Elderly Assist Robot

A Major Qualifying Project Report:
Submitted to the faculty of the
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the
Degree of Bachelor of Science By:

Udit Adhikari (ECE)
Md Monirul Islam (ECE)

Jonathan Noser (RBE)
Karissa Puliafico (RBE)

In partnership with
Shanghai University
Partners: Hua Xi, Jiang Ping, Zhang Mengzhe, Liu Fengjin,
Song Zhuoyuan, Wang Xiang, Lu Chiqin, Cao Yue, Chen Lei

Date: October 19, 2010

Approved:

Professor Yiming (Kevin) Rong, Major Advisor, Mechanical Engineering
Professor Wenjing Lou, Co Advisor, Electrical and Computer Engineering
Professor Guo Shuai, Co Advisor, SHU

Abstract

This project aimed to create a robot capable of assisting elderly people with tasks in their everyday lives. The project focused on the design, simulation, and the implementation of a mobile robotic base with an attached robotic arm. The project culminated in a prototype robot capable of performing basic chassis and arm control which can be used as a platform for future development.

Acknowledgements

The group would like to thank the following individuals for assisting us throughout this project:

Sponsor: Shanghai University

Advisors: Yiming Rong
Wenjing Lou
Guo Shuai

We would also like to thank all of the other professors from Shanghai University and the representatives from the Beckhoff Company who assisted us with this project.

Table of Contents

Table of Figures

Authorship

Udit Adhikari wrote the introduction. He also worked on the background sections – “Required Capabilities of Our Fully Developed Robot”, user location, tracking, and object avoidance. In the methodology section he wrote the following – “Arm Simulation” under “Arm Control” and “protecting function” under the “Path Following for Autonomous Navigation". At last, he wrote the “Results and findings” section, “Conclusion”, and “Future Works”. Jonathan Noser wrote the Visual Servoing in the background section and Path Planning in the methodology section. Jonathan worked on base hardware installation and testing as well as SLAM, trajectory generation and a user location beacon system which was not implemented due to time constraints. Karissa Puliafico wrote the SLAM and Base Testing Methodology sections of the report. She was also responsible for editing and proofreading most of the report. Monir Islam wrote the “business side plus how much will the parts cost?” part of the background section. He also wrote the “Arm Control Implementation” and “Arm Programming” under the “Arm Control” part of the methodology section.

Introduction

Progress in the field of robotics has always headed towards more precise automation. Robots have come a long way from primitive 1961 "hydraulic robots produced by Joe Engleberger's first plant in Danbury, Conn…to today's [robots] handling of semiconductor wafers under conditions that are so demanding that they can't be processed any other way" (Teresko, John 1996). The application of automated robotics has also spread into other areas other than manufacturing. Recently, there has been a great push towards making robots that can assist people in their daily lives. This has greatly improved the patient care system and made the process of patient care less human based and resource intensive. In our project, we seek to use the current technologies of automation to make a robot that can assist the elderly and the disabled in their daily tasks.

The objective of this project was to make a prototype that could demonstrate proper functioning of two features that we thought were the necessary foundation for the operation of a life assist robot. These were as follows – implementing an arm that can move to a location specified by a user and implementing a basic level of autonomous navigation capabilities for the robotic base. To accomplish these objectives, we broke them up into subtasks. The tasks were then divided between four groups. The first group was responsible for the mechanical design work. They started with a conceptual design and were able to produce a specific design that met all of the design requirements. The second group was responsible for simulating the arm and the base of the robot motion with MATLAB. This was important because we wanted to be sure that the mechanics of arm movement and the autonomous navigation worked in the simulation environment before we implemented them in a physical environment. The third group was assigned the task of performing the electrical installation of the arm and testing the arm movement and control. The fourth group had to implement autonomous navigation and obstacle avoidance algorithms on the onboard computer used for controlling the mobile base.

Background Research

General Problems Facing the Elderly/Disabled

The average human body is inherently somewhat fragile, and this fragility usually increases rapidly as a person ages. Because of pre-existing and new conditions that develop over time, many tasks needed to sustain daily life become very difficult for some elderly people. Many medical sources refer to such everyday tasks as "Activities of Daily Living" (ADLs). One medical website, MedicineNet.com, defines ADLs as: "The things we normally do in daily living including any daily activity we perform for self-care (such as feeding ourselves, bathing, dressing, grooming), work, homemaking, and leisure" (MedicineNet, 1998). Some of these tasks become difficult to perform because of physical ailments that can occur such as weak bones and muscles, or mental conditions such as Alzheimer's that can worsen over time. In addition to the physical struggles elderly people may face in everyday life, they can also sometimes experience social deprivation as well. For the elderly and disabled, communication with loved ones and caretakers can sometimes be extremely limited either through being unable to move and/or being unable to speak. Due to these physical constraints, communication with caretakers may also be negatively impacted, possibly leaving the elderly individual lonely or even depressed.

Problems Facing the Elderly/Disabled in China

With a population of over 1.3 billion people and with that number increasing every day, China is easily the most populous nation in the world. According to the CIA World Factbook, about 8% of China's citizens are 65 years or older (Central Intelligence Agency, 2010).This may seem like a relatively small percentage of elderly people, but considering China's total population, this means that well over 150 million Chinese are elderly and once again, that number is increasing every day. Zheng Silin, the deputy director of the Subcommittee of Population, Resources and Environment of the Chinese People's Political Consultative Conference (CPPCC) National Committee stated back in 2007 that: China is "already the only country with an aging population of more than 100 million, and their number is growing even faster" (Xie Chuanjiao, 2007).

As has been customary in China for centuries, elderly people are revered, greatly respected, and often taken care of by their children. Traditionally, because of their great respect for their parents, Chinese adults would remain at home especially to aid their aging mothers and fathers. However, in today's society, the adult children are more motivated to leave home and pursue other interests, sometimes in other parts of the world. Because of the growing number of young adults leaving the family home, some elderly people are left alone with few people or no one to take care of them. As the population of China ages, it faces many of the same physical and mental health issues and disabilities as those in the rest of the world. Due to this and the fact that more and more Chinese sons and daughters are leaving home, it is becoming critical that an alternative to human care for the elderly in China be developed.

Needs That a Robot Can Assist an Elderly Individual With

Movement is one of the earliest capabilities and ADL that may become extremely limited as a person ages. The inability to move could manifest in the form of pain, such as in the case of Arthritis, be a simple loss of muscle control in a particular area, or could be the result of Osteoporosis or any other degenerative disease or illness, in addition to other forms as the body ages. Because these conditions may develop and limit a person's range of motion and ability to walk and pick up objects, it is important that a robot designed to assist the elderly be able to be used as a device to aid in stability and be able to pick up objects and bring them to the user. These functions will be incorporated into the mechanical design and appropriate sensors will be used to aid in the mechanical operation of the robot.

In addition to physical issues that may develop in the elderly over time, mental and emotional conditions may also worsen or develop. One of the biggest emotional challenges the elderly face, especially in China, is the feeling of loneliness. According to a survey taken by Durham University and the University of Reading of people of 60 in China in 1992 and 2000, "The percentage of older people who said they were lonely has doubled from about 16% in 1992 to 30% in 2000" (RedOrbit, 2009). In order to help combat some of the loneliness that the elderly experience, the planned robot will be able to listen to a human speaker and respond verbally through the use of microphones and speech recognition technology. Giving the robot the capability to interact with a human through speech and the ability to execute verbal commands will hopefully allow the elderly user to feel more connected to the robot and help to ease feelings of loneliness.

Along with the feelings of loneliness that sometimes plague elderly people often comes a sense of being ignored. If elderly people have adult children and are put into or reside in a nursing home or assisted care facility, they sometimes feel ignored by their loved ones because the now grown-up children have their own families, jobs, and other life commitments. Because many adults do not have much time to devote to their aging parents, it is only natural that an elderly person may feel like no one cares for them anymore. This robot will act as a caretaker and friend to the elderly person so that they do not feel ignored and hopefully make them feel happier through the interaction.

Required Capabilities of Our Fully Developed Robot

In this project, we will be using a robot that has to come to the aid of the user. This will have to be done in an autonomous mode because the user is an elderly or disabled person. Putting a human control on the navigation part will diminish the benefit of having an assist robot. A comparison to this would be the article, a study of an autonomous mobile robot for a sewer inspection system. The article describes a robot that was designed to navigate through sewers. Although we won't be designing a similar robot, the logic used to design the robot to meet the certain criteria does serve as valuable information.

The sewer robot had to be designed to carry out operations for a long period of time. This meant the robot should have low power consumption. In order to improve low power consumption capability, they made the navigation wheel based rather than legged. In our case, the robot has to assist a disabled or elderly person. The person will need aid at uncalled times, which would require the robot to be functioning at all times. Thus, low power consumption during its operation becomes a priority. This means we would also be using wheel based navigation for the robot.

In the sewer robot, they designed the sensors so that it was a good trade-off "between redundancy of sensors which need to be robust and the computational cost of processing them" (Ahrary, Nassiraei, and Ishikawa). In our robot, the computational part will play a huge role because most of features such as user location, tracking, and object avoidance require a lot of processing. Thus, we will have to give more weight to the computational cost than the redundancy of sensors.

The navigation part of the sewer robot was also tackled with ingenuity. The data from GPS was not available for the robot to use in the tunnels, so the robot had to use local means to calculate the position of the robot and navigate. The robot was planted with sensors that would help it calculate the distance from local features to the robot. Once this was able to work, making the robot navigate was just a matter of using mathematical formulas.

The case study provided the general thinking process that can go into making a robot for a specific task. The following sections dig deeper into the features of the robots that are required to make our elderly or disabled life assist robot navigate.

Our fully developed robot will have to be able to perform the following three tasks to have autonomous navigation – user location (get general direction), tracking, and object avoidance.

User Location

Our robot has to be able to travel to the user when it is called for help. For this to happen, the robot has to have some sort of ability to know the position of the user or at least get a general direction of where the user is located. The robot will be performing indoor user location. This decreases the accuracy of common location technologies such as GPS, “GPS alone cannot provide sufficient data for determining the user’s location when the user is inside of a building” (Yim, Jaegeol). One of the indoor tracking methods currently popular is a WLAN-based tracking system. This type of tracking system “determines a user’s position referring to the received signal strengths (RSS) of the signals from access points (APs)” (Yim, Jaegol). The location of the user is accomplished by trilateration, which requires three wireless base stations. The coordinates of the base station and the distance of the user from the base station are known to the robot. Using these two pieces of information we can calculate the co-ordinate of the user. The same principle can also be used to calculate the co-ordinate of the robot. Once, the position of the user and the robot is known, it’s a simple matter of mathematical calculation to figure out the path of the robot.

The article Algorithms for acoustic localization based on microphone array in service robotics by Enzo Mumolo, Massimilano Nolisch, and Gianni Vercilli describes a robot and the algorithms that go into making this robot which can perform user localization based on speech, and also carry out the task of autonomous navigation and object avoidance till it reaches the user. The method by which the user location is performed is as follows:

This approach computes the acoustic source position by triangulation of some estimations of the direction of arrival of the acoustic wave. Such estimations are obtained from the phase differences between couples of microphones of the array. Since our service robot moves in real indoor environments, with an high degree of noise, one objective of this study is to work-out a practical algorithm (from a real-time point of view) with high performance at low SNR; for this reason a particular attention in the algorithm development was given to guarantee high noise robustness(Mumolo Nolisch, and Vercilli).

Tracking

To be able to perform tracking, the robot will have to have sensors that can carry out dead reckoning. Dead reckoning "is a simple mathematical procedure for determining the present location of a vehicle by advancing some previous position through known course and velocity information over a given length of time". Most "land-based mobile robots rely on dead reckoning to form the backbone of their navigation strategy" (http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/jmd/). Dead reckoning can be done using a lot of different types of sensors. According to Dixon, and Henlich, one of the best ways to perform dead reckoning is to use an active beacon system. The beacon systems are "more suitable for autonomous mobile robot control, as the position information is calculated at the mobile end" (http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/jmd/). With beacon systems, there are transmitters that are synchronized and they transmit a continuous wave. The receiver measures the time difference for the signals to arrive and calculates the position. One of the other ways to tack the robot was to use odometric sensors. These encoders are usually coupled with the wheels and provide angular position and velocity for the navigating robot. Some of the common encoders are brush encoder, potentiometer, optical encoder, magnetic encoder, inductive encoder, and capacitive encoder.

Object Avoidance

Object avoidance is one of the major requirements for the robot to be able to navigate properly through the room. This is usually done by using technologies such as radar, or infra red transmitter and receivers that can detect an obstacle. Then the data is processed in the micro-processor to calculate distance and perform obstacle avoidance techniques. The article "Visual Sonar: Fast Obstacle Avoidance Using Monocular Vision" by Scott Lenser and Manuela Veloso describes a robot that uses vision to carry out object avoidance. According to them, "General obstacle avoidance can be effectively accomplished with several robot sensors including infrared, sonar, or laser. However this requires that a robot be equipped with such sensors. Instead, we are focused on robots that have vision as their main sensor" (Lenser and Veloso). They have coined a special name for the type of sensing that they use for their robot – "visual sonar". The key items in the visual sonar are "the vision algorithms to detect obstacles, partial identification of obstacles, and fast algorithms to update a radial map of the environment" (Lenser and Veloso).

Their vision based system routines integrates domain specific identification of detected obstacles. Using this technique they can selectively choose to avoid objects or ignore them if necessary. For example, they can choose to ignore the floor as being an object to avoid but avoid the walls as an obstacle. Currently their vision can identify the following object types: wall, stripe, unknown_obstacle, red_robot, blue_robot, ball, cyan_goal, and yellow_goal. The way the objects are identified is that every object is considered to have a certain color and pixel length. They calculate the distances to the object "by intersecting rays through the closest pixels of the object on the image onto the ground plane the robot is standing on" (Lenser and Veloso). In this manner, the robot is able to recognize the obstacle and avoid it if desired.

Visual Servoing

In order for the robot to find and pick up objects it must be able to identify and locate them. The translation of visual sensory input to actuation is called visual servoing. This process is often applied to robotic arms and has been used in elderly assistance robots such as the KARES II (D.M. Wilkes, 1999). Visual servoing makes robotics powerful because it allows them to work in environments which are not predetermined. Depending on the robustness of the system the objects can be varied in shape, orientation, placement and nature. Visual servoing however proves to be a difficult task as varying illumination, motion, and processing time present challenges (Asakura, 1999). The process of Visual servoing begins with processing the image.

A simple method for identifying objects is blob detection. One method for blob detection is using single color objects which are distinctly colored from the environment. In this method the initial is image is modified to produce higher contrast for specific color ranges. Using a predetermined threshold the desired object is then distinguished from the environment through blob detection (Ponsler, 2010). Another method for object is fiduciary markers or fiducials according to RoboRealm "Fiducials are commonly known as "markers that are easy to identify" and are typically added to an environment where localization and navigation are needed for robots using machine vision" (RoboRealm, 2010). Fiducials are flat high contrast images which can be placed on objects or in specific locations. These markers can be used to identify the location and orientation of an object relative to the robot (RoboRealm, 2010). These image processing methods can be handled in code or by a variety of programs such as MATLAB, Open CV or RoboRealm. Once a robot has identified and located an object it must be able to navigate to it.

For a robot to recognize an object it must have prior knowledge (have an algorithm to identify the shape, color, etc.). The methods employed can include model-based, blob recognition, and fiduciary markers (Asakura, 1999). Once an object is recognized if its size in already known its position and orientation can be determined. With a stereo camera its position can even be determined in three dimensions. In order to do this a camera must be calibrated so pixels can be accurately converted to measurements (such as mm, inches etc). However this method is not always sufficient when dealing with environments sensitive to error. A robotic arm could cause damage to the object it is trying to pick up (Heping Chen, 2009). When an arm is used in elderly assistance, such as with the KARES II, "compliance increases the level of safety for unexpected collision…" (D.M. Wilkes, 1999). In order avoid damage and reduce error force control (such as a force sensor on the manipulator arm) can be used to increase accuracy and control force applied. When properly applied, force feedback can provide information about the z-axis; possibly reducing the need for two cameras (Heping Chen, 2009). Using a combination of image processing and force feedback a robot can safely pick up an object.

Business Side/How Much Will the Parts Cost?

There are three kinds of robot in the market. They are - industrial, personal and service robot and our robot falls under the personal robot category. It is very important that we keep the price of the robot low, both for business purposes and also to make it affordable to people of all economic status. The target market for the robot that we are going to built is elderly people. Most of the time the elderly people don't have a job or they are retired. So anything they would buy would have to be cheap. It is of outmost important that we estimate a cost for building such a robot. Although the Ugobe's Pleo robot had a great demand when it was launched but due to the high price tag of the robot the robot could not stay in market for long and so the manufacturer had huge losses. It doesn't matter how good the functionalities of the robots are, the price of the robot is very important for the market of that product.

At this point we have figured out the minimum equipment required to build any robot and found the typical cost of this parts with different specification. The parts are – navigation system with ultrasound or infrared, DC motors, encoder, Visual Serving with binocular – Camera, filter, cable driven arm.

The outline of their cost with a brief description of the features are given below -
Navigation with ultrasound –
DFRobot URM05 High Power Ultrasonic Range Finder
Product code: RB-Dfr-41 --- $ 100
Description - Range: up to 10m
Direction angle 15° (-6dB)
Operating temperature range: 0°C ~ + 40°C
Operating voltage: 6V-24V
Specification -http://www.robotshop.com/dfrobot-urm05-high-power-ultrasonic-range-finder.html
Phidgets USB Rotary Encoder
Product code: RB-Phi-08 --- $ 27.60

Smart Sensors and Applications V1.0
Product code: RB-Plx-100 ------------------------------ ----------------- -----$ 29.99
Specification - http://www.robotshop.com/content/PDF/smart-sensors-and-applications-text-122-28029.pdf

Ultrasonic Sensor
U4466 (ultrasonic sensor) -- $51
Specification – Frequency – 143 kHz
Range - 4” to 79”
Operating Temperature – -40 to 140 F
DC Motors –
Banebots in Brushed DC Motor
Product code: RB-Ban-68 --- $ 5.25
Description - Mabuchi style RS-540 motor
- 4.5 to 12V operating voltage
- 16800 no load rpm
- 5-pole balanced armature
- Supplier code: M2-RS540-120
Specification - http://www.robotshop.com/PDF/rbban68-banebots-DC-motor-specs.pdf
Banebots FF-050SK-1490 6V 15025 RPM 0.93 oz-in Brushed DC Motor
Product code: RB-Ban-65 --- $ 3.43
Specification - http://www.robotshop.com/PDF/rbban65-banebots-DC-motor-specs.pdf

Digi-Key Part Number 1022017-ND
Manufacturer CUI Inc (M487X0003) --------------------------------------- $ 121.30
Type – DC with Capacitive Encoder
RPM – 11000RPM
Shaft Rotational Direction – CW

Encoder (Programmable) –
DFRobot Wheel Encoders for DFRobot 3PA and 4WD Rovers (2pk)
Product code: RB-Cyt-39 --- $ 16.35

Digi –Key part number 102-1307-ND
Manufacturer CUI Inc (AMT102-V) -- $ 29.95
Encoder Type – Capacitive
Encoder (Absolute Octal) – Digi-Key part number GH3075-ND
Manufacturer GrayHill Inc (26ASD45-01-1-AJS) ------------------------------ $ 15.31
Encoder Type –Mechanical
Output type – Octal (Absolute)
Pulse per revolution – 8

Visual Serving with binocular – Camera
SFE CMOS Camera - 640x480 pixels
Product code : RB-Spa-116 -- $9.95
Digi-Key Part Number IN3003CAM2-ND
Manufacturer Trpp Lite (IN3003CAM2) -- $ 74.95
Resolution – 307.2k Pixels

Digi- Key Part Number MLX75403ACM-ND

Manufacturer Melexis INC (MLX75403ACM) -------------------------------- $ 361
Resolution – 640x480
Speed 30fps

Filter – Digi-Key Part Number NUF6410MNT1GOSTR ND ---------------------- $ 0.18750
Manufacturer ON Semiconductor (NUF6410MNT1GOS
Power – 0.6W
RC Network

The price of Erector Spykee - The Spy Robot is $ 369.99. This robot can be used as spy robot, VOIP phone, video surveillance, digital music player, sound and video recording. This robot has most of the features that we need in our robot and gives a sense of the cost of building our robot. The total costs of building this robot depend on how many of each of these parts we need to use for our robots and the specification of the robot. The minimum cost of the required parts is $200. An estimated total cost of the robot would be at least $400 US dollars.

Methodology

Chassis Control

Path Planning for Autonomous Navigation

Main idea

The program for path planning is based in MATLAB and the main idea of path planning is the use of the artificial potential field method. It mainly consists of force vectors, created by the obstacles or target, which may have repulsive or attractive characteristics. With the resultant of forces affecting the robot, the next point of the path can be defined. So in this way, the whole path will be generated, as seen in Figure 1.

The program allows users to put two obstacles on a map, the obstacles are represented below by the circular and rectangular shapes. Each obstacle has a unique influence region. When the robot enters the region, it will be affected by the repulsion. The radius of the influence region is adjustable and with the parameter entered, the robot will go further away when avoiding the obstacles. The dashed lines in Figure 1 below show the influence regions of the obstacles.

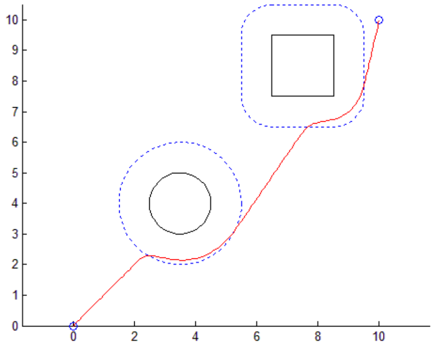


Figure 1: The path generated by the program

Oscillations

As mentioned above, the resultant of forces acting on the robot is calculated by summing up all the force vectors from the environmental effects. However, sometimes this can create problems. For example, when the attraction and repulsion vectors are in line and almost equivalent, the robot will go between two adjacent points repeatedly. This situation causes some oscillations in the motion of the robot. Sometimes the robot will fall into the oscillation region and will be unable to reach the destination. Because of the oscillations, the robot may deviate from the desired path and the longer the robot oscillates, the larger the deviation may be.

This problem can be handled by adding a piece of code which judges the direction that the base will follow. When the program determines that the robot is in oscillation, it will force the robot to deflect and go out of the oscillation region. In the process of getting rid of an oscillation region, some minor oscillations are unavoidable as shown in Figure 2. Because of this, after the whole path generated, the position of some oscillation points should be modified. For example, when there is oscillation in path (Fig.3), we could modify the position of point 2 to the middle of the line between point 1 and point 3. After the modification, the path becomes point 1 to point 4, and from point 4 to point 3. By using this optimization process multiple times, the path will be smoother as seen in Figure 4.

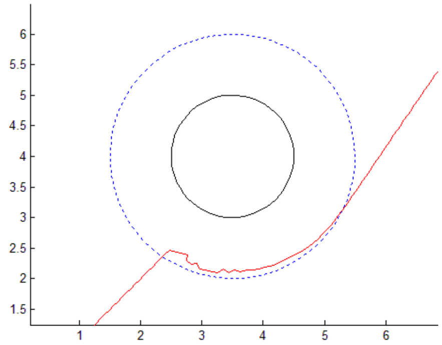


Figure 2: Oscillations in the path

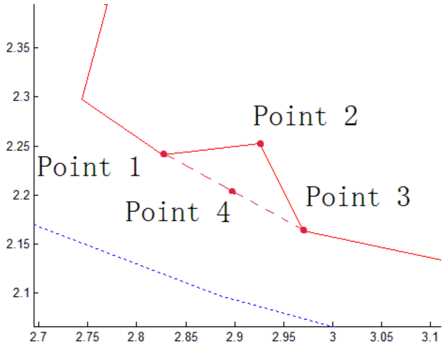


Figure 3: Solution of oscillation optimization

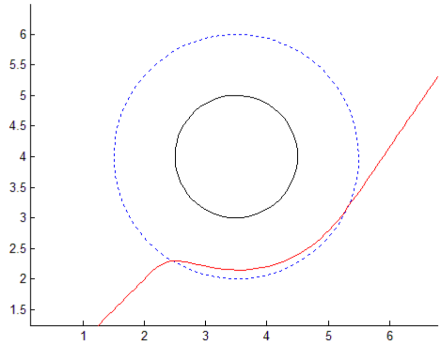
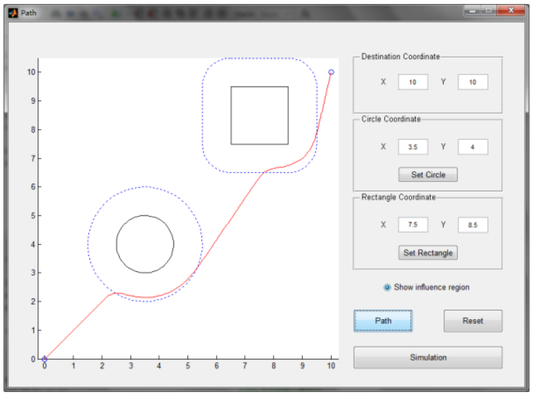


Figure 4: Optimized motion path

GUI interface

The GUI interface was created in MATLAB. Some basic elements in the interface such as axes and buttons are available. By writing the callback code, the input data can be transmitted to the program and the results are shown in the axes defined, as seen in Figure 5.

Through the GUI interface, it is convenient for users to input the parameters such as a destination coordinate, obstacle coordinates and so on. The program also allows users to choose whether to show the influence region of the obstacles. In order to connect the interface with simulation, a button was created to open the simulation.



Destination Coordinate
X 10 Y 10

Circle Coordinate
X 3.5 Y 4

Rectangle Coordinate
X 7.5 Y 8.5

● Show influence region

Figure 5: GUI interface for path planning

Simulation of the base

The model and simulation of the robot's base were created in Soildworks. Reassembling all the components causes some redundant degrees of freedom (DOFs) which can cause errors in Simulink. These were removed and the model was then imported into Simulink. Each component was shown as a block in Simulink. By adding the function blocks, the DOFs could be driven as desired.

As mentioned previously, the simulation needs to be connected with program, so we used the workspace as a temporary storage area. The calculated data was put in the matrix and sent to the workspace from which Simulink gets the data. In this way, each wheel on the base could get the speed and direction of rotation. Driven by the wheels, the robot could follow the given path.

As the robot is not a dot but a machine with radius, the influence region of each obstacle needed to be larger so that the robot could avoid the obstacle without colliding as seen in Figure 6.

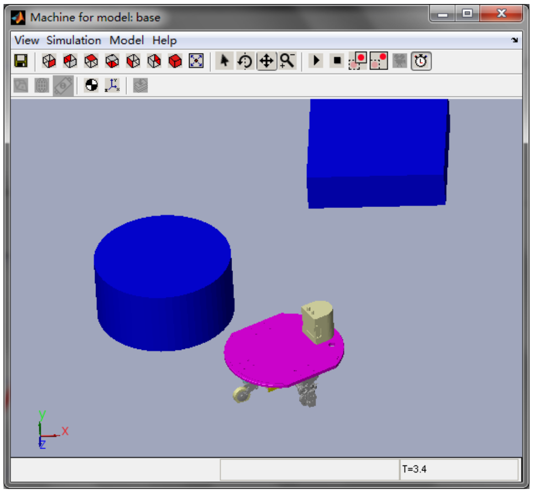


Figure 6: The base simulation

Path Following

For many applications in robotics, a robot cannot accept analog data as input. To move the robot we needed a mathematical ratio relating the speeds of the wheels. The path generated by the simulation group was an ideal path which is not easily represented by a mathematical function. Translating this analog path into digital motor signals is analogous to digitizing any analog signal, it needs to first be broken down into discrete steps. These discrete steps are states, which begin with where the robot is and end along the path generated by the simulation group. These states are defined by a two-dimensional coordinate point and an angle. The robot then needs a path between these two states. This can be achieved by using proportional feedback control; however, this method requires precise velocity control and is less robust. Another method is trajectory generation, in which it is possible to generate this path since two points and angles represent points on the function and the value of its derivatives at the two points. This trajectory could be as simple as a line or as complicated as a quintic polynomial depending on how different the angles are. For our application, the differences between the first and second angle is no larger than -90 and 90 degrees exclusive. Therefore, any transition between states can be represented by a quadratic polynomial.

To do these calculations, the coordinates must be shifted so that the parabola generated can represent a function. The origin is moved to the current location of the robot and the x-axis is moved so that it intercepts the two states. In this way the generated trajectory becomes a function as shown in Figure 7.

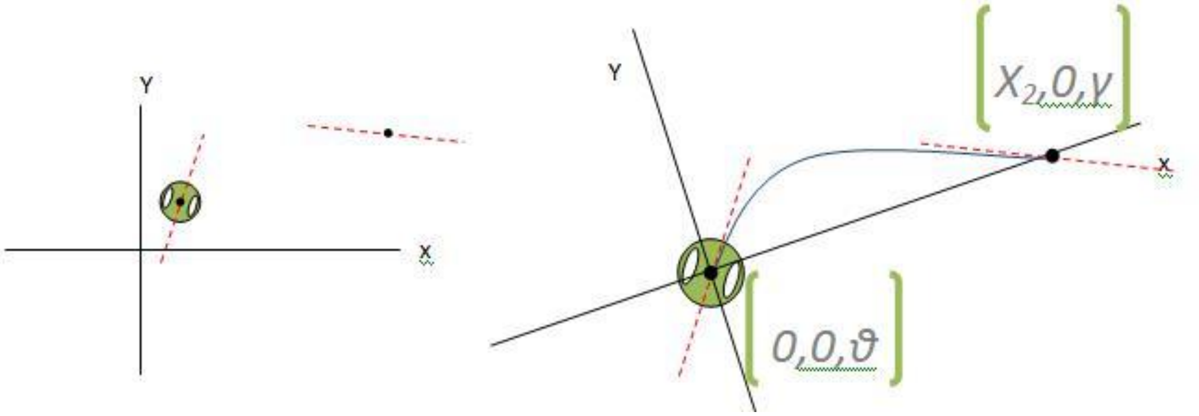


Figure 7: Parabolic trajectory equations

With the coordinates shifted we can generate the coordinates of the parabola. Theta represents the angle between the first point and the x-axis, and γ represents the angle between the second point and the x-axis.

For the quadratic equation $y = A * x^2 + B * x + C$

$B = Tan(\theta) \therefore A = \frac{Tan(\gamma) - Tan(\theta)}{2 * x_2}$ and since the x-intercept is set a zero C is zero.

To follow this generated path we used a combination of proportional, integral, and derivative (PID) maximum velocity and PID control of the relation of the wheel speeds. This control function's output is the sum of the error, its integral, and its derivative, each multiplied by a unique adjustable variable called gain. This system allows the robot to actively compensate for systemic errors such as drag from the caster wheels.

$$Output = P * error + I * \int error + D * \frac{dy}{dx} error$$

The error for PID velocity control is x_2-x_r (where x_r is the x coordinate of the robot). This is used to control the maximum speed of the robot.

The error to determine PID differential wheel speed is the difference between the y-coordinate of the robot and the y-coordinate of the parabola. This PID result is scaled and divided by the maximum speed of the inner wheel of the turn causing the robot to shift back towards the parabola. Together these PID functions drive the distance and angle error to zero.

This method is not without its limitations; however, since this method relies on encoders for accurate positioning, any errors in the encoder readings quickly build up over time limiting the accuracy of this method. In our robot, while the encoders are accurate to the nanometer, significant play in the motor itself and the gears in the drive train limit the accuracy. To deal with the build-up of errors, the points passed from the potential field method should be as close together as possible, while still being far enough apart that they are 100 times greater than the greatest level of precision.

Simultaneous Localization and Mapping

One of the major methods used by autonomous robots when navigating an enclosed environment is the Simultaneous Localization and Mapping (SLAM) method. The main goals of SLAM are for the mobile robot to be able to localize itself in an environment filled with obstacles and for the robot to create a map of its enclosing environment using data from various external sensors. Because an elderly assist robot is designed to operate in a user's home or other similarly static environment, the SLAM algorithms provide a solid foundation for autonomous movement. In this project, a LMS 100 laser sensor from the SICK Company was attached to the mobile base to act as the main distance and obstacle detector.

As the robot moves around in the operating environment, the laser sensor emits a beam which reflects off of the surfaces of objects nearby. The laser sensor gathers the raw data about an object's location and outputs the data to a terminal via a program written in the Visual C++ language. The data gathered from the sensor is then passed to a C++ SLAM algorithm which filters the raw data and creates a virtual map of the environment by adding the known locations of obstacles to a new map. Once the map has been created, the robot can continue to move along the desired path to a goal location without hitting objects along the way.

The SLAM algorithms work with data provided by MATLAB consisting of the points along the desired path between the robot's starting point and the target goal location. Through the use of SLAM, the robot knows where it is along the desired path and if there are any obstacles that may impede forward motion. Path planning coupled with the SLAM algorithms allows the robot to successfully drive around the user's home and reach a pre-programmed target goal point completely autonomously. The use of SLAM theories and algorithms will allow the robot to better assist the elderly by autonomously reaching target objects and will also aid the robot in actively navigating to the user when the user location sensors are triggered.

Protecting Function Methodology

The robot's vision system consists of a Light Detection and Ranging (LIDAR) sensor. The robot's vision system has three fields namely – the inner, mid, and outer field. The inner field is 5 cm from the LIDAR, the mid between 5-10 cm, and the outer field outside of 10 cm. The robot implements a protecting function where if it detects an object in the inner field it will back up, in the middle field it will stop and in the outer field it will continue on its journey. This protecting feature was tested and proved to work correctly. The robot backed up when a person entered its inner field. When a person stood in the mid -field range the robot completely stopped and continued to move forward once the person moved to the outer field. Using these methods, we tested that the back up system would still work correctly in a case of autonomous navigation algorithm failure.

Base Testing Methodology

Once the programs for the basic motion control of the robot's base had been written into the Beckhoff PLC environment and successfully compiled on everyone's individual laptops, it was time to transfer the code to the robot itself for testing. In order to have immediate control over the robot during the testing phase, a remote control was used and its buttons were mapped to the various basic functions. The basic motions were successfully performed by the robot and adjustments to parameters were made based on observations of the base's performance.

When the time came to test the laser sensor capabilities while the robot was in motion, the SOPAS ET software package was used as the control GUI for the sensor output. As the robot moved forward, the laser continually scanned the surrounding environment and a visual map was updated with the placement of any obstacles that the robot encountered during its routine motion.

Arm Control

Arm Simulation

General structure of manipulator

To meet the service needs of the elderly, the robot should have the ability of movement and an arm with more than three DOFs. To better serve the elderly, our mechanical design has three joints that would give the arm a total of four DOFs. The manipulator consists of two links and one upright frame, which is fixed on the top plane of the base and has the function of supporting the arm. Two of the 4 DOFS are located in the shoulder joint of the arm and the others are located on the elbow (1 DOF) and the end effector (1 DOF). The mechanic model of the arm and manipulator is shown in Figure 8.

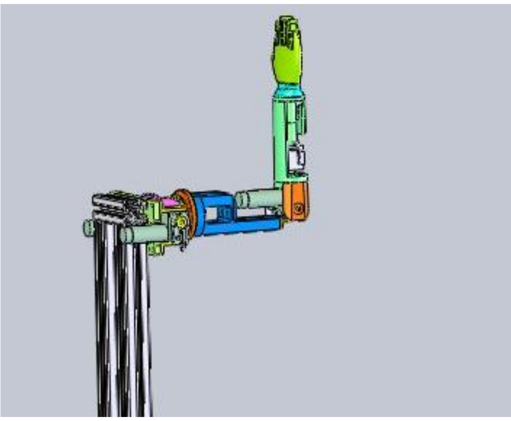


Figure 8: The mechanical model of arm and manipulator

DOFs-joint of manipulator

In order to save space, the shoulder joint is designed with two DOFs, and features three bevel gears as shown in Figure 9. Gears 1 and 2 are driven by two motors respectively, while the motion of gear 3 is driven by the synthesized motion of gears 1 and 2. When gears 1 and 2 have the same velocity and same rotation direction; gear 3 will rotate around axis X. When gears 1 and 2 have the same velocity in opposite rotation directions, gear 3 will rotate around axis Z. With this mechanical structure, a 2 DOF-joint was realized.

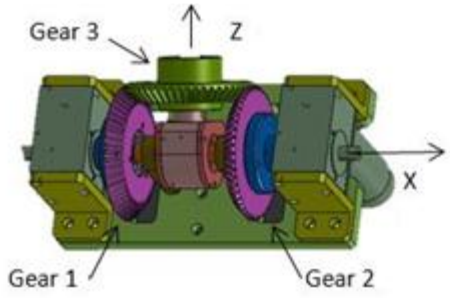


Figure 9: The structure of shoulder joint

The geographic method

Another part of the robot design was a working arm that can pick up objects. This is still in the process of being created but it is nearing its end stage. We used two methods for controlling the arm kinematics. The first was the geographic method. In this method, the coordinates x, y, and z are represented by equations that consist of angles and trigonometric formulas. We developed the forward kinematics for the final x, y, and z coordinates. These were as follows:

$$Y = L2 \times \sin(\theta_3 + \operatorname{atan}(\frac{1}{\sqrt{2} \times \cot\theta_1}) + \frac{L1}{\sqrt{1 + \cot^2\theta_1 + \cot^2\theta_2}})$$

$$X = L2 \times \cos(\theta_3 + \frac{1}{\sqrt{2}}\operatorname{atan}(\frac{1}{\sqrt{2} \times \cot\theta_1}) + \frac{L1 \times \cot\theta_1}{\sqrt{1 + \cot^2\theta_1 + \cot^2\theta_2}})$$

$$Z = L2 \times \cos(\theta_3 + \frac{1}{\sqrt{2}}\operatorname{atan}(\frac{1}{\sqrt{2} \times \cot\theta_1}) + \frac{L1 \times \cot\theta_2}{\sqrt{1 + \cot^2\theta_1 + \cot^2\theta_2}})$$

Because the implementation of inverse kinematics using the geographic method proved to be difficult, we then looked at using the ANFIS (Adaptive Neuro Fuzzy Inference System) method for calculating the inverse kinematics. The ANFIS method was also found to be difficult to implement as it required algorithms and procedures that were very complicated. Also, it could only map one theta of a joint to an x, y, and z coordinate. This was unsatisfactory because we had two thetas on the first joint which further complicated the use of ANFIS. Due to the difficulties of implementing ANFIS methods, we switched to using method of matrix calculation.

Forward Kinematics of D-H method

The forward kinematic analysis aims to find the position and the posture of the end-effector; the hand of the manipulator, based on the given angles of each joint. To solve the problem, we built the kinematic model, as seen in Figure 10, based on the Denavit-Hartenberg (DH) method.

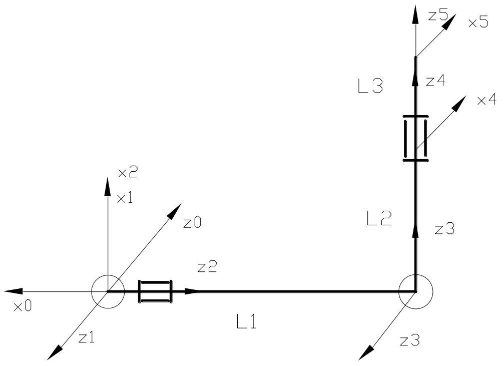


Figure 10: The kinematics model

We set up 6 coordinate systems and each coordinate system represents one DOF except O-xyz0. O-xyz0 is the reference system and O-xyz5 is the end-effector. The position of the end-effector is the relative position of the O-xyz5 to O-xyz0.

In the reference coordinate system, the destination is expressed as a 4×4 matrix:

$$P=\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vectors ***n, o,*** and ***a*** describe the post of the destination and ***p*** describes the position.

To express the coordinate system in the adjacent coordinate system, a transformation matrix is needed for calculation. From observing the basic transformation matrices, we can get the following matrix:

$${}^0_1T = rot(z, 90°) \times rot(x, 180°) \times rot(z, \theta_1)$$

$${}^1_2T = rot(x, 90°) \times rot(z, \theta_2)$$

$${}^2_3T = trans(0,0,L1) \times rot(x, -90°) \times rot(z, \theta_3)$$

$${}^3_4T = trans(L2,0,0) \times rot(y, 90°) \times rot(z, \theta_4)$$

So, the composite transformation matrix is:

$${}^0_4\mathrm{T}={}^0_1T \times {}^1_2T \times {}^2_3T \times {}^3_4T$$

In fact, ${}^0_4\mathrm{T}$ is the matrix P. In MATLAB, we define:

$$\mathrm{s1} = \sin\theta_1; \quad \mathrm{c1} = \cos\theta_1;$$

$$\mathrm{s2} = \sin\theta_2; \quad \mathrm{c2} = \cos\theta_2;$$

$$\mathrm{s3} = \sin\theta_3; \quad \mathrm{c3} = \cos\theta_3;$$

The forward kinematics result is:

$$\mathrm{px} = \mathrm{L2} \times (\mathrm{c1} \times \mathrm{s3} + \mathrm{c2} \times \mathrm{c3} \times \mathrm{s1}) - \mathrm{L1} \times \mathrm{c1}\ ;$$

$$\mathrm{py} = \mathrm{L1} \times \mathrm{s1} - \mathrm{L2} \times (\mathrm{s1} \times \mathrm{s3} - 1 \times \mathrm{c2} \times \mathrm{c3});$$

$$\mathrm{pz} = -\mathrm{L2} \times \mathrm{c3} \times \mathrm{s2};$$

Inverse Kinematics of D-H method

When analyzing the inverse kinematics problem, we need to know the matrix P, which includes both the position and the posture of the end-effector. Normally, the vector p is known, while the vectors n, o, and a are unknown. To achieve our objective, we simplify the matrix as follows: set the vector p, θ_4 as input data and θ_1, θ_2, θ_3, and vector n, o, and a as output data. θ_4 was set as an input because as the angle of wrist rotation it only affects the posture of the destination, without changing the position of the destination.

According to the forward analysis, we multiply the equation----${}^{0}_{4}\mathrm{T} = P$----both sides with inverse matrix of ${}^{0}_{1}\mathrm{T}$:

$$\mathrm{Inv}\ ({}^{0}_{1}\mathrm{T}) \times \mathrm{P} = {}^{1}_{4}\mathrm{T}$$

From the calculation, we can see that, in matrix${}^{1}_{4}\mathrm{T}$, the elements of last two columns have no relation with θ_4. In matrix $\mathrm{Inv}\ ({}^{0}_{1}\mathrm{T}) \times \mathrm{P}$, the elements of the last column have relations with the θ_1 and vector p. So when we solve the equation, we make the last column of the two matrices equal.

We can solve the value of θ_3 first, because $\mathrm{px}^2 + \mathrm{py}^2 + \mathrm{pz}^2$ only has relations with θ_3, which can be easily proved through the Law of Cosines. Then the value of θ_1 and θ_2 can be solved according to θ_3.

Finally, we can get the results as follows:

$$\theta 3 = \mathrm{asin}(\frac{\mathrm{L1}^2 + \mathrm{L2}^2 - (\mathrm{px}^2 + \mathrm{py}^2 + \mathrm{pz}^2)}{2 \times \mathrm{L1} \times \mathrm{L2}})$$

$$\theta 2 = \mathrm{asin}(\frac{-\mathrm{pz}}{\mathrm{L2} \times \mathrm{c3}})$$

$$\theta 1 = (-2)$$

$$\times \mathrm{atan}(\frac{\mathrm{py} + \sqrt{\mathrm{px}^2 + \mathrm{py}^2 - (\mathrm{L1} + \mathrm{L2} \times \mathrm{s3})^2}}{\mathrm{px} - \mathrm{L1} + \mathrm{L2} \times \mathrm{s3}})$$

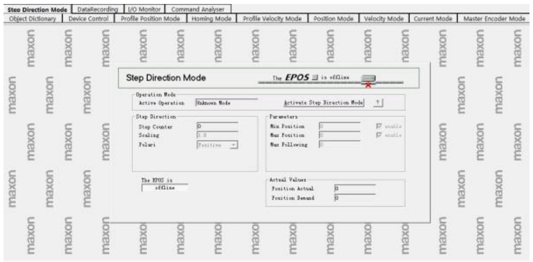
In MATLAB, after importing the Solidworks model into the simulation, the motion of the manipulator could be tested.

We input a constant signal to the position of destination, as well as the joint actuator blocks of px, py, and pz, the rotation angle, and the θ_4 actuator block.

Arm Control Implementation

We used the Beckhoff controller to connect the motor motivators with the PC. The Beckhoff system uses three main communication protocols including RS-232, CANopen and EtherCAT.

The RS-232 protocol is used to configure the hardware and in this case the motivators. UserInterface is the software that is used to communicate with the MAXON motors through the MAXON motor controller. UserInterface is also used to complete the configuration process of mapping of the software variables with the hardware variables so that we could use the hardware variables to control the motors. An example of the UserInterface GUI is shown in Figure 11.



Active Operation: Unknown Mode
Step Counter: 0
Scaling: 1.0
Polari: Positive
Min Position: 0 ☑ enable
Max Position: 0 ☑ enable
Max Following: 0
The EPOS is offline
Position Actual: 0
Position Demand: 0

Figure 11: UserInterface GUI

The CANopen protocol was used to connect the motivators together. CANopen is a network used in many everyday products consisting of multiple microcontrollers which need to communicate with each other. The two microcontrollers on the upper arm are connected together and communicate through CANopen communication to optimize functionality, as shown in Figure 12. The implementation of CANopen is based on CAN (Controller Area Network) which is implemented on the microcontroller.

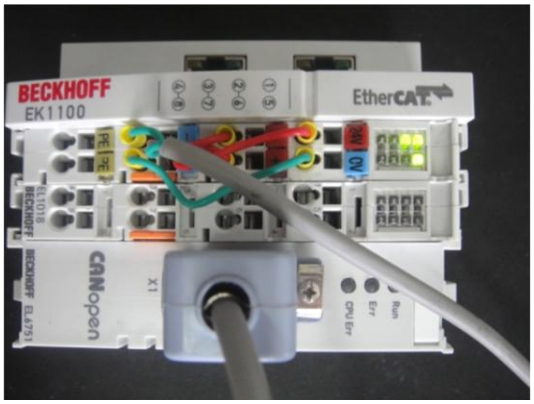


Figure 12: CANopen port

The EtherCAT protocol was used to communicate between the master motivators and the PC. EtherCAT (Ethernet for Control Automation Technology) is an open high performance Ethernet-based fieldbus system, as shown in Figure 13. It is appropriate here since we require a short data update time (cycle time) with low communication jitter and low hardware cost.

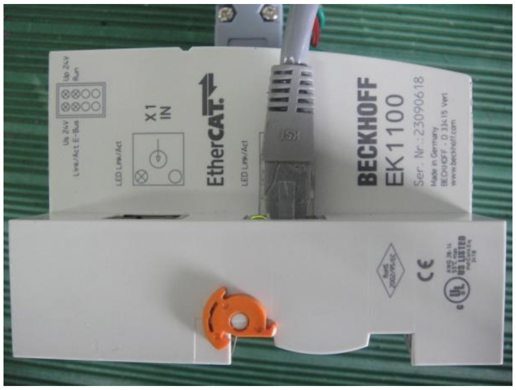


Figure 13: EtherCAT port

We used the software TwinCAT to connect the virtual PLC program to the hardware. We then downloaded the program into the motivators and drove the motors. The Beckhoff controller with the Ethernet cable connected is shown in Figure 14.

Figure 14: The Beckhoff controller

As for the position, speed, acceleration and jerk, we defined the limitations for all these perimeters to make sure that the arm wouldn’t be damaged. We use these limitations in the code and tested to make sure that the arm moves smoothly and moves within these limitations.

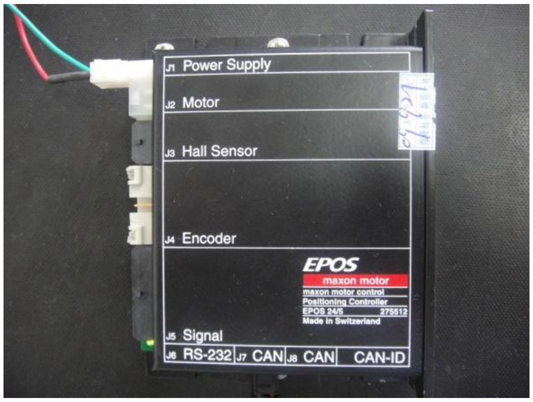


Figure 15: Motor driver

In the arm, we use two MAXON motors (313518) to drive joint 1, one MAXON motor (313518) to drive joint 2, and one MAXON motor (M090607) to drive joint 3. Joint1 can perform movement in two directions, rotation around the X axis and rotation around the Z axis. Figure 15 shows one of the MAXON motor drivers and Figures 16 and 17 show the placement of motors on the shoulder and elbow joints of the arm.

Figure 16: Maxon motor on 2-DOF joint

Figure 17: Motor on elbow joint

Arm Programming

The first step was to initialize the program, and set the initial parameter values and the temp variable values. Power should then be supplied to the motors, and any fault should be detected before ‘power enable’ and it should be cleared if errors are present. Once the motors receive power, the motors can be moved to any desired position. The fault should always be detected before any motion occurs. When the motors move to the target position, the position motion should be stopped. Another command can then be sent to activate motors or to simply disable the power.

Figure 18 shows the flow of the logic used to control the arm. We wrote a simple program according this chart to achieve some functions such as up and down motion and rotation of the arm. The programming procedure chart is just a template, to which we just needed to add the program function on the basis of the chart prototype. The flow chart shows how the motor would function for any given command and how it minimizes the risks in a general scenario.

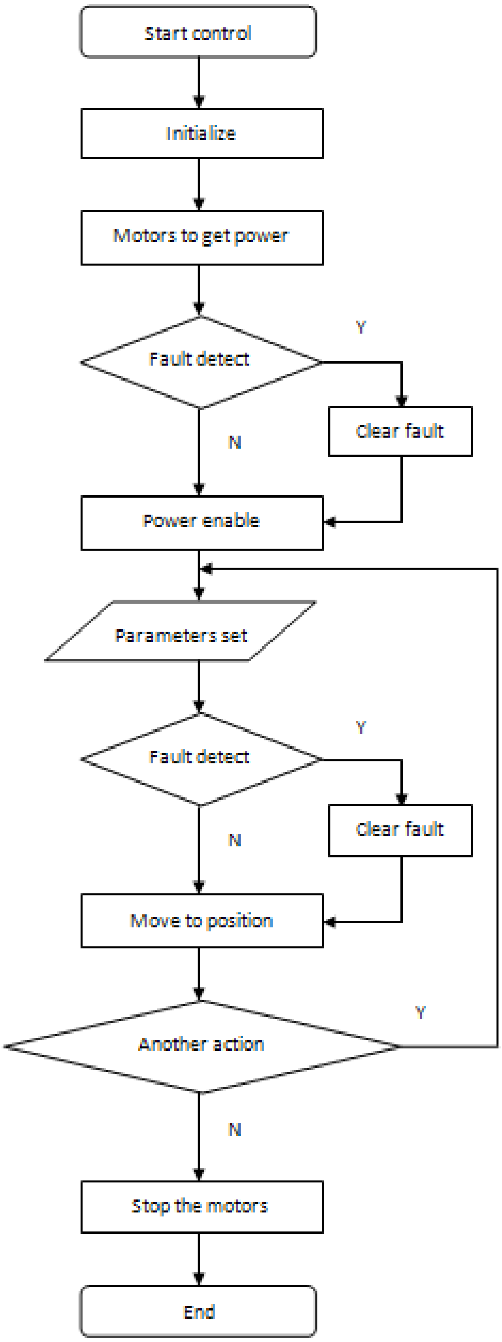
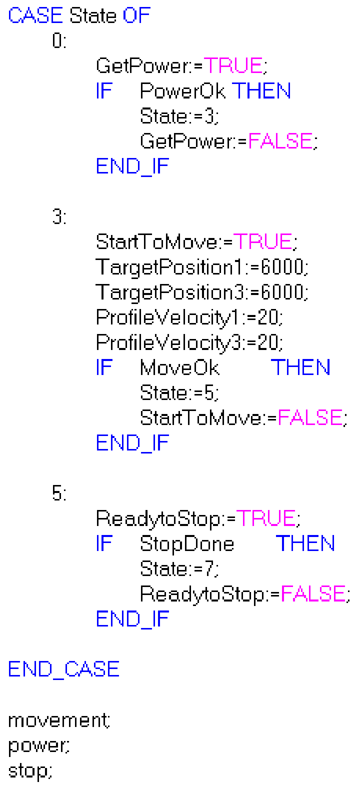


Figure 18: Program logic flow chart

Program we created to control the motors:



```
CASE State OF
    0:
        GetPower:=TRUE;
        IF    PowerOk THEN
            State:=3;
            GetPower:=FALSE;
        END_IF

    3:
        StartToMove:=TRUE;
        TargetPosition1:=6000;
        TargetPosition3:=6000;
        ProfileVelocity1:=20;
        ProfileVelocity3:=20;
        IF    MoveOk        THEN
            State:=5;
            StartToMove:=FALSE;
        END_IF

    5:
        ReadytoStop:=TRUE;
        IF    StopDone       THEN
            State:=7;
            ReadytoStop:=FALSE;
        END_IF

END_CASE

movement;
power;
stop;
```

Figure 19: Main program

Figure 19 shows the main Beckhoff program, which contains three parts according the programming procedure chart. We needed some function blocks to achieve these functions. Figure 20 shows the three core functional block diagrams needed to do any kind of movement with the arm. The first Function Block Diagram (FBD) provides power to the motor, the second FBD moves the motor according to the given direction from the Visual C++ interface and the last FBD stops the motor if there are no more pending commands from the user.

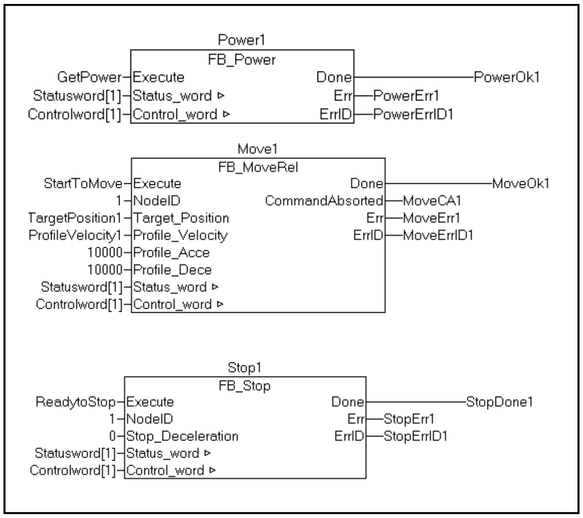


Figure 20: Functional Block Diagrams (FBDs) used to control the arm

All of the function blocks should be programmed first in the Program Organization Units (POUs) before they can be used in the main program, as demonstrated in Figure 21. The last three words in the main program are just to activate the function blocks in the POUs.

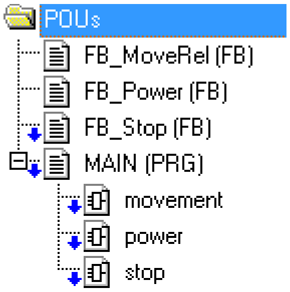


Figure 21: POU showing the functional block diagrams (FBD)

The program above cannot drive the motors without the global variables. We needed to define several global variables so that we could read or write the Process Data Object (PDO, Object for data exchange between several devices) without access, just because PDOs are transmitted in a non-confirmed mode. The variables used in the program are described below:

The Transmit-PDO (TxPDO) contains statusword, mode of operation display, position actual value, position demand value, and read profile velocity. The Receive-PDO (RxPDO) contains controlword, mode of operation, target position, profile velocity, profile acceleration, profile deceleration, and max following error.

Controlword and statusword: The device states may be changed using the controlword and / or according to internal events. The current state can be read using the Statusword.

Mode of operation and mode of operation display: The parameter mode of operation switches the actually chosen operation mode. The modes of operation display show the actual mode of operation.

Position actual value, position demand value and read profile velocity: Position demand value is generated by profile generator and is the set value of the position regulator. The actual position is absolute and referenced to system zero position. The read profile velocity shows the actual profile velocity.

Target position: The target position is the position that the drive should move to in profile position mode using the current settings of motion control parameters such as velocity, acceleration, and deceleration. The target position will be interpreted as absolute or relative depend on control word.

Profile velocity: The profile velocity is the velocity normally attained at the end of the acceleration ramp during a profiled move.

Profile acceleration: This value is used as acceleration in a position (or velocity) profile move.

Max following error: Maximal allowed difference of position actual value to position demand value. If difference of position demand value and position actual value is bigger, a following error occurs. (maxon motor)

The program also needed some temporary variable to make it feasible.

After all of the proper variables were set, the motors could then be controlled. As shown in the program, the first part shows how the motor gets the power. The second part shows how the motion of the arm could be controlled as desired. The program was constructed based on the logic flow chart. This program is only a prototype that needs modification. The program needs to be improved to respond to any accidents and increase the stability of its functionalities as a part of future work on this project.

Results and Findings

At the end of the project, we had completed a prototype robot capable of demonstrating basic autonomous navigation and user-controlled arm movement. In its current state, the robot cannot operate as a fully-functional elderly assist robot, but with future improvement, it could feasibly be developed into a commercial product. The autonomous navigation feature allows the robot to navigate from point A to point B while avoiding obstacles along the motion path. Our robot is capable of detecting objects in its path and is able to avoid them by backing up, which is the purpose of the protecting function. Unfortunately, the SLAM algorithms for obstacle avoidance described in the Methodology section were not able to be fully implemented due to time constraints. However, through research we discovered many sample SLAM software packages which can be implemented in the future or used as foundations for more code. Because the SLAM algorithms were not fully implemented, the autonomous navigation that the robot is able to perform is very basic. It can travel in the hall by avoiding the two walls. It also uses the LMS algorithm to calculate the midpoint between the two walls and travels along the path that is lies in the center of the two walls.

The final robot design also included a functional robotic arm. The arm can navigate from one point to another but has to be manually given inputs. The arm can be further developed to perform automatic movement, which means the robot will be able to reach a destination without the user providing destination coordinates. In order to implement accurate automatic motion, the arm would need to have a camera that can identify an object, determine its location, and use the already implemented arm control program to navigate to that position. In conclusion, we successfully built and programmed a prototype foundation for an elderly assist robot which can be further expanded on by future groups to include more powerful features.

Conclusions

Due to time constraints, we were only able to implement a basic form of autonomous navigation and a user controlled arm. The autonomous navigation capabilities include being able to travel in a hall and utilize a protecting feature when an object gets too close. The arm of the robot has to be manually given inputs for it to navigate from its current position to the next position. Thus, we were able to accomplish the two functions that we set out to do. Also, with these two functions we laid the foundation for a robot that with later development can become a fully capable elderly assist robot.

Future Works

In the future, we would also like the robot to implement the path planning feature that has been shown to work in the simulation but did not work on the physical robot. This just requires more testing. Currently, the autonomous navigation is at a basic level. We could not further develop this feature because of time constraints. We would also like the robot to have a complete form of autonomous navigation. This would mean the robot should be able to detect an object, track itself, and be able to locate its destination. This would require another 6-8 months of work. We believe that this could lead to future MQPs which could possibly even involve students from more majors from Worcester Polytechnic Institute. Also, implementing these features requires a good background in Electrical and Computer Engineering and in Computer Science.

The arm part of the robot can also be improved. Currently, the arm can only move to the location provided by the user. But an object location feature that would make the movement of the arm fully autonomous through the use of a camera could also be implemented. An object identification feature could also be implemented to help the robot identify specific objects and assist the elderly in a more specific way. The object identification feature is also a very difficult task and could also lead to another MQP for CS and ECE majors.

References

Ahrary, Alireza, Amir A.F. Nassiraei, and Masumi Ishikawa. "A study of an autonomous mobile robot for a sewer inspection system." Artificial Life and Robotics 11.1 (2007): 23+. Academic OneFile. Web. 17 Apr. 2010.
Asakura, M. M. (1999). Manipulator visual servoing and tracking of fish using a genetic algorithm.

Bejczy, A. K. (n.d.). *Robot Arm Dynamic Control by Computer*. Retrieved from NASA Space Telerobotics Program: http://ranier.hq.nasa.gov/telerobotics_page/technologies/0414.html

B. Zhang,W.Chen, M. Fei, An Optimized Method for Path Planning Based on Artificial Potential Field

Can Tang, Jianwei Zhang, Sheng Cheng, “Kinematics Analysis for a Hybrid Robot in

Chengjun Ding , Ping Duan , Minglu Zhang , Huimin Liu, “The Kinematics Analysis of a Redundant Mobile Manipulator

Central Intelligence Agency. (2010). *CIA - the world factbook - china.* Retrieved 04/15/2010, 2010, from https://www.cia.gov/library/publications/the-world-factbook/geos/ch.html

D-H. Kim,J-H. Kim, A Real-time Limit-cycle Navigation Method for Fast Mobile Robots and Its Applications to Robot Soccer, 2002

D.M. Wilkes, A. A. (1999). Designing for Humanrobot Symbiosis. *Industrial Robot* , 49–58.

Gregory Fischer. WPI Professor – Mechanical Engineering.

Heping Chen, G. Z. (2009). Assembly on moving production line based on sensor fusion. *Assembly Automation* , 257–262.

International Day of Older Persons. (n.d.). Retrieved 10 08, 2010, from United Nation: http://www.un.org/zh/events/olderpersonsday/background.shtml

Industrial Robot: An International Journal , pp. 278-289.

J.C. Latombe, Robot Motion Planning Kluwer, Academic Publishers Boston, MA, 1991

Jonathan Dixon, Oliver Henlich 10 June 1997
http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/jmd/

Lenser, S., & Veloso, M. (n.d.). Retrieved 04 05, 2010, from Visual Sonar: Fast Obstacle Avoidance Using

Ma Bojun, Fang Yongchun, Zhang Xuebo, "Inverse Kinematics Analysis for a Mobile Manipulator with Redundant DOFs." Proceedings of the 26th Chinese Control Conference July 26-31, 2007, Zhangjiajie, Hunan, China

Maxon motor. (n.d.). Firmware Specification.

MedicineNet. (1998). *Definition of ADLs (activities of daily living).* Retrieved 4/15/2010, 2010, from http://www.medterms.com/script/main/art.asp?articlekey=2152

Minimally Invasive Surgery." Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics December 19 -23, 2009, Guilin, China

Monocular Vision: http://www.cs.cmu.edu/~mmv/papers/03iros-sonar.pdf

Ponsler, A. H. (2010, April 30). Blob Detection with Open CV. (J. Noser, Interviewer)

" Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, China September 2008

P. Schrock, F. Farelo, R. Alqasemi, and R. Dubey, "Design, Simulation and Testing of a New Modular Wheelchair Mounted Robotic Arm to Perform Activities of Daily Living." 2009 IEEE 11th International Conference on Rehabilitation Robotics Kyoto International Conference Center, Japan, June 23-26, 2009

(n.d.). Retrieved from Lynxmotion, Inc.: http://www.lynxmotion.com/p-419-rios-ssc-32-arm-control-software.aspx

RedOrbit. (2009). *Many chinese elderly feel alienated, lonely*. Retrieved 4/15/2010, 2010, from http://www.redorbit.com/news/international/1626471/many_chinese_elderly_feel_alienated_lonely/

RoboRealm. (2010). *Fiducial*. Retrieved May 2010, from RoboReam: vision for machines.

Systems With Applications 37.1 (2010): 426+. Academic OneFile. Web. 19 Apr. 2010.(n.d.). Retrieved 04 05, 2010, from http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vo.

T. Laue, T. Rofer A Behavior Architecture for Autonomous Mobile Robots Based on Potential Fields, 2003

WuJingkai, HuangYuancan, WangShixing, Anti-collision Path Planning for Mobile Robot Based on Potential Field Method, 2007

Xie Chuanjiao. (2007, Aging population a major challenge. *China Daily,* pp. 2. Retrieved from http://www.chinadaily.com.cn/cndy/2007-03/12/content_824828.htm

XiongYoulun. The Elementary technology of Robot. Wuhan. Huazhong University of Science &Technology Press. 1996

Y. B. Li, Z. L. Jin, “Kinematic Analysis of a Novel 3-DOF Hybrid Mechanical Arm.”

Yim, Jaegeol, et al. "Improvement of Kalman filters for WLAN based indoor tracking." Expert

http://www.cs.cmu.edu/~mmv/papers/03iros-sonar.pdf

http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=102-2017-ND

http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=102-1307-ND

http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=GH3075-ND

http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=IN3003CAM2-ND

http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=NUF6410MNT1GOSTR-ND

http://www.robotshop.com/ProductSearch.aspx?qs=encoder

http://www.robotshop.com/content/PDF/sfe-cmos-camera-640-480-manual.pdf

http://www.robotprice.com/robotprice_news.html

http://www.amazon.com/Erector-Spykee-The-Spy-Robot/dp/B000N6470A