows the two gears A and B, where A was the servo gear and B was the gear that was fixed to the wood under the body.

Figure 3-35 A was the servo gear and B was the gear that was fixed to the wood under the body (Norton, 1992)

This idea worked very well. However, because more space was needed in the body, the servo and the gear were switched. Figure 3-36 shows how the gear is fixed under the body.

Figure 3-36 The driven gear under the body

The servo was fixed to the wood under the body. However, these two components were not enough to balance the body. The gear was held by a long screw, washer, and nut as shown in Figure 3-36. Furthermore, a Lazy Susan device was used to make the body turn. Its diameter is 12 inches which makes it capable of carrying up to 1,000 pounds. Figure 3-37 shows the Lazy Susan.

Figure 3-37 The Lazy Susan

Attaching the Lazy Susan was complicated. Its center had to match the center of the body in order for the body to rotate on it. There should also be a precise measurement of where the servo gear should be placed in relation to the center of the Lazy Susan. The measurement of the servo position is an important step to make the servo capable of rotating the body in a full circle. Figure 3-38 shows the wooden base with the servo and the Lazy Susan.

Figure 3-38 The wood base (top view)

Attaching the body to the Lazy Susan is also difficult. As soon as one puts the body on the base, the Lazy Susan cannot be screwed to the body. The solution to this problem was to make a hole under the base so one could screw the Lazy Susan to the body from the bottom. Figure 3-39 shows the holes that allow the Lazy Susan to be screwed onto the body.

Figure 3-39 The two holes under the wood base

Two bearings and nuts were used to hold the shaft (long screw). These bearings ensure that the two bearings are not going to become too tight or too loose when the body rotates. Figure 3-40 shows the space between the body and the basement. In conclusion, using a Lazy Susan is a good idea for future humanoid robots because of its very low cost and the availability of several types of Lazy Susan devices.

Figure 3-40 The space between the body and the wood base

3.4 Building the Left Arm and Body Modification

This section discusses how I replaced the old arm of the Curie robot with a more realistic arm possessing all the DOFs to function properly. The design should allow the hand to grab and manipulate objects, which had never been achieved on full-size android robots in our IR Lab in the past. The arm should be strong enough to hold an item like a filled bottle and the motions should be sufficiently precise and replicable for use in the play. These tasks were not achieved in our lab in the last 4-5 years. Prof. Perkowski and I decided to build this part of the Marie Curie robot from some components of a robot

called InMoov. This decision was made after 14 years of trying to build such arms in the IR Lab – many unsuccessful designs of human-like arms with forearms and fingers were created, but none were good enough for the theater. Many designs were built but they were not firm and were very fragile. They trembled and they could not hold an object firmly. Please note that this topic is very close to hand prosthesis design. Until recently prostheses were very expensive because they required complicated machining. The availability of inexpensive 3D printing technology has created a revolution in prosthesis design and thus also in the design of robot arms. In 2014 the first award in Intel’s Cornell Cup competition was given to a 3D-printed prosthesis that was very similar to the design discussed here, but had a simpler control. Moreover, it was only a forearm prosthesis. Our design has one more DOF and is attached to a full arm with a shoulder and elbow, which is more complicated.

InMoov is a 3D-printed robot that was designed by Gael Langevin, a French sculptor and designer (Langevin, 2012). InMoov’s arm is very advanced compared to the Curie robot’s left arm design that comes from previous designs of PSU students, and many other hand and arm designs from the internet that Dr. Perkowski collected since 2001. Figure 3-41 shows the Curie robot’s left arm and figure 3-42 shows InMoov’s left arm.

Figure 3-41 The Marie Curie robot after adding the new body and new left arm. This design still lacked good female body proportions, so I used the InMoov design and 3D printing.

Figure 3-42 InMoov left arm (Langevin, 2012)

The observer can easily see and appreciate the difference between the two arms. The Curie robot's forearm is a replica of the InMoov forearm. The InMoov arm was proven to be reliable and robust after testing three designs in the last year in our PSU ECE IR Laboratory. For this reason, we decide that the Curie robot should have InMoov arms. The InMoov arms finally give the Curie robot what it was missing for years. They allow the robot to interact with its surroundings in a very natural way. The robot is also able to reliably manipulate objects of various sizes and shapes, as shown in other projects of 2015/2016 ECE 478/479 classes at PSU. Several video presentations in this thesis's oral defense illustrated the advantages of the new arms, as well as other parts of the Curie robot design created by me.

After deciding to use InMoov arms for the Curie robot, her body had to be modified to accept the InMoov arms. However, building a new 3D-printed body for Curie would take a long time. Therefore, parts of the InMoov Torso design were also adapted to the Marie Curie robot body. This gave us the ability to attach the left arm when it was 3D-printed, assembled, and tuned and to also use the InMoov neck design. In addition, our design was now ready to accept the right arm whenever it is completed. Figure 3-43 shows the left arm of the robot attached to the torso.

Figure 3-43 InMoov arm attached to the torso (Langevin, 2012)

When we found how to attach the left InMoov arm to the Curie body, we were ready to build the rest of the left arm. Fortunately, the left hand and the forearm for the Curie robot were previously built and they worked properly. Therefore, we had to build the bicep and the shoulder. Langevin made building InMoov very straight forward. He

has all the 3D parts ready for printing, and all the pictures for assembling the parts of InMoov are available on his website (<http://inmoov.fr/>). He also explains everything about the servos he used, how to modify them, and how to calibrate them. Figure 3-44 shows how the bicep and the shoulder looks like when they are attached to the forearm.

Figure 3-44 The bicep of the left arm

After the left arm was ready, the next step was to add all the servos and the grease to the gears. Next we attached the left arm to the body. Figure 3-45 shows the final view of the Curie robot.

Figure 3-45 Curie robot after adding InMoov left arm

The tradeoff in robot arm design was that the robot arm was built using plastic instead of metal. 3D printing technology permits one to build body parts in human shapes, which is a better technology for many components of an android robot such as fingers and shoulders. However, it is not necessary to print the entire body of the robot as was done for InMoov. Therefore, we mixed of our previous body and the InMoov torso design. This mixture gave us the robustness of the old body, which allows the robot to carry the arms, neck, and head, and the shape of the human arm from the InMoov design. The design of the arms, torso, and neck cost \$550, which includes \$350 for 10 big servos, \$120 for 12 small servos, \$60 for the filaments, and \$20 for other materials.

3.5 Building the Neck

The goal of adding a neck to the Curie robot was to make it track things with its head. The neck together with face and head make the Curie robot look lifelike and able to “look” at objects and people around it. However, the Curie robot never had a functional neck before. Since we used the InMoov torso to attach the InMoov arm to Curie body, the body was also ready to use InMoov neck. This neck has two DOFs that allow the Curie’s head to move in all directions, but it does not allow the head to lean to the right nor to the left. However, this additional DOF is not as important as the other two. It is planned to add it in the future. All the steps for building the neck are explained in detail on Langevin’s website (<http://inmoov.fr/>) where he shows all the details for 3D-printing and assembling the neck. Figure 3-46 shows InMoov’s neck that was used on Curie robot and Figure 3-45 above shows the neck attached to Curie robot.

Figure 3-46 InMoov robot neck (Langevin, 2012)

4. Using the Interactive Genetic Algorithm to Evolve Motion of Leg

After building the leg for the Curie robot was achieved, the next goal was to use an algorithm that finds the servo values which makes the leg able to kick the drum and generate a strong sound. Therefore, the IGA was used to find the position values of the servos that will make the leg capable of hitting the drum. IGA is an algorithm that uses the principles of the Darwinian-like Genetic Algorithm with random mutation and crossover operations to find adequate sequences of servos position values (Perkowski, Bhutada, Lukac, & Sunardi, 2013). Those values should get closer and closer to the solution with every generation of the IGA. The solution is not precisely defined, so the observer feedback on whether there was a sufficiently good drumming action is required. There are many variants of IGA and similar algorithms in the literature. In principle they are similar to a standard GA but it is characteristic for all variants of IGA that the human is involved in one way or another to create values of the fitness function that is used for feedback. The fitness function is the evaluation of every chromosome. The evaluation allows the algorithm to find the best two chromosomes that can be used to generate the other chromosomes. In general, IGA is used only in the area of computer art, for instance for paintings or sculptures. As far as I know, the paper by Perkowski, Sunardi, Riedl, Huffman, and Goetz (2013) and this thesis are the first attempts to use the IGA in the area of robot motion design for robot theaters.

Initially the algorithm depends on the size of the population, which is group of chromosomes. Each chromosome holds a movement for the leg. The size of the population for the algorithm used for the leg was set to four, so the algorithm can switch

to a new generation more frequently. After the leg movements of the four chromosomes has already happened, the program calls the fitness function that will ask the observer to enter his or her feedback, which means choosing the best two leg movements. They are called parents. This feedback is used to generate two new chromosomes, which are the children for the new generation. The generation of the new children is created using crossover and mutation. Because the Genetic Algorithm concepts are well-known, this thesis does not intend to elaborate their details or the process of selecting their parameters and evaluating them. The crossover splits each parent's chromosomes into two sections and switches them with its spouse, and the mutation takes an arbitrary bit and mutates it based on the percentage of mutation, which in our case was set to 5%. Therefore, we get two new children for the next generation who inherit the genes of their parents selected for reproduction based on top values of the fitness function. This new generation is the next four animations of the leg for which the observer decides if there is a winner who kicks the drum and generates a strong sound, or decides to generate a new generation that might have a winner. The algorithm successfully achieved the goal and found the servos values that generate a strong sound when the Curie robot's leg kicks the drum. Because this method, with its hardware switch based feedback, was so successful in the design, we decided to use this method or similar principles to design other motions for the Curie robot.

4.1 The Steps to Implement the Algorithm in the Program

- 1- Generate a random population that is used as the first generation.
- 2- Send the code to the servo controller board for the leg movement.

- 3- Ask the observer if there is a winner in this population. If there is a winner, then exit. If not, continue making new generations.
- 4- Evaluate the current generation by having the observer choose the best two members in the generation.
- 5- Cross over the two parents to form two new children.
- 6- Mutate the bits in each of the four chromosomes in the population.
- 7- Go back to 2.

4.2 Chromosomes

Two of the biggest problems with implementing the IGA were how the chromosomes should be implemented and how to encode the positions of the servos in the program. Therefore, we decided to convert the positions of the servos from decimal to binary which allowed us to add them together to create whole chromosomes. Since the servo moves 180 degrees, we needed the $\text{Log}_2 180 = 7.5$ bits and by taking the ceiling of that number we needed 8 bits to represent each angle of the servo. Therefore, the length of the chromosome would be $8 \text{ bits} * 5 \text{ servos} = 40 \text{ bits/chromosome}$.

In fact, the Pololu Maestro used to control the servos also allowed the user to set the speed and acceleration in addition to the position of the servos. However, because the leg was supposed to kick the drum for certain fast rhythms, we needed to set the speed and the acceleration to the maximum value.

4.3 Fitness Function

Since the IGA was used to control the leg, it needed human-robot interaction (HRI) to determine the best two chromosomes out of the four. The determination of the best two chromosomes allows the algorithm to generate two new chromosomes. The two chromosomes chosen are called parents of the next generation, and the two new chromosomes are called children of the next generation. Therefore, the circuit shown in

Figure 4-1 was built to allow a human to change the flow of the program by entering his or her feedback. The Circuit shows four buttons and each button represents one of the chromosomes. After the observer pushes one of those four buttons the program uses the chosen chromosome as the first parent for the next generation. Then the observer chooses the next chromosome that the program then uses as the other parent. After this, the program creates the new generation of behaviors based on the new parents. The program repeats this procedure with every generation if there is no winner that was able to kick the drum and generate a strong sound.

Figure 4-1 The Circuit that is used in the feedback of the algorithm

5. Using a Genetic Algorithm to Develop Manipulation Behavior for an Arm

Because of the successful use of a GA-like algorithm to develop the robot's leg behaviors, Prof. Perkowski and I decided to use a similar approach for obtaining high-quality arm motion, which was very difficult to program. In general, a GA is used to generate a solution for a problem where we do not know how to find a solution, but have criteria for evaluating the quality of solutions (Goldberg, 1989). We looked for a problem that we do not know its solution in order to use GA to solve it. We found that GA can be implemented in the scene of manipulating liquid containers in the play. However, a human does not know how to evaluate partial attempts to manipulate liquids in containers. The manipulation of liquid containers means that there is no human feedback to solve this problem, like when we used the IGA. Therefore, there is a need for feedback to adjust the solution, which is handled by the fitness function. We decided to create a new type of algorithm based on general principles of a GA, but in which the fitness function is calculated by special hardware with simple measurements.

The task was to make the Curie robot arm capable of balancing a cup or a test tube in the lab and not to spill liquids while mixing and manipulating. In addition, there should not be any human help involved into solving this problem, so that the Curie robot will learn how to balance the cup on its own. Therefore, a GA can be used to solve this problem, but a new way of calculating the fitness function was used. Also, the use of the GA for this task is not present in the literature for any type of feedback function, even one software-realized. However, similar solutions based on learning from repeated

measurements are known from industrial robotics. The algorithm was successfully able to give the Curie robot the ability to balance the cup without needing human help.

5.1 Closed-loop Control System

The closed-loop control system is used to get feedback from the cup to the fitness function and then uses the feedback to generate a better movement. We hoped that the new movement can balance the cup. Otherwise, the algorithm would keep running. To build the closed-loop control system we needed feedback from the cup that the system built by me was intended to control. This could be accomplished by having a feedback circuit that will tell the main loop about what happens to the liquid in the test tube (levels of liquid on various walls of the container). Figure 5-1 shows the block diagram of the system.

Figure 5-1 Closed-loop control system to control the liquid levels while manipulating the cup.

5.2 Feedback Circuit

In order to have correct levels of liquid in the cup, four wires were fixed in the cup at the same height, as shown in Figure 5-2. Then the ground wire was attached in the bottom of the cup. Each one of these four wires was a switch that would be closed if the water level reached that wire. If the cup tilted to the front, the north switch would be closed and its corresponding Light-Emitting Diode (LED) would turn on. These four switches are connected by four wires to input pins in the Microcontroller that saved the status of the switches, as shown in Figure 5-3. Figure 5-4 shows a screen shot of the

serial monitor of the Microcontroller software that read all the values of the four switches.

Figure 5-2 The cup from the inside. The leveling wires are attached to the side of the cup while the ground is attached to the bottom of the cup.

Figure 5-3 Feedback circuit that is used to get the level of the liquid in the cup

Figure 5-4 Snapshot of the GA code running to test each of the four wires in the cup.

5.3 The Steps to Implement the GA Algorithm in the Program to Control the Cup Motion

- 1- Generate a random population that is used as the first generation.
- 2- Send the code to the servo controller board for the arm movement.
- 3- Get the feedback from the circuit and save the fitness score for each chromosome.
If any of the chromosomes has a score of 100, then exit and call it a winner.
Otherwise, continue making new generations.
- 4- Choose the best two parents based on their score for the next run.
- 5- Cross over the two parents to form two new children.
- 6- Mutate the bits in each of the four chromosomes in the population.
- 7- Go back to 2.

5.4 Fitness Function for the GA

After every animation (arm movement), there were four values for the four directions (wires) that the microcontroller gets from the feedback circuit. If any of the wires is ON, the microcontroller knows that the liquid reached that level and gives zero points to that direction. However, if the liquid does not reach the wire, the microcontroller considers that direction to be OFF and gives it 25 points. These four directions values are added to get a score out of 100 points. The highest scoring chromosome was more likely to be chosen for the next run. If one of the chromosomes got 100 points, then it was the one that could balance the cup. Therefore, if there was a chromosome that could balance the cup, then the program stopped.

6. Tracking

Tracking movement is one of the most important basic functionalities expected from a humanoid robot. Especially for a humanoid robot meant to perform in a theater. A robot that can track its surroundings is more human-like. In addition, people feel that the robot is alive and that it can see them. The main goal of designing tracking abilities for our robot was to make the Curie robot track the face of the nearest person, and recognize and track objects. According to much of the literature, conference material and our own observations from the lab, tracking is a very important component of robot's "life-like behavior" and is one of the first input-output behaviors usually implemented on android robots.

However, there exist some limitations that make the robot unable to perform some tracking tasks perfectly. Those limitations can show the difference between a human and a robot. In addition, it is very important to evaluate those limitations to find where and when the humanoid robot cannot act like a human. There is a vast amount of literature on robot tracking and control system tracking that points out many of the issues.

Since tracking behavior should make the robot able to look at the nearest person's face, there is no need to use libraries like Open-source Computer Vision (OpenCV). OpenCV allows the robot to recognize objects and track them without moving the robot's head which is not my goal. The PrimeSense integrated circuit provides the data that allows us to get the position of the nearest person's face. Then the tracking algorithm can use this data to predict and move the robot's head to look at the person. OpenCV is useful if the robot has eyes that can move rather than a moving head.

6.1 Related Work – background for humanoid tracking

Many researchers have presented different ways of tracking a human face using various types of cameras. Microsoft Kinect is one of the cameras that uses depth to recognize the human body in the form of a so-called “skeleton” (similar to “stick-diagram”). Its ability to track a markerless human body (no marks on human body) is a huge benefit for researchers solving old recognition problems in a new way. Garstka and Peters have shown that they can use the depth information to track the human head (Garstka & Peters, 2011). They were able to project a 3D item on an even surface. The viewing of the item is based on the position of the viewer. The viewer of the scene can move around the projected item and see it from different angles. Raheja, Chaudhary, and Singal (2011) used the Kinect sensor to find both human hands. Moreover, they divided the hand into smaller sections using the depth image from the Kinect sensor and detected the center of the palm and the fingertips in these smaller sections. Kondori, Yousefi, Li, and Sonning (2011) created an algorithm that can sense head motion. Using the depth camera, they found the face and ran their algorithm that detects the six DOFs of the head.

Other researchers have used the AForge.NET software package for object recognition and tracking. Ondrej Lufinka (2015) used an AForge glyph to add the ability of object tracking to his mobile platform using Microsoft Kinect. The AForge glyph gave the robot the ability to recognize objects and follow them. An AForge glyphs was also used by Redlarski, Pałkowski, and Ambroziak (2013) whose algorithm used the AForge glyph to recognize and locate several mobile robots. They proved by experimentation that their algorithm is capable of detecting up to ten glyphs at once.

6.2 Robot Tracking Scene

Since the robot that this thesis talks about called Marie Curie robot, who is a chemist, the robot should perform a scene that is related to chemistry. Therefore, the Marie Curie robot starts performing some scripted or interactive behavior when someone approaches it. The robot waits until the Kinect sensor is able to see a human's depth image or a human's skeleton image. After that, the Curie robot moves her neck to track the face of that skeleton and waves to the person represented by this skeleton. This makes the person feel that the Curie robot is aware that there is someone standing in front of her. Next, she starts the chemical experiment.

While she is doing something with her hand, she is using the AForge glyph tracking to track the cups that are labeled by different glyphs. She is able to tell if the cup is full or empty. If the cup is upside down, she will know that the cup is empty. Otherwise, she sees the cup as full.

6.3 Tracking Algorithm

A new algorithm has been created for the purpose of this thesis. The algorithm is triggered when a human is detected by the Kinect sensor. When the Arduino board, which is a microcontroller that controls the robot's movements, that controls the whole robot is connected, the algorithm is activated. Based on the head position taken from the depth camera, the algorithm directs the robot's neck to move. The movement of the robot's neck allows the robot to look at the closest human face.

The way the robot keeps looking at the human face is by making sure that human head is in the robot's sight of view. The robot's sight of view is a rectangle placed at the center of the frame. Figure 6-1 shows the rectangle inside the video viewer.

Figure 6-1 The robot's sight of view in the red rectangle in the middle of the picture

Things get more complicated for tracking the AForge glyph. Since the size of the glyph is very small, the robot's sight of view is going to change according to how far or how close the glyph is located. This is an important feature because when the glyph is far from the sensor it is harder for the robot's software to keep the glyph centered.

6.4 Using Kinect for Face Tracking

The Kinect sensor is a peripheral device used to control the Microsoft Xbox 360 and Xbox One. Players use their body language that Kinect recognizes and transforms to commands that the console can understand. By using the depth information, the console sees the players and reads their body language. Researchers use it to control other devices and systems like robots and home lights. Since Kinect is affordable, it can be embedded

into most electronic systems. Thus, Kinect improves the ways that people communicate with various devices. Figure 6-2 shows the Kinect Sensor (Zeng, 2012).

Figure 6-2 The Kinect sensor (Zeng, 2012)

The Kinect sensor consists of three main elements that are noticeable when you look at the device. The first element is the Infrared laser projector that distributes a pattern of infrared dots to the scene that is seen by the infrared camera. The second element is the IR camera that sends out a 640×480-pixel depth image. It is a monochrome complementary metal oxide semiconductor (CMOS) sensor. The last element is the RGB camera that sends out the color image with the same resolution as the depth image. Figure 6-3 shows the Kinect without the cover for a clear look at the sensors (Garstka & Peters, 2011) (Zeng, 2012).

Figure 6-3 The Kinect sensor without the cover (Zeng, 2012)

Inside the Kinect sensor is the PrimeSense integrated circuit. It uses the information about the IR dots to figure out the distance between the Kinect and the object. Figure 6-4 shows the internal design of Kinect. This chip allows maximally simplifying many motions and data capture algorithms.

Figure 6-4 Internal design of the Kinect (Raheja, Chaudhary, & Singal, 2011)

The depth image is a gray scale image. It shows the distance from the Kinect sensor to the object by changing the brightness of the pixels. This means that the pixels get darker when the object is closer to the camera. However, if the object is very close or very distant, the pixels turn to black which indicates that there is no depth information. Also, pixels get a black color if they do not correspond to any infrared dot (Zeng, 2012).

6.5 Using Kinect for AForge Glyphs Tracking

As mentioned above, the robots in the Portland Cyber Theater, in contrast to robots in other robot theaters, are interactive and they interact with humans and other objects. For instance, the Curie robot manipulates chemical devices and liquids. Thus, the robot should be able to find, grasp and correctly manipulate objects. It has to know where each liquid is currently located on its laboratory desk. Therefore, another important goal of the tracking behavior is to make the Curie robot able to recognize objects and track them, so it can perform the grasping task. Therefore, there should be a way to label the objects that the Curie robot wants to grasp. One of the best algorithms to label the object is AForge.NET. AForge.NET is an open-source framework for researchers who use C# language. It can be used in vision and artificial intelligence areas where objects that need to be recognized and characterized are labeled with AForge glyphs. The robot theater designer can design glyphs, print them out on paper and place them on the object that he wants the vision system to recognize. The only limitation is that the glyph has to be squared and surrounded by a white border. In addition, the first and the last row and column have to be black, as shown in Figure 6-5. Furthermore, each row and column (except the ones on the edge) has to have at least one white cell.

Figure 6-5 A group of AForge glyphs (Kirillov, 2006)

Glyphs have some great features that make them a great addition to any vision project in which high reliability and precision of motion is demanded. In addition to the unique shape that every glyph has, there is a way to know the orientation of the. Figure 6-6 shows two robots moving around on stage observed from above where the AForge glyph is shown atop of each. Every robot has its own name, coordinates and rotation angle (orientation) (Redlarski, Ambroziak, & Palkowski, 2013).

Figure 6-6 Two robots moving around (Redlarski, Ambroziak, & Palkowski, 2013)

As shown in Figure 6-7, there is no need to have the glyph perpendicular to the camera. Several glyphs can be laid on a ground and the camera can still recognize them.

Figure 6-7 AForge glyph on the ground (Kirillov, 2006)

Any camera can be used to find the AForge glyphs. Since the Kinect sensor is used to track the face, it can also be used to track the AForge glyphs. However, there should be some adjustments to the image taken, so the AForge glyph algorithm can detect the glyphs. Once the glyph is recognized, the main application, shown in appendix D, obtains all the information associated with the recognized glyph and draws a quadrilateral shape around it. Then the X and Y coordinates of the glyph can be found for the recognized glyphs. The X and Y coordinates of the glyph are used in the tracking algorithm to give the Curie robot the ability to track several glyphs located around the robot, even at different heights. The Curie robot's software uses the same algorithm that was used to track faces. However, the size of the glyph becomes smaller when it is far from the Kinect sensor. For that reason, the viewing sight of the robot becomes narrower when the glyph is far away from the sensor in order for the face to look exactly at the glyph.

Since the size of the quadrilateral is changed according to the size of the glyph, it can be used to find the distance between the sensor and the glyph. Assuming that the

actual size of the printed glyph is 2'' by 2'', one can measure the distance between the glyph and the Kinect sensor. Thus, the size of the quadrilateral shape corresponds to the distance between the Kinect sensor and the glyph.

6.6 Evaluation of Tracking

It is necessary to test the robot's visual abilities for the Marie Curie project and how close the robot's operation is to that of a human being. To test these things two experiments have been done. The experiments test the motions of the two DOFs in the robot's neck, and they are used to compare the robot's neck movements to human movements. The first experiment started by having the robot stand in a fixed position while a person moved in front of the robot. The robot would then try to track the person once the person left the robot's sight of view. While the robot tracked the person, the main application counted the time it takes the robot to find the person in milliseconds. The person moved to five points marked on the ground. These points were chosen based on the angle of rotation of the robot neck, and numbered from one to five.

In order to design the experiment area, some measurements for the robot's neck movements need to be taken. First, measuring the angle of the neck rotation, which is simply how far the robot can rotate its face to the right or to the left. The robot has 90 degrees of rotation (45 degrees for both sides). The robot can also move its neck up and down a total of 36 degrees (18 degrees for each way). Those angles were measured using a printed protractor for the horizontal angle and a digital angle gauge for the vertical angle. The digital angle gauge was fixed on the top of the head to allow the device to measure the angle of the up and down head movements, as shown in Figure 6-8. The

protractor was printed and placed in a position that allows the observer to see the angle of head rotation. Figure 6-9 shows the protractor that measures the head rotation angle.

Figure 6-8 The Digital Angle Gauge

Figure 6-9 The protractor

The robot uses these two DOFs in the neck while moving its face. Since they must move together, there was no reason to test each one separately. Therefore, the horizontal angle was split into four segments of 22.5 degrees. The segmentation of the horizontal angle allows us to have five points to where the tracked human can move. These points are at 45, 67.5, 90, 112.5, and 135 degrees, respectively.

After all the measurements for the robot's neck were completed, the robot was placed in a large area. Then a person stood facing the robot to measure the furthest point where the Kinect sensor could still recognize the person. With distances longer than 234 cm, the Kinect sensor will start to lose the person. Therefore, there was a distance of 234 cm between the Kinect sensor and each of the five points where the person stood. Figure 6-10 shows an overview of the experiment area.

Figure 6-10 An overview of the experiment area where the five points are placed according to the segmentation of the neck rotation angle

The main application was modified to count the time the robot took to find the person again. The counter started counting in milliseconds when the person left the robot's sight of view. The counter kept on counting until it found the person again. Then the observer wrote down the time that it took the robot to find the person.

In both experiments the person started at point 1 and finished at point 5. However, the difference in the person's movements was in the moving distances. In the first experiment the person moved one point at a time. This means that the person moved from point one to point two and then the time was recorded. After that the person moved from point two to point three and then the time was recorded again. The person kept on moving until the person reached point five. This gave us four uniform movements that do not change when the experiment is repeated. This tests the robot's ability to track the person and the time it takes to find the human again. Figure 6-11 helps visualize how experiment one worked.

Figure 6-11 The movement of the person in experiment one

In the second experiment the person still moved from point one to point five. However, the person skipped points two and four. The goal was to test the time that the robot takes to find the person and whether a long distance affects the robot's system tracking ability. Figure 6-12 helps visualize how experiment two worked.

Figure 6-12 The movement of the person in experiment two

After both experiments were completed, a human was put in the robot position. Then the same two experiments were done with a human. The human wore the Kinect sensor on its head and used the same tracking algorithm as the robot. The human tried to see the moving subject through the Kinect camera, and once the subject moved, the human tried to track it. The time was then recorded and compared with the robot's tracking time.

6.7 Results of the Evaluation of Tracking

The experiment was repeated several times and a significant amount of data was gathered and made ready for comparison. The first group of data is for experiment one. Table 6-1 shows the time (in milliseconds) that the robot took to find the person at each point.

Table 6-1 Experiment one for the robot

Run number Person's movements	1	2	3	4	5	6	7	8	9	10	average
1 -> 2	144	158	194	208	428	340	286	244	210	246	245.8
2 -> 3	82	198	374	198	194	168	174	216	264	160	202.8
3 -> 4	148	324	330	234	106	172	188	196	258	182	213.8
4 -> 5	150	274	154	258	274	440	158	182	366	208	246.4

The next group of data is for experiment two. Table 6-2 shows the time (in milliseconds) that the robot took to find the person in each movement.

Table 6-2 Experiment two for the robot

Run number Person's movements	1	2	3	4	5	6	7	8	9	10	average
1 -> 3	530	464	566	472	404	328	464	474	628	574	490.4
3 -> 5	424	284	412	394	340	406	508	466	558	422	421.4

After the two experiments for the robot were completed, the human was placed in the robot's position. Then the two experiments were run again. The following two tables, Table 6-3 and Table 6-4 present the data that were gathered from the program for the human tracking another person.

Table 6-3 Experiment one for the human

Run number Person's movements	1	2	3	4	5	6	7	8	9	10	average
1 -> 2	180	272	224	288	296	102	188	156	266	142	211.4
2 -> 3	138	120	132	228	296	318	156	226	258	158	203
3 -> 4	242	122	300	144	262	338	146	204	158	232	214.8
4 -> 5	230	208	158	262	174	162	138	144	168	262	190.6

Table 6-4 Experiment two for the human

Run number Person's movements	1	2	3	4	5	6	7	8	9	10	average
1 -> 3	564	376	594	510	438	690	324	560	370	552	497.8
3 -> 5	294	278	616	504	580	536	374	450	532	592	475.6

After the experiments for both the robot and human tracking algorithm were completed, the data were ready to be analyzed. The following Figure 6-13 shows the line graph that illustrates the average time that both the robot and the human take to track the person in experiment one.

Figure 6-13 Experiment one analysis

By taking a glance at the average time for experiment one for both the robot and the human, the reader can see that they take almost the same time to track a person. However, the robot takes slightly more time to track the person again when the person moves to the furthest point on either the left or right side. This happens because the robot is slower in moving its up and down DOF. There is a difference in the mechanical design of the robot's neck and that is what causes the time to go up slightly. When the person is at point one and wants to move to point two, the robot moves its face down then left to track the person. In fact, when the robot moves its neck down, it uses the vertical gear which is slower than the regular gear which is used with movements left and right. Figure 6-14 shows the vertical gear and the Figure 6-15 shows the regular gear. The same thing happens when the person moves from point four to point five, but with even more

additional time. The additional time comes from going up instead of going down. That makes the servo slower because it is lifting weight.

Figure 6-14 The vertical gear surrounded by the blue rectangle is responsible for moving the head up and down

Figure 6-15 The regular gear surrounded by the blue rectangle is responsible for moving the head left and right

The other part of the evaluation regards the second experiment where the person is moving for a longer distance. Figure 6-16 shows the line graph that illustrates the average time that both the robot and human take to track the person in experiment two.

Figure 6-16 Experiment two analysis

In the second experiment both robot and human take almost the same time to track the person from point one to point three. However, when the person moves from point three to point five, the robot finds the person faster than the human. This is a very interesting result and it shows how the controller can affect the smooth movement of the servos. The Pololu Mini Maestro Servo Controller sets the acceleration and the speed of the servo. It has been adjusted, so the robot's head motion is smooth. This adds more life to the robot and makes it look more like a human being. Therefore, the smooth motion can be counted as another reason for the higher time the robot took in the first experiment when tracking the person from point one to point two and from point four to point five.

However, the robot was faster than the human when the distance became longer. The lower acceleration makes the robot move smoothly and slower in tracking the person in short distances. Nevertheless, the higher speed makes up for the time lost in the initial movement and allows the robot to go faster than a human.

The last point that has to be mentioned is the limitation of the robot neck compared to a human neck. The robot neck was designed with two DOFs, but the human being has three DOFs. Adding the third DOF to the robot neck will make the robot reach more angles. Moreover, the robot cannot look up or down as much as a human being. It can move 18 degrees up or down while the human neck moves further.

7. Grasping

Grasping is another important task that shows the robot behaves like a human. Grasping actions are common in the Portland Cyber Theater, such as when Einstein gives a book to Newton or Curie grasps a probe. Grasping behaviors add realism to scenes and are appreciated by the audience. Among robot theater robots, only the Honda robot is able to execute grasping behaviors. It will be a nice component of the Portland Cyber Theater that the robots will execute various grasping behaviors realistically and reliably in order to get the viewer attention. For the purpose of this work, the IK is used to move the robot hand to the object, which mimics the way humans move their hands. Since the object's position is known, calculating the IK is enough to get the hand to the desired coordination. The advantage of using IK over other approaches is that they can be applied dynamically to every starting and terminal point of motion, which makes them universal in contrast to learning one particular grasping motion as Curie learned the shaking or drumming actions in previous chapters.

7.1 Related Work – Background on android robot grasping

Researchers have worked on the grasping problem for a long time and they have proposed different ways to solve different variants of the grasping problem. Solving the IK is different for each robot and researchers are approaching this problem from various perspectives. This shows how important the IK is. Researchers discussed an approach for solving the IK for a robot that has more than one DOF (Neppalli, Csencsits, Jones, & Walker, 2008). By knowing all the end-points for the trunk section, they compensate the difference in orientation to get a solution for the IK. Another advanced approach was presented by Wesam Mohammad Jasim (Jasim, 2011). He solved the IK for four DOFs

SCARA manipulators. The Adaptive Neuro Fuzzy Network (ANFIS) was implemented using matrix laboratory (MATLAB) to solve the IK problem in a very short time.

Researchers showed that the IK can be solved using geometric solutions. Adelhard Beni Rehiara (2011) used the geometric method to solve the IK problem. In his book he showed several methods to solve the IK problem. For the purpose of this thesis, the geometric solution has been used to solve the IK for two DOFs arms. However, this thesis shows the calculation of IK using the geometric method with more details for our humanoid robot arm from the InMoov robot. The obtained motion has been tested and discussed in the following chapters. It can be used as a starting point for future electrical engineers looking to build their own grasping behaviors for various robot arms. In contrast to many advanced papers on IK written by theoreticians in mechanics, electrical engineers have no background in advanced mechanical engineering methods such as Lagrangians. Consequently, giving electrical engineers a simple, detailed example without complex background material and notations is important.

The evaluation method has been inspired by a paper that analyzes human motion based on what Kinect can see. Ronald Poppe (2007) discussed the analysis of a markerless human body. He presented some modeling definitions that can be used as a reference to analyze human motion (Poppe, 2007). The author created the human body model and next analyzed how to use the kinematic model based on it. This comparison gives us a way of human body modeling that the Curie robot can be compared to.

7.2 Robot Grasping Scene

This part completes the scene, which started in section 6.2, that the Curie robot does as a chemist. After Curie tracks a person and waves to them, she starts an experiment by grasping one of the cups and pouring its contents into another cup. Then she picks up the second cup and shakes it. Finally, she pours the contents of the second cup into the third one. Each cup has to have a different substance that will react with the substance in the other cups.

7.3 Using Inverse Kinematics

Calculating the kinematics for a robot is an important step to write efficient software that will control the robot movements. In other words, when the IK is solved, the designers have an easy way to control the robot movements. The goal of using IK is to give the robot the ability to move its arm based on where the end-effector (hand) is in near space.

In order to move the end-effector to a desired point in the space, the software needs to calculate some joints angles. These joints are the DOFs that give the robot arm or leg the ability to move its hand or foot. In other words, they are the movements of the shoulder, elbow, wrist, hip, knee and ankle. They are different from each other. For instance, there are three DOFs in the shoulder while there are two DOFs in the elbow. The shoulder can move the whole arm away from the body, up or down, and right or left. However, the elbow can move the arm in two directions up or down, and rotate the arm to left or to the right. Figure 7-1 shows the DOFs of the shoulder and elbow.

Figure 7-1 DOFs in the shoulder and elbow (Abdullah, Tarry, Datta, Mittal, & Abderrahim, 2007)

There are two types of kinematics. The first type is the forward kinematics (FK) which finds the position of the end-effector in the space using joint angles. This means that the software knows the angle of each joint and it uses them to figure out the coordination of the end-effector in the space. Figure 7-2 shows an explanation of the FK.

Figure 7-2 FK explanation (Wyeth & Milford, 2011)

The other type of kinematics is called the IK. It is the exact opposite of the FK. It finds the joint's angle using the position of the end-effector in the space. In other words, it uses the position of the end-effector in the space to find the joint angles. Figure 7-3 shows an explanation of the IK.

Figure 7-3 IK explanation (Wyeth & Milford, 2011)

This type of kinematics is similar to how humans move their arms or legs. People start moving their arms by locating the object they want to reach and move their arms toward that object. What they do not know, is that their mind is changing the angles of their arm joints to move their hands. In addition, our mind gives new instructions to the arm muscles to shrink and stretch in order to change the angle of the joints until we catch the object. However, some objects are not stable which means that our mind has to get the new position of the object and calculate the new angles for our joints. We also use the FK, but only for very small tasks. In addition, FK does not need to use any new information from our eyes. For example, when we type on a keyboard, we already trained our unconscious mind to memorize the angles of our fingers to each key using IK. After some training we can memorize those angles and use FK to press those keys that we trained on. This is confirmed when we change to a new keyboard and then press a wrong

letter. This happens because the key position was changed slightly and the memorized angle for that key is not working any more. Therefore, we need to use the IK to set new angles for the key positions in our unconscious mind (Morasso, Casadio, Mohan, Rea, & Zenzeri, 2015).

Since we care about moving Curie's arm to grasp an item, all the calculations here are for Curie's arm. In addition, we only need the IK since we know the location of the item we want to grasp.

The InMoov arm has three DOFs in the shoulder and one DOF in the elbow. Using two DOFs is enough to prove that IK can control the Curie robot's arm. The first DOF is in the shoulder that moves the arm up and down and the second one is in the elbow. Having two DOFs allows the robot hand to move in the two dimensions. This means that the hand can go forward, backward, up, and down.

Before we started the calculation for the IK we needed to take some measurements for the arm. The first one was the distance between the shoulder and the elbow, which is 23cm, as shown in Figure 7-4. The second measurement was the distance between the elbow and the end-effector (the middle of the hand), which is 37cm, as shown in Figure 7-5.

Figure 7-4 The distance between the shoulder and the elbow

Figure 7-5 The distance between the elbow and the end-effector

7.3.1 Geometric Method

One of the ways to solve the IK for two DOFs is by using the Geometric method.

It applies the Cosine rule to find the angles of the two joints. The following Figure 7-6 shows the geometry of the link between the two joints in the arm.

Figure 7-6 Geometric of two DOFs robot

In order to imagine the geometric graph on the robot, Figure 7-7 shows the right side of the robot with the graph that was drawn on top of it.

Figure 7-7 Drawing the geometric of the two DOFs for the robot

The cosine rule is used with the triangle to find a side or an angle. Figure 7-8 shows an example of the triangle followed by the equations that can be derived from it.

Figure 7-8 Triangle

$$a^2 = b^2 + c^2 - 2(b)(c) \cos \emptyset$$

$$b^2 = a^2 + c^2 - 2(a)(c) \cos \emptyset$$

$$c^2 = b^2 + a^2 - 2(a)(b) \cos \emptyset$$

If you look at Figure 7-6 above, you will see that Theta2 (Θ_2) with Phi (Φ) is the 180 degree. Therefore, to find Θ_2 we first need to find the value of Φ . To find Φ , we need to use the cosine rule. The following equations use the cosine rule to find Φ .

$$\Theta_2 = 180 - \phi$$

$$d^2 = 230^2 + 370^2 - 2(230)(370) \cos \phi$$

$$\text{Since } d^2 = Xc^2 + Yc^2$$

$$\therefore Xc^2 + Yc^2 = 52900 + 136900 - 170200 \cos \phi$$

$$\cos \phi = \frac{189800 - Xc^2 - Yc^2}{170200}$$

$$\text{Let } D = \frac{189800 - Xc^2 - Yc^2}{170200}$$

$$\therefore \phi = \text{atan2}(D, \pm \sqrt{1 - D^2})$$

$$\therefore \theta_2 = 180 - \text{atan2} \left(\frac{189800 - X_c^2 - Y_c^2}{170200}, \pm \sqrt{1 - \left(\frac{189800 - X_c^2 - Y_c^2}{170200} \right)^2} \right)$$

We can use the above equation by substituting for the new XY coordination to find the angle of the elbow. Based on the value of Θ_2 we can find Θ_1 . The following equations derive the equation of Θ_1 with known XY coordinates.

$$\theta_1 = \beta - \alpha$$

$$\beta = \text{atan2} (X_c, Y_c)$$

$$\alpha = \text{atan2} (230 + 370 \cos \theta_2, 370 \sin \theta_2)$$

$$\therefore \theta_1 = \text{atan2} (X_c, Y_c) - \text{atan2} (230 + 370 \cos \theta_2, 370 \sin \theta_2)$$

Knowing Θ_2 , we can find Θ_1 for the new XY coordinates using the above equation. This means that the robot is ready to receive the new angles for the joints and to move the whole arm to reach the desired point in the near space.

7.4 Reachable Coordinates

The equation to get the two angles of the shoulder and the elbow are ready, so the next step is to run a program that will calculate the angle for every coordinate. This program will give us the angles of the elbow and of the shoulder and will show the reachable coordinates and the unreachable coordinates. It will also show the arm's limitations and how far it can go.

In order to do this, the servo value for each angle must be known. Otherwise, there is no way to move the servo to the desired angle. To find each angle, the digital

angle gauge was used to measure the angle of each pulse width modulation (PWM) value that is sent by the Pololu to the servo. Figure 7-9 and Figure 7-10 show how the measuring tool was placed to measure the angle of the shoulder and the angle of the elbow.

Figure 7-9 Measuring the angle of the shoulder

Figure 7-10 Measuring the angle of the elbow

Measuring the angle for the shoulder and the elbow may lead to confusion because the value of the servo is not very accurate. This means that increasing the PWM value by ten will keep the servo in the same position and increasing the PWM value by twenty may make the servo jump to a far point. However, using the Pololu Maestro Controller, which controls the servos, gave us the advantage of setting the value of the acceleration and the speed of the servo. Having a lower speed and acceleration value prevented the servo from jumping to a far point.

After adjusting the acceleration and the speed of the servo, the servo gave accurate movements for every five-degrees increment in the angle. Therefore, the

reachable angles for the shoulder started at -20 degrees and go up by 5 degrees. The last angle that the shoulder can reach is 80 degrees. The elbow has a narrower angle; it starts at 22 degrees and ends at 87 degrees. It also has increments of five degrees like in the shoulder. Table 7-1 and Table 7-2 show the look-up table for the shoulder and elbow angles that link each angle to its corresponding servo PWM value.

Table 7-1 The look-up table for the elbow angles

Elbow angle	Servo PWM value
22	1000
27	1033
32	1073
37	1144
42	1209
47	1282
52	1363
57	1439
62	1522
67	1608
72	1688
77	1777
82	1880
87	2000

Table 7-2 The look-up table for the shoulder angles

Elbow angle	Servo PWM value
-20	1000
-15	1050
-10	1100
-5	1150
0	1200
5	1250
10	1300
15	1350
20	1400
25	1450
30	1500
35	1550
40	1600
45	1650
50	1700
55	1750
60	1800
65	1850
70	1900
75	1950
80	2000

Since we know the reachable angles, it is possible to know the reachable coordinates. The coordinates that are reachable were found by filtering the equation result using Microsoft Excel. Figure 7-11 shows the reachable angles for the shoulder and the elbow. The reachable angles by the shoulder are colored by blue and the reachable angles by the elbow are colored by green.

Figure 7-11 The calculation for the reachable coordinates by the arm, and the snapshot shows the area that the arm can move in where the blue and green cells intersect

The horizontal axis is the Y-axis which starts at 0 mm and ends at 600 mm. The vertical axis is the X-axis that starts at -310 mm and ends at 600 mm. The negative part of the X-axis is where the hand reaches the coordinate above the shoulder line. By taking a close look at the snapshot of the Excel sheet, one can notice that the intersection of the two colors, which is where the hand can reach. Each one of them has a different angle for the shoulder and the elbow.

In conclusion, the Curie robot is now able to control the movement of its shoulder and elbow based on the position of its hand in the space. Therefore, the goal of using inverse kinematics to mimic the human arm movement was achieved. The next step was to analyze how close the arm movement is to the movement of a human arm.

7.5 Evaluation of Grasping

The evaluation method for this part of the thesis is done by comparing the DOFs used in the robot to the same DOFs in the human model. The main task was to measure how close the movement of the Curie robot's arm is to the movement of a human arm. Since the Curie robot is a humanoid robot, the best method for evaluating the arm movement was comparing the range of motion of the used DOFs with the range of motion of the corresponding human DOFs.

The kinematics calculations were used for the Curie robot with only two DOF, which are the shoulder and the elbow. The shoulder has 100 degrees that were split into two sections. The first section is the 80 degrees that allows the arm to move from a resting position to being raised all the way up. The second section is the 20 degrees that allows the arm to move behind the body. This range of motion can be changed through the servo potentiometer that is placed behind the shoulder. When comparing the robot shoulder movement with the human shoulder movement, we can see the limitations that the robot has and why the robot's hand reaches fewer coordinates. Figure 7-12 shows the angles of a human shoulder.

Figure 7-12 The angle of a human shoulder (American Academy of Orthopedic Surgeons, 2011)

By looking at Figure 7-12 one can see that a human being has a wider angle for shoulder movement than the Curie robot. This deficiency has a huge impact on the coordinates reachable by the robot. In other words, the robot's hand cannot reach all the points that a human's hand can reach using this DOF. Furthermore, the human reaches 140 degrees more than the robot's hand can reach. It affects the IK calculations that show the advantage of the human arm over the Curie robot arm.

However, the elbow, which was also used in the calculation of the IK, is a part of the robot arm that allows the hand to reach more coordinates. For the Curie robot, the elbow can rotate up to 65 degrees. Compared to a human elbow that can rotate up to 150 degrees, we can see this holds the robot back from reaching more coordinates with its hand. Figure 7-13 shows the angle of a human elbow. This observation can help us define the problem in order to make the robot reach the human levels of movement.

Figure 7-13 Human elbow angle (American Academy of Orthopedic Surgeons, 2011)

Since the internal design of the robot shoulder's gears has an endless rotation, the robot is able to rotate its arm 360 degrees. Therefore, the only limitation left is the potentiometer. By using a wider angle potentiometer, the designer can design a robot with a higher shoulder angle.

The robot elbow design has a trade-off for the robustness of the elbow design. Since it is the link between two main parts of the arm, it should be strong enough to carry the load for the forearm, the hand and the carried item. Gael Langevin designed this part of the arm with one bolt screw that allows the elbow to move up and down. In addition, he used a big servo (Hitec HS-805BB) to motorize the joint and allow the software to control it. This lost the joint some degrees of rotation. Consequently, there is another effect of the IK based motion analysis, which results in even fewer reachable coordinates.

Overall, the goal of using IK was to make the robot move its hand like a human with two DOFs. This goal was achieved in the presented work. However, the limitation in the design of the used DOFs made the robot's hand unable to reach as many coordinates as a human hand can reach. Even though the arm design of the Curie robot looks similar to the model that satisfies the goal of building the humanoid arm that was described in chapter three, the Curie robot's arm has less reachable coordinates. This deficiency is because of the smaller rotation angle that the Curie robot has in both its shoulder and elbow joints. However, calculating the IK for the remaining DOFs may give the robot arm the ability to reach more DOFs and make the arm more functional. This example explains how the phases of mechanical design, kinematics design and even software motion design of a simple robot arm are related. This example should give the reader some appreciation for how much more complex and difficult solving the IK would be if one were designing the entire robot where the robot would also be expected to walk.

8. Conclusion

Building a humanoid robot that has to look and act like a human is a very complicated process because one has to consider many criteria and behaviors. Those behaviors show the complexity of building a humanoid robot. In this work, I built a complete robot for Portland Cyber Theater that will play the character of Marie Curie from existing screenplay “Quantum Debate”. The Curie robot is able to do behaviors that might be useful for other purposes and not only for the Portland Cyber Theater. For example, future students can animate several interactive behaviors like hand shaking. Overall, The Curie robot can be the foundation for many humanoid robots in the future. Developers can use ideas or designs presented in this thesis to make other robots humanoid or develop a new behavior. Since partial comments on evaluation were given in the previous chapters, this chapter evaluates the entire integrated robot according to the criteria formulated in the introduction.

The criteria will be repeated below and short answers are given for each.

1. Design a humanoid, android robot with natural body proportions to play Marie Curie in the robot theater. The robot should be sitting on a desk, and should have movable hands, legs, neck, head and body.

As illustrated in figures and videos, the new robot has human female body proportions of a person with a height 160 cm. The robot sits at a desk, but it can move its legs, arms, and head as desired. Its body proportionality is similar to the top android robots from the internet.

2. The robot should move similarly to humans, should be able to track humans and objects and grasp objects.

Drumming and kicking leg motions and various arm motions such as waving, grasping and liquid raising were designed. Tracking for neck motion is realistic and is reminiscent of human behavior, as can be seen in videos. Comparisons with human tracking have been done. Based on comparisons of behaviors of known theater robots, the behavior is comparable to the top behaviors and sometimes even exceeds them. For instance, the commercial robot RoboThespian cannot grasp objects and bend to reach an object while the Marie Curie robot can. There are currently no papers that evaluate theatrical robots. Comparing our robots using criteria developed for assistive robots would take a lot of time (Sunardi, 2010). These criteria will be created in the future.

3. The robot should be inexpensive, at most \$2,000, and the construction should be easy enough to be reproduced by undergraduate students and high school students by using the description created in this work.

The robot costs around \$1,000 and the filament only costs \$60. One high school student already printed arms, and a few undergraduate students completed several replicas of the robot's components. With this thesis and a separate manual that being prepared, every student will be able to print and assemble robot arms. Legs can be reproduced with combined bracket/3D printing technology. Heads are more complicated at this point, but head/face gestures were not a subject of this thesis.

4. The robot should be programmed in C# from scratch. It should not use complex software packages such as Orange, Open-source Computer Vision (OpenCV), or Robot Operating System (ROS) which are difficult to use for non-experienced programmers.

All algorithms were written by me from scratch in C#, and complex libraries such as OpenCV or PCL and “robot operating systems” such as ROS were not used. To compare these algorithms with other robots’ algorithms such as Mr. Jeeves, which uses OpenCV and ROS for tracking, would take a lot of time and was deemed unnecessary because OpenCV provides unused data. OpenCV is useful when the robot has eyes that can track objects without needing the robot’s head to move. The comparison between the tracking systems of Mr. Jeeves and Marie Curie robots will be done when both robots are ready.

5. The system should be complete and integrated, so that the next users and designers can program on top of the software developed in this thesis. The first users will be PSU students from Intelligent Robotics classes.

The system is integrated and a new group of students will soon start working on it. The thesis and software developed in it, plus additional documentation, will be used to expand and copy the robot. Collaborating with these new students will allow me to improve the quality of the entire documentation.

Now that the above list show that the goals of this research have been accomplished, let us present in detail some other issues.

1. The **main goal** of this thesis was to build a humanoid robot called Marie Curie that can perform the behaviors needed for the Portland Cyber Theater. The Curie robot is not a general purpose robot, but a robot for specific tasks from a given script.
2. In order to achieve the main goal, it was divided into **sub-goals**. The sub-goals were mechanical and software designs. In humanoid robotics mechanical tasks are closely related to designing elementary motions of components. Therefore, they are treated here as parts of the mechanical design. As we are not designing a sculpture but an android robot, a mechanical design is useless if it is not able to execute basic motions in a robust and human-like way.
3. The first **mechanical goal** was to build the **leg** that gives the Curie robot the ability to do some entertaining behaviors like play a drum or kick the non-obedient Schrodinger Cat robot. The design of the leg has five DOFs that were controlled with the IGA. The algorithm was successfully able to achieve the goal and find the leg controls that generate a strong sound when the Curie robot leg kicks the drum.
4. The second mechanical goal was to build a **body** for the Curie robot that is robust enough to carry the neck, the head, and the two arms with two DOFs. In addition, the two DOFs make the body capable of bending forward and rotating to the left and to the right. Furthermore, the goal of the bending motion was to give the Curie robot the ability to look at the Schrodinger Cat robot and to the other robots

on the ground. The goal of adding the rotating motion was to make the Curie robot able to interact with its surroundings during the robot theater performance.

5. The third mechanical task was to design an arm that could perform all motions from the play Quantum Debate. The 3D models used were from the open-source robot InMoov that was designed by Gael Langevin (2012). Using Mr. Langevin's design adds five DOFs in each arm, sixteen DOFs in each hand and two DOFs in the neck. Therefore, the Curie robot has up to fifty-six DOFs, which makes it the most complicated robot in the lab. However, the Curie robot is missing its right arm, which means that it has thirty-five DOFs assembled and ready to use. Simple arm motions such as waving were programmed. An algorithm was developed that allows the Curie robot's arm to balance a cup or a test tube. In addition, there will be no human help involved – the Curie robot should balance the cup by itself. The GA was used to achieve this goal. Furthermore, the main objective of building a new arm was to replace the old arm of the Curie robot with a more realistic arm that had all DOFs functioning properly. The old arm was not good enough for use in the theater, whereas the new arm is considered sufficient according to several motion tests.
6. The fourth mechanical task was adding a neck to the Curie robot that allows it to track things with its head. This was also achieved using 3D models from InMoov. Motions of the neck and head were programmed and are realistic. Neck motions add a lot of naturalness to an android robot. Comparing the motions of our Curie

robot's neck with the very unnatural motions of other humanoid robots, one can appreciate how important a neck is for android robots.

7. The first complex software task was the development of the **tracking behavior**.

The goal of the tracking behavior was to make the Curie robot able to track the incoming person with her head. By using the Kinect sensor to get a depth image, the Curie robot was able to recognize the position of the nearest person. The tracking algorithm then uses the vision data to make the robot look at the person's face. Furthermore, the same algorithm was used to achieve the goal of tracking labeled objects. Any object can be tracked with glyphs attached to them. Using the AForge.NET glyphs the Curie robot was able to recognize various objects and track them. The goal of the tracking behavior was to make the Curie robot able to move its arm to the position of the object and grasp it with its hand.

8. The second complex software task, **grasping**, was achieved using the IK to calculate the angles of the shoulder and the elbow to move the end-effector (hand) to the desired coordinates. The IK allows the robot to use the object location that is figured by the robot vision system and move the arm to where the object is to grasp it.

An important component of the entire integrated robot design is **evaluation**.

Evaluation of android robots is especially difficult because subjective criteria must be taken into account and very little if anything is published so far on evaluation of such robots.

After I finished controlling the robot vision and the arm, I started evaluating the robot tracking and grasping. To evaluate the tracking system, two experiments were created that compared the Curie robot's head tracking to a human's tracking. The first experiment showed that the Curie robot can move its neck faster to the right and to the left, but slower up and down. This deficiency took the robot more time than the human when the tracked person was moving to the furthest point on the left or the right. The second experiment shows that the robot gets better at longer distances. In fact, the robot is faster than the human since the longer moving distance of the tracked person makes the robot neck move faster. The Pololu Maestro Servo Controller allows the user to set the acceleration of the servo speed, which affects shorter movements.

The grasping movement was evaluated by analyzing the IK and the coordinates the robot can reach with its hand. Since the Curie robot was built to be an android robot, the best method for evaluating the arm movement was by comparing the range of motion of the used DOFs to the range of motion of the corresponding human DOFs. The robot shoulder can move within 100 degrees and 65 degrees for the elbow DOF. When I compared this to a human shoulder and elbow, I could see that the range of motion of the robot shoulder is 140 degrees less and the elbow is 85 degrees less than the normal human shoulder and elbow rotation. This shows that the design of the robot arm is still not as good as a human arm. Thus, not all motions of a human arm can be reproduced in the robot theater. However, it works perfectly for the Curie robot grasping tasks, as well as waving and greeting behaviors. In other words, the new arm is sufficient for the Portland Cyber Theater and the play Quantum Debate.

In order to evaluate if the robot is “*human-like*”, one can ask people what they feel and think when observing it. In my case, the robot was demonstrated and discussed with Dr. Perkowski and PSU students and my friends. In addition, I used the help of my first child, Nora, who was born when I started building the Curie robot. My question was: “How does a child that is raised with an android robot perceive this robot? Who is this robot for her? What does she think the robot is?”

For almost two years Nora saw me working a lot at home on humanoid robot parts and next on the robot itself. This gave her the privilege of being raised in a house that has an unusual member. By the time she started realizing the world around her, I constantly introduced her to the humanoid robot that I had been working on. Her first impression was normal when the robot had only a torso, arm, and no face. After I printed the robot face, she was afraid to enter the room where I keep the robot. Then she got used to it and start touching its hand, but I observed that she was afraid of the robot’s actions when she touches it. After I started to test the arm and the neck, she became more scared. It took her some time to get used to the robot. The loud sound of the servos made her feel that there was something different in this humanoid robot than a normal human being. I think that having less noisy motors and some sound around the robot would make her accept the robot sooner. Furthermore, when the robot was able to track her face, she was impressed and started pointing at the robot.

As a father, I knew she felt that the robot is alive, which means I have achieved my goal of building an android robot. Figure 8-1 shows my daughter shaking the robot's hand.

Figure 8-1 Nora shaking the robot's hand

Working on a big embedded system project makes me feel that I have achieved all goals of this thesis project, but also my life goal of designing a complex and realistic, integrated embedded system entirely from scratch.

8.1 Contribution

The following list shows the contributions in hardware design of the robot.

1. A Complete design and build of legs of the robot including kinematics and programing.
2. 3D printing of both arms based on InMoov models and assembly.
3. Creating the concepts of bending and rotating motions of the robot's body and design of the first variant of the body.

4. 3D printing of neck and upper body from InMoov models and integrating them with modified version of the Curie robot's body.
5. The design of the Monster torque device together with calculations that allow future students to use similar inexpensive servos when needed in their designs.
6. Basic control programming of all motions of body, neck, and arms to verify the correctness of the hardware design.

The following list shows the contributions in software design of the robot. All algorithms were written by me from scratch in C# language, and some libraries were used to make programming easier for some devices and boards. For instance, Using Arduino-Pololu library and MS Kinect library to control devices.

1. Developing IGA to make the drum playing and kicking motions of legs.
2. Developing GA with hardware feedback to balance the cup with the robot's arm.
3. Developing a tracking algorithm for the robot's neck to track human faces and labeled objects.
4. Calculating the IK for the robot's arm to grasp an object with the robot's right hand.

Bibliography

- 3D printer and 3D printing news. (2011). Retrieved from Poppy: Open source 3D printed humanoid robots: <http://www.3ders.org/articles/20131021-poppy-open-source-3d-printed-hackable-humanoid-robots.html>
- Abdullah, H. A., Tarry, C., Datta, R., Mittal, G. S., & Abderrahim, M. (2007). Dynamic biomechanical model for assessing and monitoring robot-assisted upper-limb therapy. *The Journal of Rehabilitation Research and Development JRRD*, 43.
- American Academy of Orthopedic Surgeons. (2011). Retrieved from <http://www.aaos.org/Default.aspx?ssopc=1>
- BostonDynamics. (1992). Retrieved from Atlas - The Agile Anthropomorphic Robot: http://www.bostondynamics.com/robot_Atlas.html
- Electronic Products. (2016). Retrieved from InMoov: The 3D Printed Robot: http://www.electronicproducts.com/Videos/InMoov_The_3D_Printed_Robot.aspx?terms=printable
- Engineered Arts Ltd. (2004, Qctoer). Retrieved from RoboThespian: <https://www.engineeredarts.co.uk/robothespian/>
- Garstka, J., & Peters, G. (2011). View-dependent 3D Projection using Depth-Image-based Head Tracking. *8th IEEE International Workshop on Projector-camera Systems*, 52-59. Retrieved from <http://www.cs.ubc.ca/labs/imager/PROCAMS2011/0008.pdf>
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison Wesley.
- Jasim, W. M. (2011). Solution of Inverse Kinematics for SCARA Manipulator Using Adaptive Neuro-Fuzzy Network. *IJSC International Journal on Soft Computing*, 59-66. doi:10.5121/ijsc.2011.2406
- Kajita, S., Hirukawa, H., Harada, K., & Yokoi, K. (2005). *Introduction to Humanoid Robotics*. Ohmsha.
- Kirillov, A. (2006). *AForge.NET Framework*. Retrieved from <http://www.aforgenet.com/>
- Kondori, F. A., Yousefi, S., Li, H., Sonning, S., & Sonning, S. (2011). 3D head pose estimation using the Kinect. *International Conference on Wireless Communications and Signal Processing (WCSP)*. doi:10.1109/wcsp.2011.6096866
- Langevin, G. (2012, January). *InMoov Open Source 3D Printed Life-Size Robot*. Retrieved from <http://inmoov.fr/>
- Lufinka, O. (2015). Design of the new advanced modes for the mobile platform control application. *23rd Telecommunications Forum Telfor (TELFOR)*, 677-680.
- Morasso, P., Casadio, M., Mohan, V., Rea, F., & Zenzeri, J. (2015). Revisiting the Body-Schema Concept in the Context of Whole-Body Postural-Focal Dynamics. *Frontiers in Human Neuroscience*. doi:10.3389/fnhum.2015.00083

- Nave, R. (2009). Retrieved from HyperPhysics: <http://hyperphysics.phy-astr.gsu.edu/hbase/torq2.html>
- Neppalli, S., Csencsits, M. A., Jones, B. A., & Walker, I. (2008). A geometrical approach to inverse kinematics for continuum manipulators. *IEEE/RSJ International Conference on Intelligent Robots and Systems*. doi:10.1109/iros.2008.4651125
- Norton, R. L. (1992). *Design of machinery: An introduction to the synthesis and analysis of mechanisms and machines*. McGraw-Hill.
- Perkowski, M., & Dhawan, S. (2014, December 30). *Quantum Consciousness*. The Portland Cyber Theatre, Portland, Oregon, USA.
- Perkowski, M., Bhutada, A., Lukac, M., & Sunardi, M. (2013). On Synthesis and Verification from Event Diagrams in a Robot Theatre Application. *IEEE 43rd International Symposium on Multiple-Valued Logic*.
- Perkowski, M., Sunardi, M., Riedl, K., Huffman, C., & Goetz, A. (2013). Artistic Robots through Interactive Genetic Algorithm with ELO rating system. *22nd International Workshop on Post-Binary ULSI Systems*.
- Poppe, R. (2007). Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 4-18. doi:10.1016/j.cviu.2006.10.016
- Raheja, J. L., Chaudhary, A., & Singal, K. (2011). Tracking of Fingertips and Centers of Palm Using KINECT. *Third International Conference on Computational Intelligence, Modelling & Simulation*. doi:10.1109/cimsim.2011.51
- Redlarski, G., Ambroziak, D., & Palkowski, A. (2013). Optical Glyphs Based Localization and Identification System. 266-271.
- Rehiara, A. B. (2011). Kinematics of AdeptThree Robot Arm. In *Robot Arms* (p. 262). InTech. doi:10.5772/17732
- Sunardi, M. I. (2010). *Expressive Motion Synthesis for Robot Actors in Robot Theatre*. Portland: Portland State University. Retrieved from http://pdxscholar.library.pdx.edu/cgi/viewcontent.cgi?article=1719&context=open_access_etds
- Thomas, B. (1994). *ServoCity*. Retrieved from <https://www.servocity.com/>
- Willow Garage. (2008). Retrieved from PR2: <https://www.willowgarage.com/pages/pr2/overview>
- Wyeth, G., & Milford, M. (2011). *Forward and Inverse Kinematics*. Retrieved from <https://www.qut.edu.au/>
- Zarins, U. (2011). *Anatomy Next for Arts Education*. Retrieved from https://www.anatomynext.com/about_us.php
- Zeng, W. (2012). Microsoft Kinect Sensor and Its Effect. *IEEE Multimedia*, 4-10. doi:10.1109/mmul.2012.24

Appendix A: The Power Supply

Powering the whole system takes a lot of current since most of the servos used are large (Hitec HS-805BB). According to the HS-805BB servo data sheet, each servo needs 800 mA at their initial movement. Therefore, I found that using a computer power supply with a high output current would be a good power supply to run the servos. The power supply that I decided to use is the Logisys PS480D-BK 480 Watt ATX Power Supply. The reason behind using this power supply is that it has +5v, -5v, +3.3v, +12v, and -12v. The red wire coming out of the power supply, which is the +5v output, can be used to run all the boards and the servos. Its output current is 36Amp which means that this power supply can run up to 45 big servos like the HS-805BB. The Curie robot has 10 servos in each arm, two servos in the neck, two servos in the body, and six servos in each leg. Therefore, it has 36 servos and if we consider the bad scenario where we only use large servos, we do not have to change the power supply.

The orange wire coming from the power supply is the 3.3v output. That can be used if there is a need to run some boards with 3.3v like a mini Pololu Maestro Servo Controller. This board can run up to six servos and since it is small, it can be used to control the hand's servos.

The yellow +12v can be useful for future improvement of the robot. It can be used to run the motors that need this high DC voltage. For example, if the robot needs to be moved, four 12v motors can be used to move the wheels that move the whole robot around.

This power supply has safety features like short-circuit protection. This is a very important feature since the robot moves a lot, which may make wires come loose. A loose wire may cause a short circuit, which is not safe. Finally, in order for this power supply to run, it needs to have its green wire connected to the ground.

Appendix B: Pololu Maestro Servo Controller

The Pololu Maestro Servo Controller is a board that has a number of channels used to control servos. It is designed mainly to move the servo. However, it has more features that make working with robotics easier than using an Arduino board. One of those features is the graphical user interface (GUI) application that comes with it. It simplifies controlling the servos and allows the user to simply plug the USB cable and control the servos without having to write any code. This is an important feature since it quickly tests if the servo can control the robot parts. It also allows the user to set the speed and the acceleration of the servo. These options can give a smooth movement of the servo. Furthermore, the controller has a separate power adapter for the servos which gives the user the ability to raise the servos voltage. The most important feature is the controller allows for universal asynchronous receiver/transmitter (UART) communication with another board like an Arduino. This allows me to write the main program in C language and upload it to the Arduino and the Arduino uses the UART communication to send commands to the Maestro Servo Controller. Figure B-1 shows the mini Maestro 24-Channel USB Servo Controller.

Figure B-1 The mini Maestro 24-Channel USB Servo Controller

The Pololu Maestro Servo Controller has a Maestro-Arduino library that makes the code for controlling the Pololu by the Arduino much simpler. It is uploaded to the GitHub (<https://github.com/pololu/maestro-arduino>) where you can find all the information needed to write a C code for the Arduino that allows for controlling the servos.

Appendix C: Hardware Controlling Program

The hardware controlling program is an Arduino code written in C language whose main task is to control the servos. Overall, it gets the directions of the robot face movements from the main application and converts them to a series of commands that are sent to the servo controller. The program starts with establishing the serial communication with both the Pololu Maestro Servo Controller and the main application (the C# application). Then it waits until there is a robot's face or arm movement command coming from the main application.

The face movement commands are different from the arm movement commands. The main application sends either center, left, right, up, or down commands to the Arduino. Then the Arduino increases or decreases the servo position based on the saved value. The initial command is always to center the face, or, in other words, make the face move to the center position in the vertical and the horizontal servos. On the other hand, the main application sends other commands to move the arm. It starts with sending an arm movement command that make the Arduino prepare for the hand coordinates. Then the main program sends the x-axis and the y-axis coordinates of the hand. After that the Arduino calculates the IK for the arm joints and saves those angles. A look-up table for both joints is saved in the Arduino to convert the joint angle into a servo value. Finally, if the calculated joint angle is within the limits, the Arduino sends the servos values to the servo controller to move the servos to the desired position. Otherwise, the arm will not move which means that the desired coordinate is unreachable.

The last command that the main application can send to the Arduino is the waving command. The Arduino move three servos to do the hand waving motion for almost 15 seconds.

Appendix D: Main Application

The main application is an application that is written in C# language. The application starts by opening the port with the Arduino to allow serial communication with it. In addition, it initializes the Kinect sensor that provides the depth image. After these steps are done, the application sends a command to the Arduino to set the face to the center and show the image from the Kinect. All the hardware status is shown in the right bottom corner of the application. Figure D-1 shows two different hardware statuses that can be shown in the application.

Figure D-1 The hardware status in the application

After all the set-up actions have been completed, the application starts tracking the human face by default. It shows what it sees on its screen and surrounds the tracked person with a red rectangle and the hands with circles of different colors. The right hand is circled with a purple circle and the left hand with a blue circle. There is a larger rectangle where the robot's sight of view is placed in the middle of the camera viewer. Whatever we see in that rectangle is what the robot is directly looking at. The coordinates of the head are shown under the camera viewer. Figure D-2 shows a snapshot of the application where a person stands in the middle of the camera viewer and the coordinates of the head are shown at the bottom.

Skeleton tracking

Head Position

X: 323 Y: 107

Figure D-2 A snapshot of the application with a person detected and the head coordinates shown in the bottom

The other tracking mode is tracking the AForge glyph. It can use the same tracking algorithm used to track the human face, tracking one glyph at a time. However, the glyph is surrounded with a green rectangle and the name of the glyph is shown in the middle of the glyph. Also, the glyph coordinate is shown at the bottom of the camera viewer. The camera viewer is different than the first camera viewer that is used to track the person's face. In other words, there are two camera viewers, one for tracking people's faces and another for tracking the AForge glyph. They both have a red rectangle that informs the user what the robot is looking at. Figure D-3 shows a snapshot of the application where the glyph is in the middle of the camera viewer and the coordinates of the glyph are shown at the bottom.

Figure D-3 A snapshot of the application where the glyph is in the middle of the camera viewer and the coordinates of the glyph are shown at the bottom

The application shows the user what are the commands that are sent to the Arduino. These commands control the robot's face movements. If the application is not sending any commands, it shows that the face is centered. Figure D-4 shows a snapshot of the application with different commands.

Head direction of movement (horizontally):	Head direction of movement (vertically):
center	center
Head direction of movement (horizontally):	Head direction of movement (vertically):
right	down
Head direction of movement (horizontally):	Head direction of movement (vertically):
left	up

Figure D-4 A snapshot of the application commands that are sent to the Arduino

There are three tracking modes that the user of the application can switch between using buttons. When the user wants to track a glyph, he or she should write the glyph

name in the textbox, and then click the “track the glyph” button. The other mode is the face tracking mode which directs the robot to track the detected person’s face. The last mode is when the robot is not tracking anything.

The robot arm can be moved by typing the XY coordinates of the hand. Then the robot tries to reach these coordinates. If the coordinates are unreachable, the arm will not move. The other behavior of the arm is hand waving. If the user clicks the wave button, the robot starts waving for almost 15 seconds. The user is not able to send any other arm command until the waving is over. Figure D-5 shows the buttons that control the arm.

Figure D-5 Buttons that control the arm

There is a button that is configured specifically for the tracking algorithm experimentation. If the button is clicked, the application waits until person’s face is out of the robot’s sight of view (the red rectangle). Then the application starts counting the time that the robot takes to find the human face again in milliseconds.

Appendix E: The System Design

The system controls the humanoid robot motions based on the vision information.

This means that the system has to have a camera that allows the humanoid robot to see the world. In addition, there some motion control system must exist that will react to the environment received by the camera in order to demonstrate human-like behaviors. The camera used in my thesis is the Kinect sensor. It provides the depth image that is used to find the person standing in front of the robot. Furthermore, the motors used in the system are two kinds of servos, large servos (Hitec HS-805BB) and small servos (Hitec HS-311).

The input from the camera is taken by the computer and analyzed to find the person's face or an AForge glyph. Then if the tracked person's face or the object is not in the robot's sight, the computer sends a series of commands to the Arduino to move the robot head. Those commands are translated to servos values that make the robot look at the person's head or the object again. The Arduino calculates the values and sends them to the Pololu Maestro Servo Controller in order to move the servos. Figure E-1 shows a picture of the whole system.

Figure E-1 A general view of the system

Appendix F: The Supplemental Files

Some important files are attached as supplemental files. The first one is the main application, which is an executable file called Vision-Based Motion for a Humanoid Robot. The second file has the code for the main application and is called MainApplication. It is written in C# language that needs a compiler to generate the executable file. Visual Studio 2015 was used to compile the code.

The other types of supplement files are Inno setup script files (.ino) that are written in C language. They can be run using the Arduino Software. In addition, an Arduino Uno is needed to run the program. There are three files of this type and they are called Controlling_servo, Leg_IGA_V2 and Hand_GA_Balancing_Cup.

All of these files were running perfectly on windows 10 Pro (64-bit Operating System). The used computer was running on Intel i7-5820K CPU and 16 GB RAM.