

Development of a Robotic Manipulator to Assist Disabled Children

Gateway Engineering Coalition
Concurrent Design and Manufacturing Project 1997



by

Zan Kuang, Marcus Paula-Pereira, Javier Velez
Prof. Stan Wei, Advisor (The Cooper Union)

Demetrios Christoforatos, Ben Gontarz, Steven Kopf, Michael Rippo,
Spiros Koulas, Prof. Jack Zhou, Advisor (Drexel University)

Michael Collie, Hudson Etienne, Alvaro Garcia, Jr., Cowander Gillespie, Snehal Patel,
Nirmala Padmanaban and Pramod Palavi,
Prof Golgen Bengu, Advisor (The New Jersey Institute of Technology)

Jay Hladish, Ben Krupp, Seth Larris, Kevin Narcisi, David Nasse,
Tanya Schneider, Prof. Gary Kinzel, Advisor (The Ohio State University)

Graeme Lemmer, Solomon Steplight, Tom Sugar,
Prof. Vijay Kumar, Advisor (University of Pennsylvania)

May 17, 1997

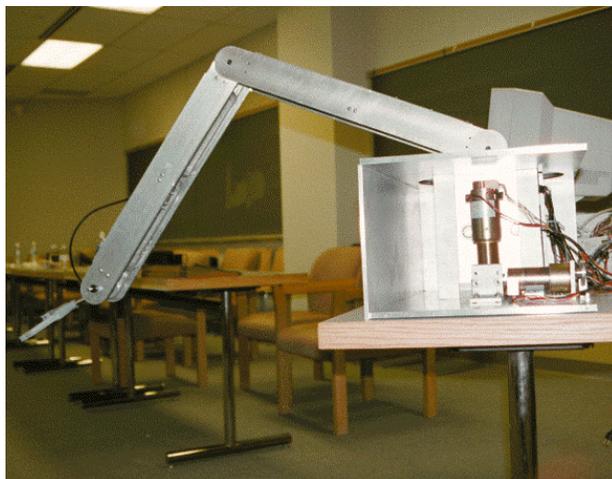


TABLE OF CONTENTS

<u>SECTION</u>	<u>TITLE</u>	<u>PAGE NO.</u>
I.	ABSTRACT	5
II.	INTRODUCTION	5
III.	BACKGROUND	8
IV.	PROBLEM STATEMENT	17
V.	PHASE I	18
	A. Design of a Wheelchair-Mounted Robotic Manipulator (Cooper)	19
	B. Five Degree Electromechanical Robotic Manipulator (Drexel)	30
	C. A Power Wheelchair Mounted Robotic Arm (N.J.I.T.)	41
	D. A.R.M. (Assistive Robotic Manipulator) (O.S.U.)	52
	E. W.A.R.M. (Wheelchair Assistive Robotic Manipulator) (Penn)	59
VI.	SPECIFICATIONS FOR PROTOTYPE DESIGN	66
VII.	PHASE II	67
	A. Arm Skeleton (Drexel)	67
	B. Power Transmission System (O.S.U.)	74
	C. Motor/Gearbox Selection and Sizing/Design of Computer Interface (Penn)	80
	D. Design of User Interface (Cooper)	88
	E. The Final Product	95
VII.	PROJECT LOGISTICS	96
VIII.	CONCLUDING REMARKS	99
X.	APPENDIX	101
	A. Sample Calculations	102
	B. Simulation Code	108
	C. Joystick Code	120

LIST OF TABLES

<u>NUMBER</u>	<u>TITLE</u>	<u>PAGE NO.</u>
1	Age distribution of electric-wheelchair users	9
2	Input device familiarity	11
3	Percentage of disabled in different countries	11
4	Percentage of disabled with reduced functions	12
5	Transmission gear sizing	78
6	Final motor/gearbox selections	82
7	Torque limiter specifications	83
8	Electronic assignments of the D15 connector	90

LIST OF FIGURES

<u>NUMBER</u>	<u>TITLE</u>	<u>PAGE NO.</u>
1	Age groups	9
2	Marital status	10
3	Employment status	10
4	Sex distribution of subjects	10
5	Analysis of subjects living at home	10
6	Disability groups	13
7	Average time/activity/subject/day	13
8	Link member 1 of Design I	20
9	Link member 2 of Design I	20
10	Link member 3 of Design I	21
11	Link member 4 of Design I	21
12	The gripper assembly of Design I	22
13	Gripper accessories of Design I	22
14	Member 1 of Design II	23
15	Member 2 of Design II	23
16	Member 3 of Design II	24
17	The gripper assembly of Design II	24
18	Gripper accessories of Design II	25
19	Cross-section of middle part of arm member	25
20	Cross-section of end part of arm member	26
21	Back view of arm member 1	26
22	A Robix rendition of Design I	27
23	A Robix rendition of Design II	28
24	A Robix rendition of Design III	28
25	Rendered Image of Design I	28
26	Rendered Image of Design II	29
27	Rendered Image of Design III	29
28	Door way view	32
29	Arm reach	32
30	Arm description	33
31	Standard children's wheelchair	34
32	Underside view of mounting	34
33	Arm installation	35
34	Base assembly	36
35	Base detail	37
36	Shoulder detail	37
37	Cross-section of arms	38
38	Elbow detail	39
39	Gripper detail	39
40	Alternate Mounting	41
41	The dimensions of Power Wheelchairs for Children	43

42	and the robot reach length requirements	
43	Robot arm length calculations	44
44	Folded robot arm design	45
45	Retractable robot arm	45
46	Dimensions of the robot arm	45
47	Sequence of motions of the robot arm	46
48	Overall view of the robot arm	47
49	Wheelchair stability	48
50	Arm transmission system	53
51	Wrist transmission system	54
52	Structural frame of arm linkage	55
53	Gripper assembly	55
54	Final linkage design	56
55	<i>Embedded</i> coordinate system	58
56	The W.A.R.M.	60
57	W.A.R.M. assembly drawing	61
58	W.A.R.M. in tucked position	62
59	Motor control system	64
60	Motor control system	64
61	Hardware setup	65
62	Arm link	68
63	Structural elements of the arm linkage	69
64	Elbow joint components	69
65	Lower arm tensioning system	70
66	Arm covers	71
67	Rotating base	72
68	Side view of entire base	73
69	Base tensioning system	73
70	Original cable design	74
71	Cable chain	75
72	Pitch link chain	75
73	Torque transmitting pulleys	76
74	Transmission pulley design	76
75	Differential	77
76	Gripper with shielded cable	77
77	Gear train architecture	79
78	Rotation of arm	79
79	Simplified robot	85
80	D15 Connector	89
81	Game port – PC bus	90
82	Joystick axis bit definition	91
83	Joystick working area and calibration parameters	92
84	Imaginary three-dimensional input vector	93
85	CompUSA joystick controlling elements	93
86	Gateway Coalition student representatives	95

87	Final product	95
88	Ohio State and Drexel in a video-conference	96
89	Drexel University's Gateway Web Page	97
90	Original assembly design	99
91	Proposed assembly design	100
92	Gateway Coalition students	100
93	Final Product	101
94	Video-Conference	102
95	Drexel University's Web Page	103

I. ABSTRACT

This year's Gateway project, funded by the National Science Foundation, has developed a robotic manipulator to assist children with disabilities and confined to wheelchairs. The multi-university design project was accomplished using concurrent engineering methodology through a joint venture between Cooper Union, Drexel University, New Jersey Institute of Technology, Ohio State University, and University of Pennsylvania. Undergraduate engineering students integrated design and manufacturing processes in the fabrication of the manipulator from initial design conceptualization to a finished product. Phase I required each team to develop a conceptual design for the manipulator using common design criteria established at UPenn in October 1996. Following a critique of the five conceptual designs at Cooper Union in January 1997, the teams combined their engineering expertise and resources to implement key design features to manufacture a prototype manipulator. Each university was assigned a specific task in Phase II of the project. These five subsystems were presented and finalized at the University of Pennsylvania in March 1997. The final design was presented at Drexel University on May 17, 1997, along with a demonstration of the manipulator.

II. INTRODUCTION

Multi-University Design Project Concept

The intent of the Gateway Coalition in establishing this multi-university design project is to develop the human relation skills of undergraduate students by shifting the focus of ordinary undergraduate engineering education from classroom theory to include the practical application of the course content. In this way, the learning gained by the students encourages and enhances their teamwork skills necessary for today's industry. One goal of this design project, consisting of teams of students from five universities, is to provide significant interaction among the schools, in which the students are able to openly communicate and cooperate to overcome inevitable technical problems. Concurrent Engineering methodology is important in the development of the project concept as it serves as a powerful tool to enhance the interaction between the teams using advanced communication systems. It employs the practice of integrating marketing, manufacturing, design issues, and requirements to follow the fabrication of a product from its initial conceptualization to a finished prototype.

Project Logistics

The five universities, The Cooper Union, Drexel University, New Jersey Institute of Technology, Ohio State University, and University of Pennsylvania took part in this year's project in the development of the design and manufacture of a wheelchair-mounted robotic manipulator to assist children with disabilities. The student teams were introduced to the project and to each other at an initial design meeting, which took place in October 1996 at the University of Pennsylvania. At this meeting, the scope and the goals of the

project, along with the chosen design boundary conditions and requirements were discussed in detail. At the conclusion of this meeting, each university set up a World Wide Web page on the Internet so that the student teams could find information on the project with weekly updates and progress reports. The project consists of two phases, Phase I took place from October 1996 to January 1997, and Phase II lasted from January to May 1997. The five schools used frequent video-conferencing and electronic mail to communicate with each other during both phases, as well as some on-site visits to some of the universities.

Design Conference at University of Pennsylvania

On October 13, 1996, the student teams representing various disciplines of engineering from the five universities, along with their graduate and faculty advisors, assembled at the University of Pennsylvania for an introductory design meeting. Dr. Vijay Kumar, faculty advisor for the University of Pennsylvania team, introduced the intent and goals of the project in which the teams would combine engineering backgrounds to develop a robotic manipulator to assist disabled children with limited movement in their hands and arms. The project consists of two phases. In the first phase, each school would independently develop a kinematic design of the manipulator. The second phase would entail all five universities working as a sufficient engineering team to manufacture a prototype manipulator for the chosen design configuration. The teams discussed the scope and timeline of the project and worked profusely to develop design constraints. For most of the meeting, each team voiced opinions on who should be the candidates for such a design concept. Once the teams agreed on the assembled design specifications and constraints for the project, the advisors were informed of the students' obligations. The meeting was adjourned with specific instructions for each university's design team for the two phases of the project.

Goal of the Project

For this year's Gateway-sponsored Concurrent Engineering project, the goal is to develop a robotic manipulator which will assist children who are confined to wheelchairs and have limited arm and hand use. Using concurrent engineering principles in which the phases of design are completed sequentially, the teams from the Cooper Union, Drexel University, New Jersey Institute of Technology, Ohio State University, and University of Pennsylvania, worked independently in Phase I of the project to develop an optimal kinematic configuration of the manipulator. The teams presented their ideas at the January design meeting, at which the most sufficient and economical design configuration of the manipulator was selected. In Phase II, the teams combined engineering disciplines to work together and exchange ideas to generate a prototype manipulator. The final design of the manipulator, as well as a prototype model, was presented at two final meetings, the first in early May for the faculty advisors, and the second meeting in mid-May for the members of the Gateway Coalition. At both presentations, a working demonstration of the prototype ensured that the design of the manipulator is sufficient for production.

Scope of This Report

This design report, prepared in conjunction with members from all five design teams, will ultimately describe the design configuration for the working prototype of the manipulator, as well as each school's role in the two phases of the project. The report is divided into eight subsections arranged in the following order:

- **Background:** The report will discuss previous work from past industry and university projects related to this year's project. A market analysis will then be discussed to inform the reader of the current market for the product as well as the need for such a device.
- **Problem Statement:** A concise statement of need will educate the reader with the intent of the project, a description of the design's end-user, and the design configuration's solution to the problem.
- **Phase I:** Each of the five universities' preliminary kinematic designs and key features will be discussed in detail for Phase I of the Gateway project.
- **Design Specifications for Prototype:** Using feedback from the January design meeting, the design configuration will be revised to allow for the manufacture of the manipulator in Phase II of the project. The highlights of the January meeting will be discussed, followed by the chosen configuration of the manipulator using key elements from each team's preliminary design specifications, and detail of the power transmission for the manipulator will conclude this subsection of the report.
- **Phase II:** This subsection will consist of individual discussions of each team's responsibilities for the manufacture of the prototype manipulator. Drexel is responsible for the design of the arm skeleton and base of the manipulator. Ohio State will focus on the power transmission of the arm from the motors to the endpoint. University of Pennsylvania will develop the program code to control the actuation of the motors as well as motor selection for the final prototype. Cooper Union will concentrate on designing the user interface to allow it to communicate with the motor actuation, as well as graphical simulation and analysis for the final prototype design. New Jersey Institute of Technology will prepare an extensive market analysis report for the manipulator as well as assist the other universities in manufacturing the prototype.
- **Logistics:** This subsection will describe the methods the five teams used to communicate their ideas and opinions to each other outside of design meetings. It will mention the extensive use of video-conferencing between the schools as well as frequent use of electronic mail and the teams' World Wide Web pages for the project. The three design meetings of all five universities will be discussed in further detail as to how the final design, manufacture and assembly decisions were accomplished. Finally, this subsection will be completed with a conclusion for the effectiveness of these communication tools used in this project.
- **Concluding Remarks:** The overall opinions and concerns of the project's design and format will be discussed here in which the teams have the opportunity to express their thoughts on the project's achievements, the important aspects of the project, lessons learned, and what can be done in another engineering project to improve product

development. Also, future work for this project will be mentioned in order to increase and maintain improved form and functionality of the manipulator.

- **Appendices:** Programming and joystick code, along with part and assembly drawings, and a comprehensive parts list will be included in this subsection.

III. BACKGROUND

Before any attempt to develop a conceptual design of the manipulator, it was necessary to research the current status of robotic designs. Two manipulators, the Helping Hand and the Manus Manipulator, are the most feasible designs currently on the market. Both robotic devices were manufactured in England. Although these designs were impressive, a careful inspection of each one shows that improvement was waiting in the future.

The Helping Hand is a wheelchair-mounted manipulator that operates on five degrees of freedom. It is made out of a lightweight plastic material, similar to polyvinylchloride. It is actuated by electric servomotors and is controlled by a computer microprocessor interface. Movements from the device are based on joint control only, that is, the computer interface moves one joint at a time to reach a destination. This motion is found to be unstable (jerky) and also disadvantageous because this type of motion does not simulate the human arm. Another concern about the Helping Hand is that it costs \$10,000 to manufacture, a price well beyond the scope of this project.

The Manus Manipulator also mounts to a wheelchair, but it operates on seven degrees of freedom. It is made out of a lightweight aluminum and is also actuated by electric servomotors. The main difference between the Helping Hand and this manipulator is that all the motors are mounted in the base, decreasing the weight of the arm itself. This is an advantage because it eliminates unstable motions. More importantly, this device is controlled by end point manipulation, that is, the computer interface (a keyboard) calculates the most favorable path to a destination and moves all joints simultaneously in reaching this point. Unfortunately, the computer interface is very complex for a child to use, and it cost \$65,000 to manufacture, a price that is unaffordable to most patients.

A brief yet concise market analysis found that 36.1 million people are disabled in the United States. Approximately four million of these patients are children under the age of eighteen, and about 1.5 million are confined to wheelchairs. These studies show that there is an extended market with a considerable amount of demand for the product. The challenge here is to design a safe, low-cost, simplistic device that can satisfy these market needs. The following sections discuss the project market analysis.

Survey of Powered Wheelchair User Requirements

In rehabilitation robotics, many designs have failed. This can be attributed to some basic design flaws, namely, expense, ergonomic factors, and control difficulties. Probably the

most important reason for these flaws is that a detailed need and requirement analysis of disabled people was not necessarily part of the engineering design process. However, it is critical for a designer of today's cost conscious, competitive market, to develop the requirements of a rehabilitation robotics based on the customers needs.

In literature, there are several surveys completed in the U.S. and Europe to investigate and evaluate the needs and abilities of disabled people. However, there have been very few surveys of robotic aid with focus on electric-wheelchair users with little or no upper body ability. One such survey completed in United Kingdom [Prior S. P., "An electric wheelchair mounted robotic arm: A survey of potential users," Journal of Medical Engineering and Technology, Volume 14, Number 4, 1990, pages 143-154] shows the following age distribution of electric-wheelchair users.

Table 1: Age distribution of electric-wheelchair users

Age	Distribution of Users
Under 16	3%
16-35	20%
36-65	23%
Above 65	10%

It is clear from the above statistics that there are more adult users of electric-wheelchairs than children. According to the same survey the average electric wheelchair user is 40 years old, single (68%), living at home (58%) with family support (69%) and without any paid employment (79%).

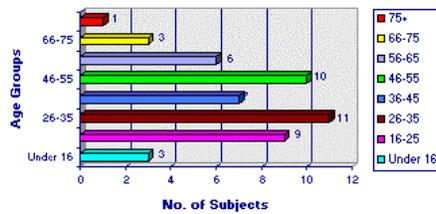


Figure 1: Age groups

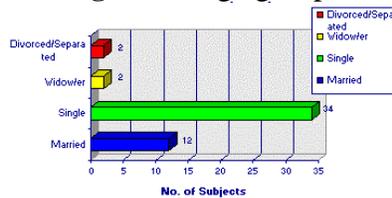


Figure 2: Marital status

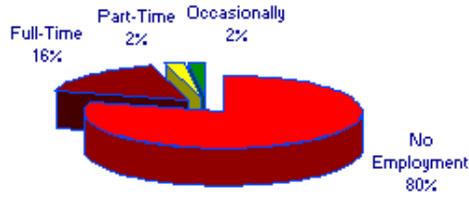


Figure 3: Employment status

Further, the average user is likely to be male (56%) than female (44%). This may be due to the high number of spinal cord injuries caused by male participation in dangerous sports such as diving, skiing, martial arts, etc. On the other hand, women users for the most part are those that suffer from conditions resulting from old age. The majority of wheelchair users are those living at home, followed by those living in hospitals, care homes or hostels. Hence, the majority of the robotic aids must be designed to operate within the confines of the home environment. Of the subjects living at home, the majority receive help from family members with no outside care. A robotic aid will not only relieve some burdens for disabled people but also help to regain self-esteem.

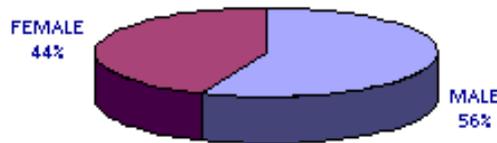


Figure 4: Sex distribution of subjects

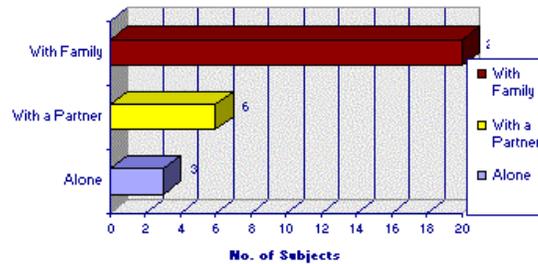


Figure 5: Analysis of subjects living at home

Table 2: Input device familiarity

Type of Input Device	Familiar Users
Joystick	84%
Remote Control	72%
Head Movement Sensor	
Rollerball Control	
Chin Operated Control	

Eye Movement Control Ultrasonic Sensor Voice Activated Suck-Blow Switches Mind Switch (EEG based)	Few %
---	-------

Another survey revealed that a mobile robotic aid device would be of far greater use than a workstation-based device [Hilmann, M.R., " A feasibility study of a robot manipulator for the disabled," Journal of Medical Engineering and Technology, 11, 160-165.]

Table 3: Percentage of disabled in different countries

Country	Percentage
Denmark	10.0-12.0%
Finland	5.2-8.3%
France	5.0-8.3%
BRD	10.8-13.1%
Ireland	3.3-5.0%
Italy	1.7-17.1%
Luxembourg	10.0-11.0%
The Netherlands	9.5%
Norway	12.0%
Portugal	7.4%
Spain	25.0%
Sweden	12.0%
Switzerland	1.6%
U. K.	7.3%
Canada	13.0%

Statistics

The United Nations usually estimate that about ten per cent of the population in various countries may be considered disabled. However, there is great variation in the incidence of disabilities in the statistics from different countries. These differences may be caused by different criteria for reporting, degrees of industrialization, rate of traffic accidents, and participation in wars, for example. Some of these differences between countries should be taken into account when considering a probable incidence of various disabilities in individual countries.

Proposed Estimates

To obtain a realistic estimate of the number of people with disabilities, it seems necessary

to make individual estimates of specifically defined impairments and disabilities, where the definitional criteria are clearly related to the purpose of the survey. This will also make it reasonable to compare statistics, and, if necessary to discuss definitional criteria.

Percentage of People with Reduced Functions

Table 4: Percentage of disabled with reduced functions

Type of Disability	Percentage
Blind	0.2
Partially Sighted	2
Profoundly Deaf	0.1-0.2
Hard Hearing	10.0-15.0
Cannot speak clearly	0.4
Cannot use fingers	0.2
Cannot use one arm	0.2

Types Of Diseases Causing Disability

The most prevalent disability is spinal cord injury (24%) followed by multiple sclerosis (16%), rheumatoid arthritis and cerebral palsy (10%). The 84% of the survey subjects indicated that they would consider buying a robotic arm mounted to an electric wheelchair. The tasks that they would most like to be able to do but cannot, due to their disability, starts with reaching, stretching and grip, and continues with gardening, as listed in the following figure. This figure illustrates the possible task list activities.

Funding

New technological developments have raised the expectation that people with disabilities can employ assistive devices to function more independently in their homes and communities, and pursue vocational goals they were unable to accomplish before.

Expectations are often dashed or delayed when people seek financial assistance to purchase devices and are faced with a maze of funding sources, eligibility restrictions,

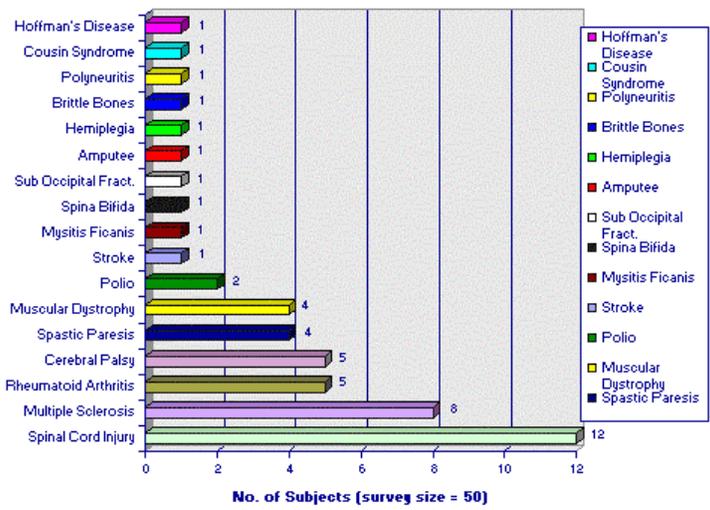


Figure 6: Disability groups

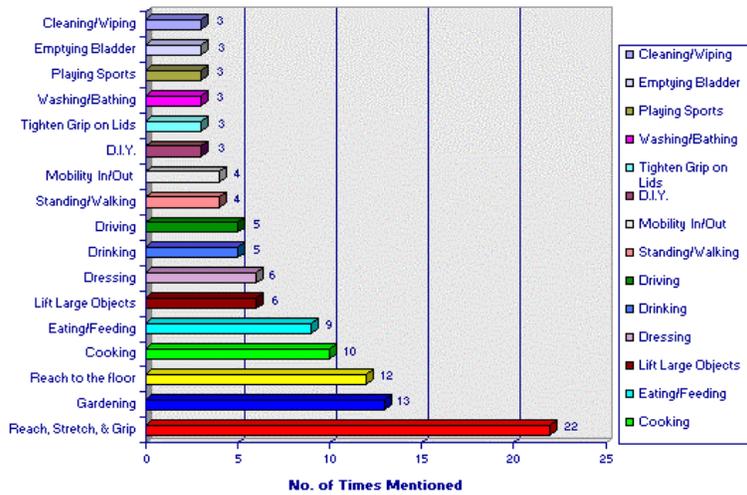


Figure 7: Average time/activity/subject/day

paperwork requirements, and rules and regulations governing the funding of such equipment. This fact sheet introduces tools that will help:

- Select an appropriate device
- Identify funding sources
- Prepare a justification statement that convincingly demonstrates to the funding sources that the device is appropriate.

Identifying Equipment And Technology Needs

Before considering funding sources and how to use them, consumers and their families

should take a serious look at what the consumer wants and needs to do independently. This process is a type of a task analysis.

A general listing of activities might include the following:

- Early childhood and pre-school
- Transportation
- Elementary and secondary education
- Recreation and sports interests
- Post-secondary education and training
- Wheelchair or other ambulation aids requirements
- Employment and career development
- Prosthetic and orthotic needs
- Sensory and communication.

Under each of these general areas, there may be several specific functional tasks that could be made easier by the use of assistive technology.

Selecting Funding Sources

Sources of financial assistance for the purchase of assistive technology are many. Each has different eligibility criteria. Most require that an application for services or financial assistance be completed. All have specific rules regarding what equipment can be purchased. Most primary sources of financial assistance are programs authorized and funded by the federal government, programs such as Social Security, operated by federal agencies. Others, such as vocational rehabilitation programs are federal/state partnerships and are operated by state agencies.

In addition to government agencies, some private organizations can be sources of funding. Foundations, corporations, and trust funds are potential sources of direct financial assistance. Some research at a local library can help identify an appropriate source of funding. Usually, the person seeking funds must write a funding proposal or justification statement indicating the nature of the need and how the money will be spent. This may be a letter or it may require a more formal write-up.

Civic organizations and service clubs are another resource. They often raise funds on behalf of individuals in need of financial assistance for the purchase of assistive devices. Lions Clubs, Kiwanis, Rotary, and Pilot International are examples of organizations that have a history of assisting people with disabilities.

Finally, there are several organizations that have established low-interest, revolving loan programs. These programs are particularly valuable to individuals and families whose income makes them ineligible for programs based on financial need. The low-interest loan program administered by the National Easter Seals Society was established specifically to aid people with the purchase of assistive technology. The American Foundation for the

Blind, in cooperation with Xerox Corporation and the Bank of Boston, established a low-interest loan program to help people who are blind pay for the purchase of personal readers. A few manufacturers of assistive technologies provide financing or other forms of time payments, but this is a rare practice. However, manufacturers and vendors are often sensitive to the difficulties that consumers face when trying to find funding sources. Prentke Romich Company of Wooster, Ohio, is one example of a vendor attempting to aid its current and potential customers by educating them about the funding process. At the end of this document is a copy of their handout "Funding Devices and Services in Augmentative and Alternative Communication," which provides additional information about funding sources and an overview of the funding process in the context of communication devices.

Preparing A Justification Statement

Some funding sources require the applicant to prepare a justification statement; this is particularly true for government programs. When the funding source is a public or private insurance policy, either the expected beneficiary, a physician, or a therapist usually must submit a statement indicating the medical necessity of the purchase. When applying for funds from a state vocational rehabilitation agency the applicant must demonstrate that the device or technology will assist in preparing for, getting, or keeping a job. If work is not an expected outcome, then the justification must indicate that the device will enhance the individual's ability to function independently. Often local school districts will pay for devices and auxiliary aids used in the classroom. Families requesting that the school pay for assistive technology should be prepared to demonstrate how the device will enhance the child's ability to obtain an appropriate education in the least restrictive environment. Other funding sources will have their own specific requirements. Success in securing funding is often dependent on the applicant's ability to address each agency's unique requirements in a funding request.

Eligibility

Different funding sources have different eligibility requirements. Below is a synopsis of the eligibility requirements of major funding sources for acquiring assistive technology.

Credit Financing

- Eligibility: Based upon applicant's credit history, collateral used to secure the loan, and other assurance of likelihood that loan will be repaid.

MEDICAID

Medicaid eligibility groups are:

- Recipients of Aid to Families with Dependent Children (AFDC);
- Supplemental Security Income (SSI) recipients (or in States using more restrictive

criteria--aged, blind, and disabled individuals who meet criteria which are more restrictive than those of the SSI program and which were in place in the State's approved Medicaid plan as of January 1, 1972).

- Infants born to Medicaid-eligible pregnant women. Medicaid eligibility must continue throughout the first year of life so long as the infant remains in the mother's household and she remains eligible, or would be eligible if she were still pregnant;
- Children under age 6 and pregnant women who meet the State's AFDC financial requirements or whose family income is at or below 133 percent of the Federal poverty level. (The minimum mandatory income level for pregnant women and infants in certain States may be higher than 133 percent, if as of certain dates the State had established a higher percentage for covering those groups.) States are required to extend Medicaid eligibility until age 19 to all children born after September 30, 1983 in families with incomes at or below the Federal poverty level. This phases in coverage, so that by the year 2002, all impoverished children under age 19 will be covered. Once eligibility is established, pregnant women remain eligible for Medicaid through the end of the calendar month ending 60 days after the end of the pregnancy regardless of any change in family income. States are not required to have a resource test for these poverty level related groups. However, any resource test imposed can be no more restrictive than that of the AFDC program for infants and children and the SSI program for pregnant women.
- Certain aged, blind, or disabled adults who have incomes above those requiring mandatory coverage, but below the Federal poverty level.

MEDICARE

Eligibility: Persons who are:

- 65 years of age or older
- Under 65, but disabled severely enough to qualify for Social Security Disability Insurance (SSDI) for at least 25 months. Equipment Payment Policies: Durable Medical Equipment (DME): equipment which (a) can withstand repeated use, and (b) is primarily and customarily used to serve a medical purpose, and (c) generally is not useful to a person in the absence of an illness or injury; and (d) is appropriate for use in the home.
- Internal prosthetic devices
- External braces
- Artificial limbs or eyes

TEFRA

- Eligibility: Provides coverage for children deemed diagnostically eligible (as established by SSI definition), but would be financially ineligible for SSI due to parent income. Children must meet medical necessity requirements for institutional care.

Private Insurance

- Health Insurance
- Disability Insurance
- Liability Insurance
- Self-Insured Employers' Insurance
- Eligibility: Persons recognized as beneficiaries/dependents under particular insurance policy.

Special Education

- Eligibility: Children with disabilities aged birth through 21.

Veterans Administration

- Eligibility: Veteran's service/financial status: Category A: Service connected veterans, or non-service connected, but with income below \$15,000 (single); \$18,000 (with dependent). Category B: Veterans not in Category A, but with annual income below \$20,000 (if single); \$25,000 (with dependent). Category C: All other veterans.

PASS

- Eligibility: An income and/or resource exclusion that allows a person who is blind or disabled to set aside income and/or resources for a work goal such as education, vocational training, starting a business.

IV. PROBLEM STATEMENT

Statement of Need

People with disabilities constitute a large percentage of the American population, including those who have difficulty using their hands and arms to control and manipulate their environment. In order to live their daily lives, they must be under constant supervision should a problem or difficulty occur. Often, this requirement is not met because such care is hard to find due to lack of assistance or for financial reasons. If a robotic manipulator could be designed and manufactured for these people, the need for constant supervision could be reduced drastically or even eliminated. The robotic arm could assist them in their daily activities, like picking things up from the ground or exploring their immediate environment. Furthermore, if such a device was simple to operate it could be used by younger patients. Also, a robotic arm would rebuild confidence and self-esteem lost in the depths of the illness. The aim of this project is to design a lightweight, easy to use, and low-cost robotic manipulator, which could act as an extension for the affected people to regain their independence from the disease.

Design Specifications

At the initial design meeting at University of Pennsylvania in October 1996, the members of each team from the five participating universities discussed the requirements for this year's Gateway project. This discussion was based on the following key elements assigned to the project's design configuration:

- Kinematic configuration (number of joints)
- Degrees of freedom
- Linkages
- Transmission
- Actuation from DC servomotors using 12 volt batteries

After a detailed analysis of the suggestions from each university, the following requirements pertain to the design of this project:

- The manipulator will be designed for use by a young child with limited hand and arm movement. (ages 6-12 years). The device will mount on the child's motorized wheelchair.
- The manipulator will lift an object with a maximum dimension of 7.5 centimeters (3.15 inches) and a weight of one kilogram (2.27 pounds). Also, the device is to lift the object from the ground at a minimum distance of one meter (39.3 inches) from its base.
- The speed of the manipulator in reaching its destination point will range from 0.15 meters per second to 0.20 meters per second (5.9 inches per second to 7.9 inches per second). The gripping force supplied by the device in lifting the an object cannot exceed one kilogram (2.27 pounds).
- The manipulator will operate with endpoint control based on a program written in C language on an IBM platform. The program will specify where the manipulator's endpoint is at a certain time and location in space.
- The cost of the manipulator, including parts, material and labor, excluding the computer control interface, is not to exceed four thousand dollars.

V. PHASE I

Using the design requirements as determined by the five student teams, it was decided that each university would use the first phase of the project for developing an independent conceptual design. In addition to their kinematic design considerations, each university was assigned a specific task to research to aid in the final design and manufacture of the manipulator.

The Cooper Union team was responsible for the design of the manipulator's user interface. This included the selection of the interface (i.e. joystick, keyboard, etc.) and the generation of the necessary coding information. The Drexel University team was assigned

the task of graphical simulation and analysis of the proposed design (i.e. kinematics, finite element analysis, etc.). The NJIT team was responsible for DC servomotor selection, sizing and sensing based on load requirements. The Ohio State team was assigned the task of preparing programming code for the computer interface and task level activities. Finally, the University of Pennsylvania team was responsible for design of the computer interface and real time control

The five teams used the chosen design requirements as the basis for their kinematic design criteria. The decision on the final design to serve as the blueprints for the manufacture of the manipulator in Phase II of the project was made at the design conference held at Cooper Union in January 1997.

A. Design of a Wheelchair-Mounted Robotic Manipulator (The Cooper Union)

Design Considerations

The system should also be robust, lightweight, modular, compact, easy-to-operate and safe. Aluminum was the material selected in virtue of its robust and lightweight properties, high strength to weight ratios and relatively low cost. In order for the robotic arm to serve as an extension of the user's arm, the proposed robotic arm must have a jointed-arm configuration. Regarding its articulation, the robotic arm should have 5 to 6 DOF. As to the physical size of the robotic arm, its length and cross-sectional area should be similar to that of a human arm. The robotic arm's workspace should be similar to the arm of an average man sitting on a wheelchair. The robotic arm's speed, accuracy, repeatability, and resolution are governed by its actuators and controllers.

Design I

Design I has a six degrees of freedom. The 1st degree of freedom provides a lateral motion in a range varying from 0 degrees to 180 degrees, the 2nd degree of freedom a vertical motion ranging from 0 to 180 degrees, the 3rd degree of freedom another lateral motion ranging from 0 to 180 degrees, the 4th degree of freedom another lateral motion ranging from 0 to 180, the 5th degree of freedom provides a rolling motion about the wrist axis, and the 6th degree of freedom handles the opening and closing of the gripping claw.

Design I has an extended length of 112 cm, and a 2:2:1 lengthwise member ratio. Figures 8-13 depict the dimensions of the link members and gripper of the robot system, and Figure 22 shows a prototypical illustration of the system. An assembly rendering of Design I is given in Figure 25.

Member No 1

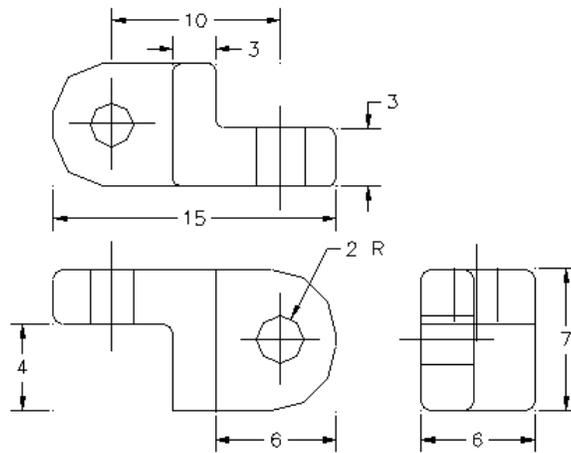


Figure 8: Link member 1 of Design I

Member No 2

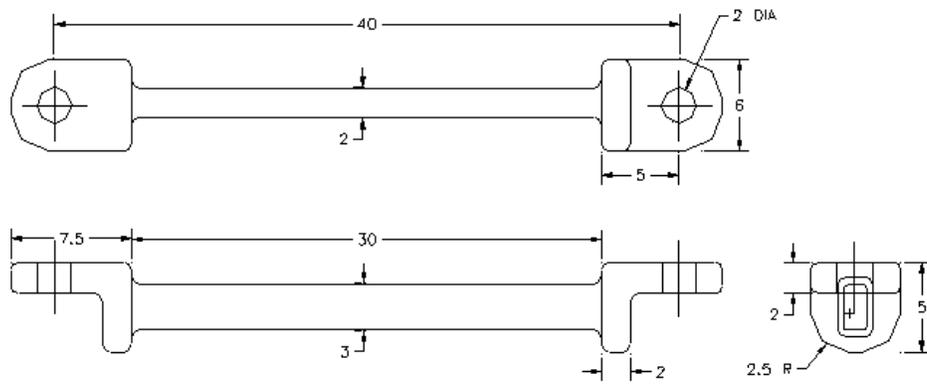


Figure 9: Link member 2 of Design I

Member No 3

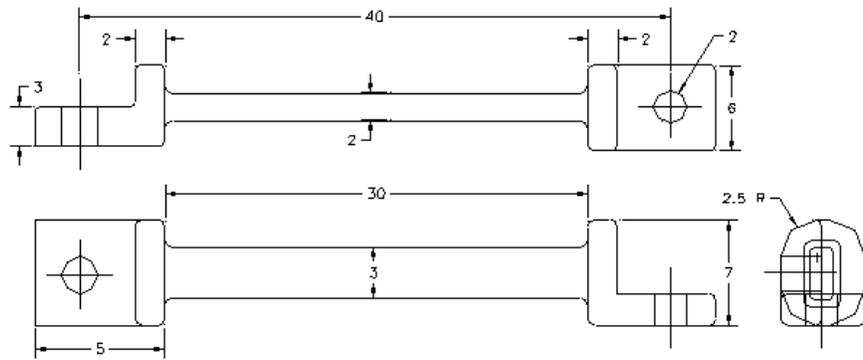


Figure 10: Link member 3 of Design I

Member No 4

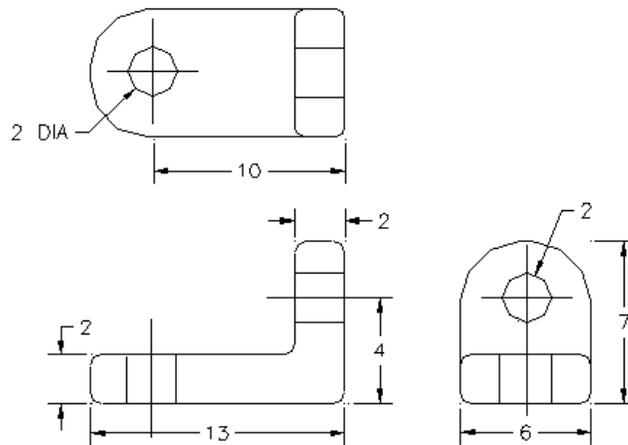


Figure 11: Link member 4 of Design I

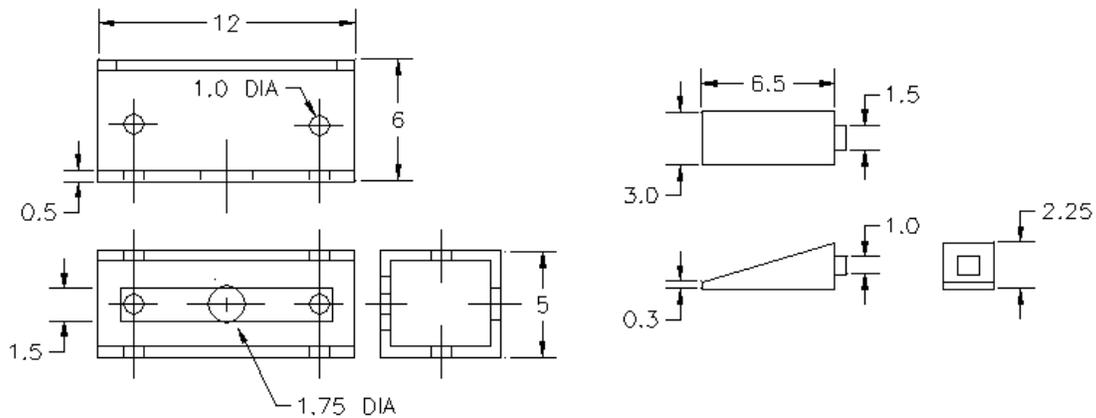


Figure 12: The gripper assembly of Design I

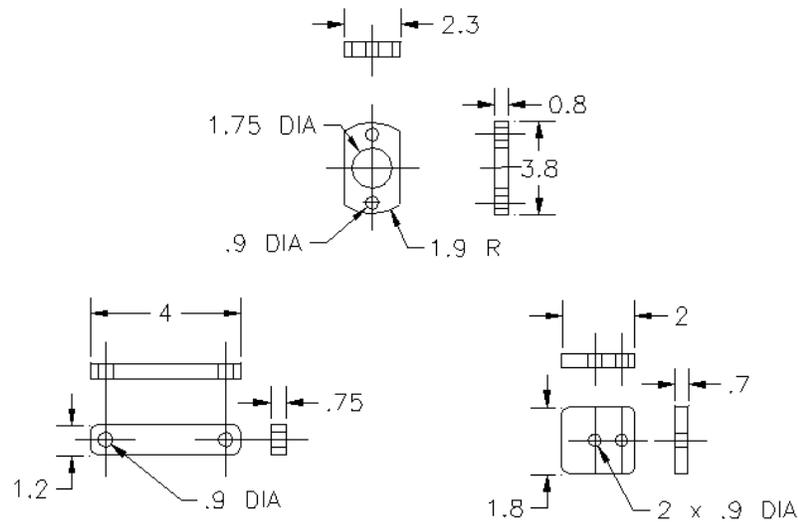


Figure 13: Gripper accessories of Design I

Design II

Design II has a total of 5 degrees of freedom. The first degree of freedom provides lateral motion ranging from 0 to 180 degrees, the second degree of freedom provides a vertical motion ranging from 0 to 180 degrees, the 3rd degree of freedom provides for another lateral motion ranging from 0 to 180 degrees, the fourth degree of freedom provides the rolling motion about the wrist, and the fifth degree of freedom accounts for the opening

and closing of the gripper.

Design II, whose total length when fully extended is 112 cm, has a 1:1 lengthwise member ratio. Figures 14-18 depict the dimensions of the link members and gripper of the robot system, and Figure 23 shows a prototypical illustration of the system. An assembly rendering of Design II is given in Figure 26.

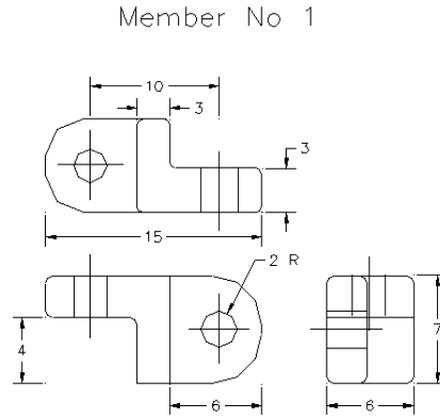


Figure 14: Member 1 of Design II

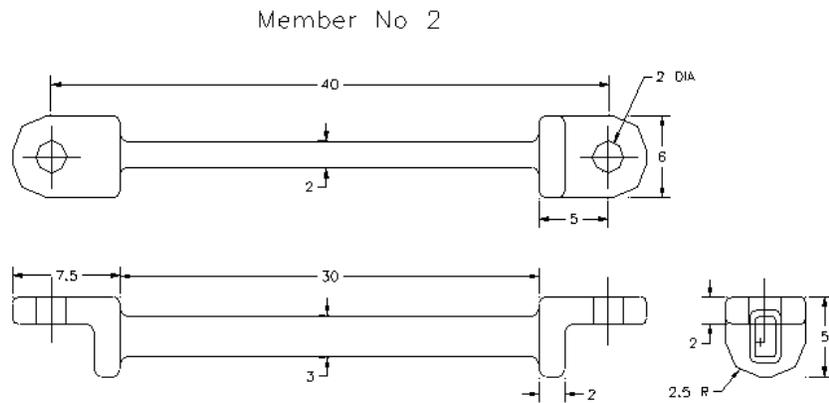


Figure 15: Member 2 of Design II

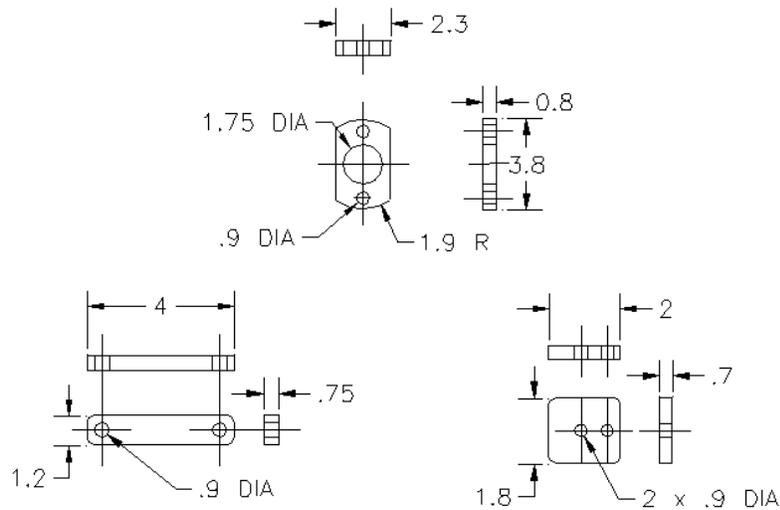


Figure 18: Gripper accessories of Design II

Design III

Design III consists of three link members. The first member rotates vertically; the second member and the third members rotates laterally.

When the total length of the arms is 1.0 meter, the first member is 0.40 meter long, and the length of the second and third members is each 0.30 meter. The grip is 5 cm long, which will make the total length little longer than 1.0 meter. Though it is specified that the maximum reach distance is 1 meter, in order to make the robotic arm more reliable and more efficient, the 5 centimeters is necessary to the design.

The general cross section shown in Figure 19 has three sides. Because the arm is only designed to pick up object with a weight of one kilogram and aluminum doesn't contribute too much weight and moment to the arm, it is not necessary to use solid aluminum, and it gives room to the installation of wires, belt and other accessory. The thickness of the aluminum plate is 0.5 centimeter.

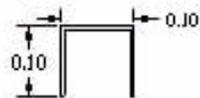


Figure 19: Cross Section of middle part of arm member

Though the arm member structure is strong enough to perform the specified task, the arm needs to be strengthened at both ends. Any sudden strong force may cause the aluminum arm bent at the joint area, thus cause the joint to malfunction. The solution to prevent the

bending is to add another side to the arm. The cross section at the end part of the arm member is shown in Figure 20. It decreases the bending moment at the two edge in Figure 26 dramatically.

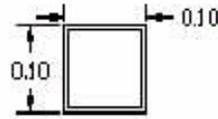


Figure 20: Cross Section of end part of arm member

The back view of the arm member is shown in Figure 21. As it can be seen that the 3 cm wide aluminum bar fixed the distance between two sides of the arm member.

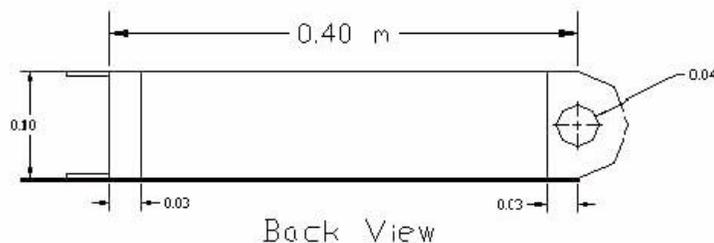


Figure 21: Back view of arm member 1

The bar is not only designed to strengthen the joints, it is also designed to limit the movement of each arm member. It is controlled by the distance between the edge of the arm member and the edge of the bar. In this project, there are two things that need to be considered which may limit the movement of the robotic arm. First, it is very important that the arm can not only reach most of the region around the user, but also limited in moving too freely, which may cause inconvenience to the user. Second, the safety of the user is also a serious consideration that the arm should not reach further than the point where the robotic arm can barely touch the patient's face. Under these considerations, a thorough experiment needs to be done to determine the location of the bars to limit the arm movement. The angle limitation can range from 0 degree to 135 degree depend on the location. But bars only limit rotational movement on its side, on the other side same result can be achieved by cut that side of aluminum to create a gap allowing movements.

The initial design is that the joint between the wheelchair has 90 degrees of movement up and 120 degrees of movement down; the joint between the first and the second members has 135 degrees movement left and 135 degrees movement right; the joint between the second and the third members has 90 degrees left and right. After testing the design using simulation, more information will be collected to reach a more reliable design.

The arm member design is different between member one and member two. Member two and member three have same design. Member one is longer and having connections with other members on different sides of the member, since, at one end, it rotates vertically and

at the other end it rotates horizontally. Thought the basic design is same in member two and three, member two have a small height so it will be able to fit in the ends of member one and three. The height of member two is 9 cm, which is 1 cm shorter than member one and member three due to the thickness of the aluminum plate.

With a cross sectional area of 0.01 meters squared and aluminum bar thickness of 0.5 cm, it is a strong robotic arm for home use. Five degrees of freedom may not be enough to reach any point the users like to reach, but with the extra freedom that the wheelchair itself can provide, this design will cover all regions around the users. Another concern is that the direction of the grip may not be at the right direction for the best results, but less degrees of freedom are easier to control. There has to be some compromise between the user and the design.

Kinematics

The Robix™ RC-6 Robotic Construction System is used to validate the kinematics and operation of each conceptual design. Robix RC-6 has six servomotors, and a Robix adapter - an electronics package with power attachable via to either LPT1 or a 2 parallel printer port on a PC running DOS. It can drive up to 6 servos simultaneously. This particular construction kit also comes with 16 punched and machined aluminum links, a parallel-jaw gripper with wrist subassembly, a weighted base and pivot post, and a manipulator breadboard.

It has also been determined that the maximum number of degrees of freedom of the design should be six. Although the additional degree of freedom provided by Design I may enhancing the robotic arm's reaching ability, it may hinder the user's ability to effectively control the mechanism. Figure 22-24 depict the Robix renditions of the three conceptual designs than five degrees of freedom would restrict the robotic arm's functionality and flexibility.

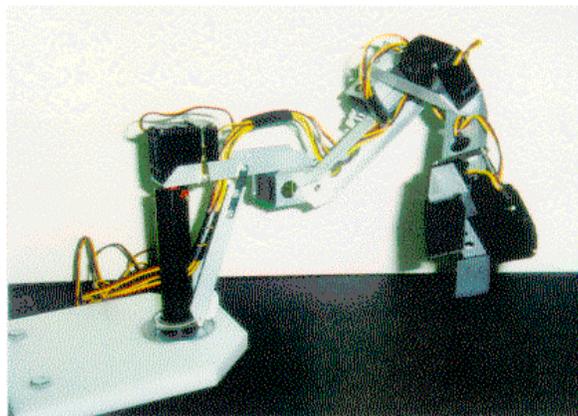


Figure 22: A Robix rendition of Design I

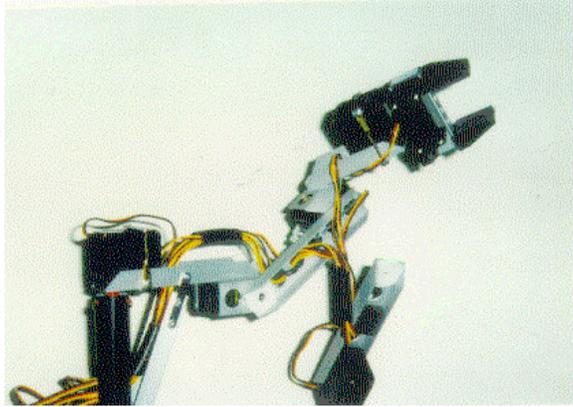


Figure 23: A Roblox rendition of Design II

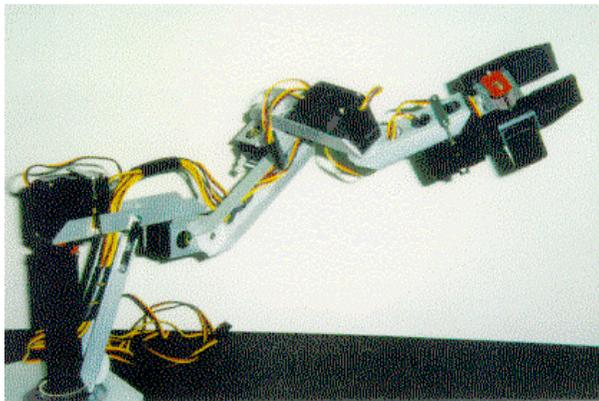


Figure 24: A Roblox rendition of Design III

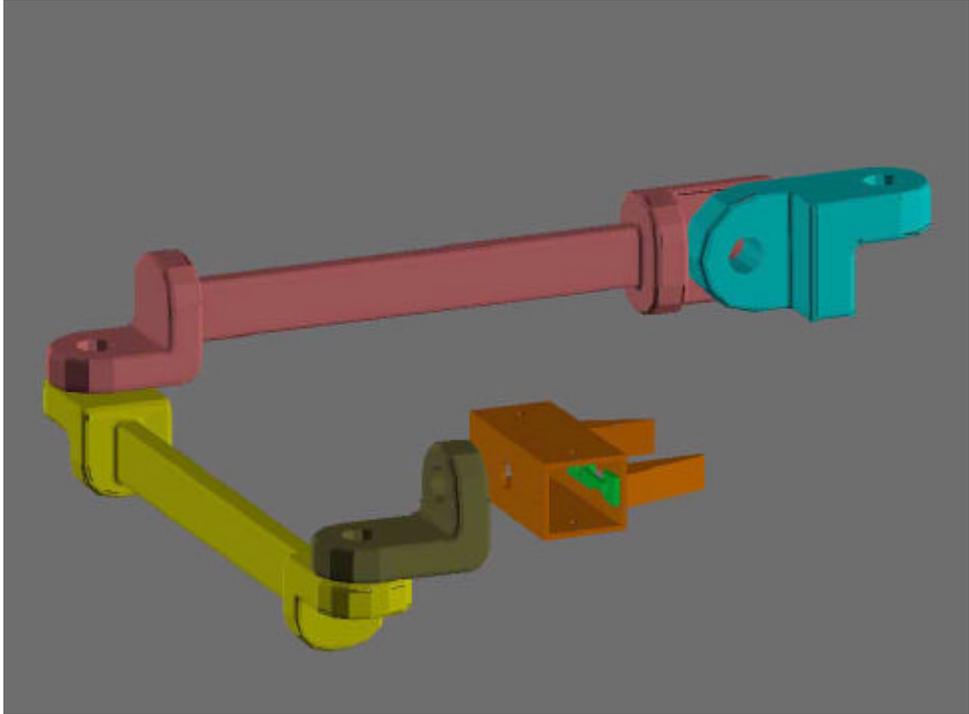


Figure 25: Rendered Image of Design I

Figure26: Rendered Image of Design II

Figure 27: Rendered Image of Design III

Conclusion

Having validated all three designs with the Robix construction system, it is determined that Design I and Design II satisfy all the design requirements.

Design I has a solid cross-sectional area throughout its length. The relatively small and compact cross-sectional area (approximately 0.0006 m²) enhances the device's functionality, aesthetics, robustness and the possibility of being easily stored when not in use. It is best suited for implementation using solely servomotors as actuators.

Design III is best suited for actuating systems made up of servomotors, chains and belts. The hollow cavities throughout its members provide the required space, shielding, and

support to the installation of such belt-driven actuating systems.

B. Five Degree Electromechanical Robotic Manipulator (Drexel University)

Design Considerations

The primary objective in designing the wheelchair-mounted robotic manipulator is to assist children who are unable to use their hands and arms in normal daily activities. Along with the requirements determined at the University of Pennsylvania design meeting, the Drexel team also proposed additional considerations in order for them to produce an optimal design of the manipulator.

User Requirements

Intended User: Since the manipulator will be designed for a young child, it must be simple to operate from a non-sophisticated user interface such as a joystick or keyboard.

Aesthetics: Since the device will be mounted to a child's motorized wheelchair, it will operate quietly and be non-obtrusive to the child's surroundings. The device should not intimidate or scare the child as its designed operation will act as an extension of the child's body. Also, since a young child's attention span is sometimes limited, the manipulator will be inviting to the user by looking desirable in a young child's interests.

Simplicity and Safety: Because of the manipulator's intended use, the design must be simple to operate on a child's intelligence level, and therefore, very little training will be required to use it. Complex movements in the device will be minimized as to prevent the threat of injury to the user. The child will have complete control over the manipulator to establish maximum efficiency and effectiveness in using the device.

Functional Requirements

Weight: The weight of the manipulator will be minimized to provide quiet and smooth operation. A lightweight design is preferable to keep the internal mechanisms simple and cost-efficient.

Degrees of Freedom: The degree of freedom of the manipulator will be minimized to ensure simple controllability of the device from the user-interface platform. Also, it will be cost-effective since highly complex movements will be eliminated.

Care and Maintenance: The manipulator must be easy to maintain, clean, and store during idle periods. Most importantly, it must provide the user with comfortable and

lasting operation.

Design of the Five Degree Electromechanical Robotic Manipulator

In accordance with both the user and functionality requirements, the Drexel team will design a five degree of freedom electromechanical robotic manipulator. The chosen configuration of the manipulator is based on the following studies on existing robotic manipulators on the market:

- Analysis of the Helping Hand shows that the manipulator is noisy, due to the size of the electric motors and its operation is unstable due to the excessive weight added to the arm resulting from the placement of the motors. The Helping Hand uses four degrees of freedom and costs approximately \$10,000 to manufacture.
- Analysis of the Manus demonstrates a very powerful and capable robotic manipulator, but the price of the arm is well beyond the scope of this project as it cost \$65,000 to manufacture. Also, the increased degrees of freedom (7) complicates the programming required to operate it.
- Both manipulators add excessive width to the wheelchair, making it difficult for the chair to pass through narrow doorways.

Past designs of these arms have not been accepted by consumers, creating a need to review their design. First, in reaching an optimal design, the manipulator must be able to easily pass through a doorway structure. Both the Helping Hand and Manus arms add excessive width to the wheelchair making it difficult to pass through narrow doorways, while the Drexel arm only adds 4 inches to the width of the wheelchair as shown in Figure 28.

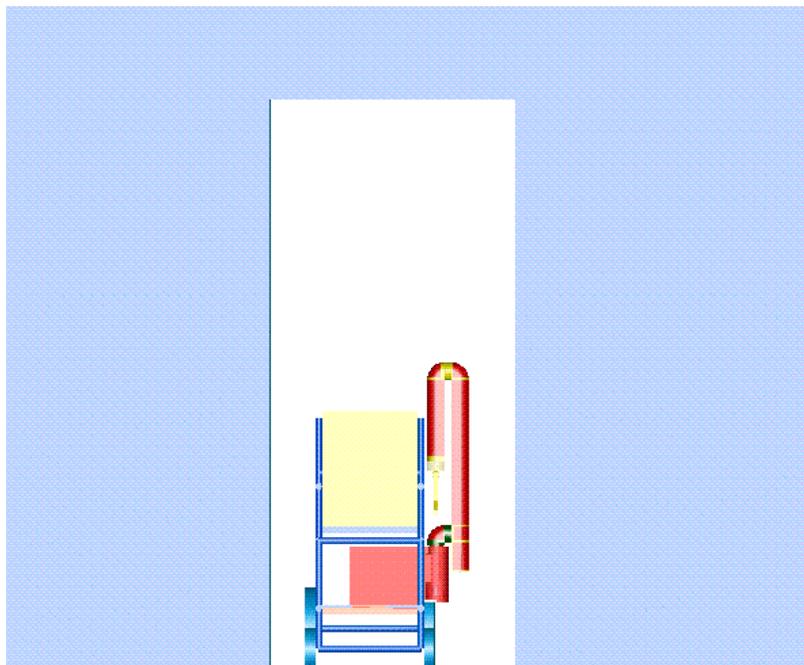


Figure 28: Door way view

Programming the manipulator to perform its required tasks can become quite cumbersome, especially if there are a high number of degrees of freedom governing the design. The Helping Hand operates with four degrees of freedom (DOF) which minimize programming. However, the state-of-the-art Manus uses seven DOF, making the programming quite difficult. Also, having seven DOF poses a problem for the user because there would be more knowledge required for operation. The new design proposed by Drexel will use five DOF to move the manipulator. The arm was designed to satisfy the required one meter reach as shown in Figure 29.

Figure 29: Arm reach

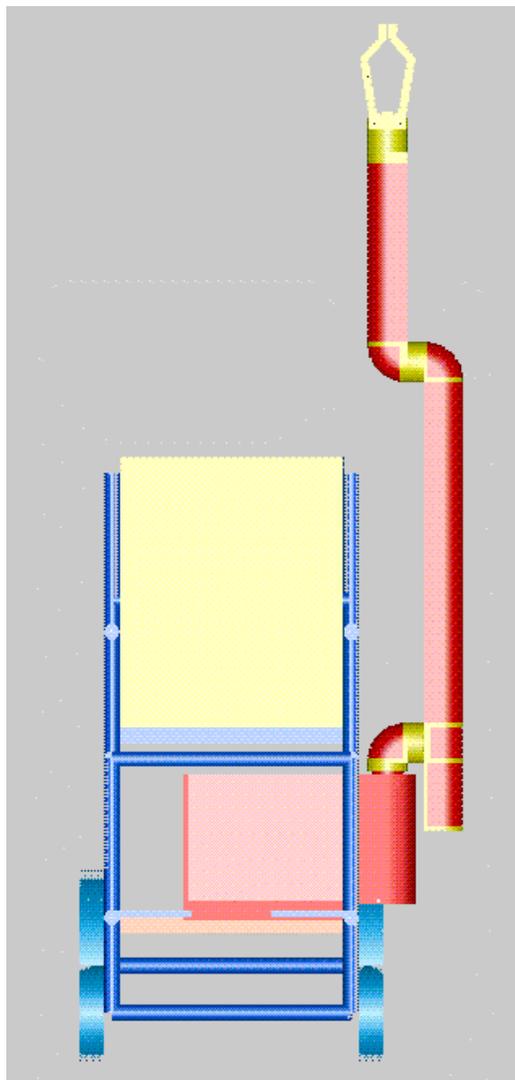


Figure 30: Arm description

To meet the safety requirements, clutches will be installed on motor drives and the rate of motion of the arm will be limited. Cost will be kept down by using "design for manufacturing" principles and by building the design around standard hardware. Ease of repair will be designed into the arm through the use of access panels. The arm will be made quiet by using belts and cables for motion drivers rather than gears. The motors of the arm will be mounted in the base to improve the motion of the arm and to keep a low center of gravity.

Base

The base is a key point in the design of the robotic arm. It is a unique design that the Manus, Helping Hand or other manipulators do not have. A study of several mechanical children's wheelchairs, found that each had three to four inches of space beneath the seat in front of the battery as seen on the next page in Figure 31.



Figure 31: Standard children's wheelchair

The robotic arm takes advantage of this by placing the base of the arm in this location, holding all the heavy motors. A mounting bracket containing a track will be fastened to the bottom of the wheelchair, as seen in Figure 32. The base can then slide on and off this track for easy removal as shown in Figure 33. Studies show that people with disabilities need the arm to be removed for storage, travel, or simply when they do not want to use it. This design allows for a simple removal in minutes by a helper. The base also has a removable front panel that allows easy access to all motors, clutches, cables, and wires for easy maintenance and repair.

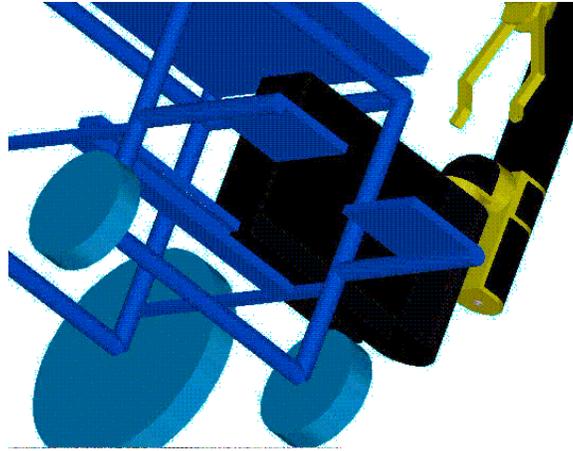


Figure 32: Underside view of mounting

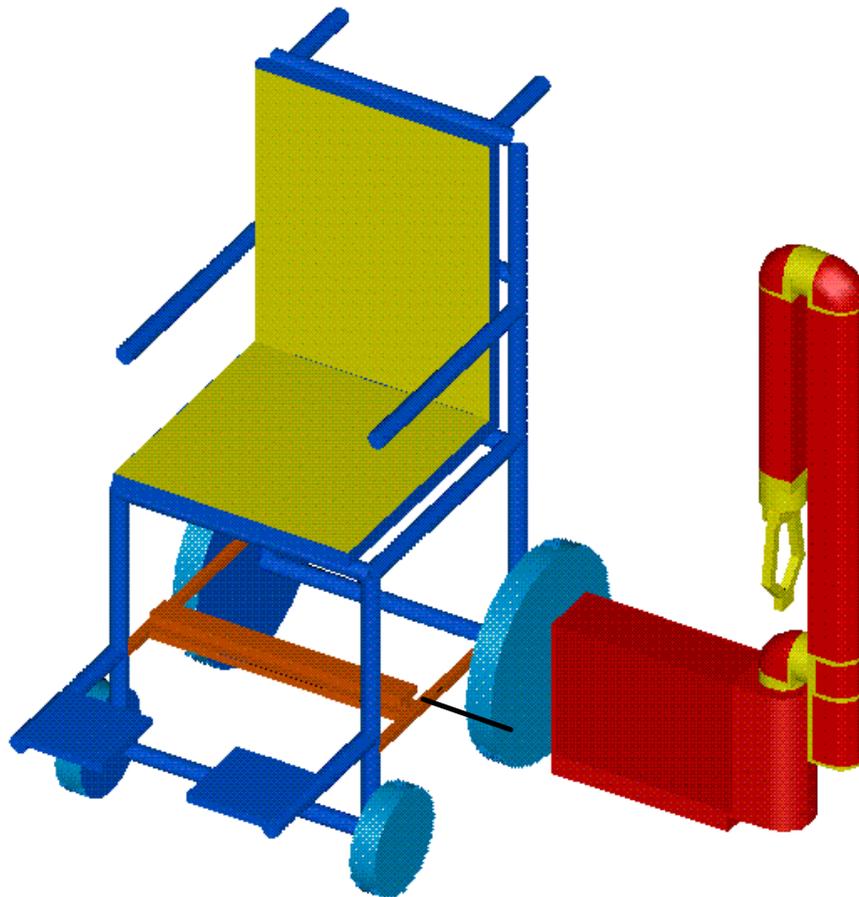


Figure 33: Arm installation

Wheelchairs with mounted arms such as the Manus, mounted to the side of the wheelchair, tend to track to that side due to the excessive weight of the arm. Drexel's base design allows for most of the weight, specifically the motors, to be placed beneath

the wheelchair dispersed over the width of the chair. The arm itself is therefore very light and will cause much less interference, with the regular operation of the chair. Similarly, the arm will operate smoother, with no jerky movement from the weight of the arm itself. Larger, heavier motors can be used since they will be mounted in the base, instead of the arm. These motors produce much less noise and aid in the smooth operation of the arm.

The motors have a simple three-bolt bracket mounting within the base. Each motor contains a clutch, causing the motor to slip when it comes in contact with another object, such as the floor or a table. This also makes the arm safe for the child to use, with no danger of the arm contacting the child or breaking surrounding objects.

Three motors are located in the main portion of the base as shown in Figure 34. One controls the gripper, one the wrist and one the rotation of the shoulder. A fourth motor is placed in a hollow cylinder at the end of the base where the arm assembly is connected. This motor controls the shoulder movement about the x-axis. The cylinder runs the entire height of the base as shown in the exploded view in Figure 35. A smooth rotation of the cylinder and arm assembly is provided by a small diameter thrust bearing at the bottom of the cylinder and a large diameter bearing at the top of the cylinder. Also at the bottom of the cylinder is a small sprocket, which connects to the closest motor by means of a toothed belt to apply rotation of the shoulder. This provides the cylinder and the entire arm from the shoulder out a range of 120 degrees of rotation. The cylinder contains an opening in the side to allow a path for the cables going from the motors to the rest of the arm. In the bottom of the base just beneath the cylinder there is another access panel through which the cylinder is inserted into the base. The base has also been designed to allow the PC board to be attached to the back of the base. A cover can be added to protect the board.

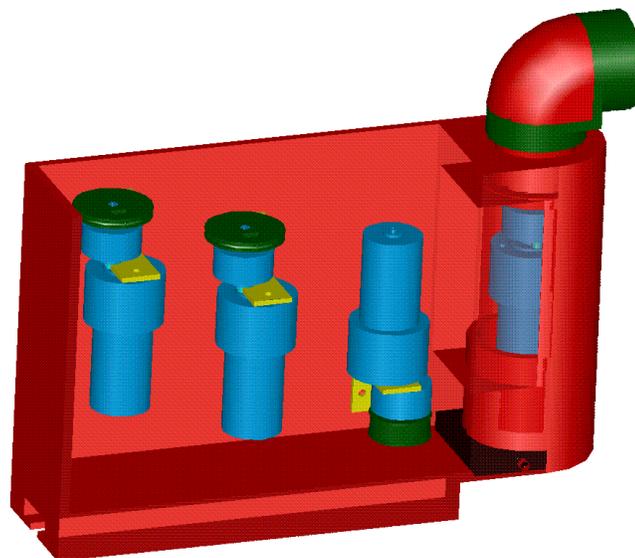


Figure 34: Base assembly

Shoulder

The shoulder assembly consists of a L-shaped rounded bracket. All the brackets are made of Aluminum 6061-T6. One side of the bracket is attached to the top of the cylinder coming out of the base. The other side of this bracket mates to a small cylinder, which is attached to a bracket on the upper arm. There are two bearings between the brackets, which ensure smooth rotation. The flanges on the brackets have large holes to allow the cables to pass through to reach the rest of the arm. The cylinder from the upper arm bracket has a larger inner diameter miter gear on it. This gear mates with a similar gear which is attached by a rod to a clutch and then to the large motor in the bottom of the cylinder. The shoulder has a range of 135 degrees of rotation. To enclose the L-bracket, a cover will be applied to hide the cables, provide easy access, and to be aesthetically pleasing.

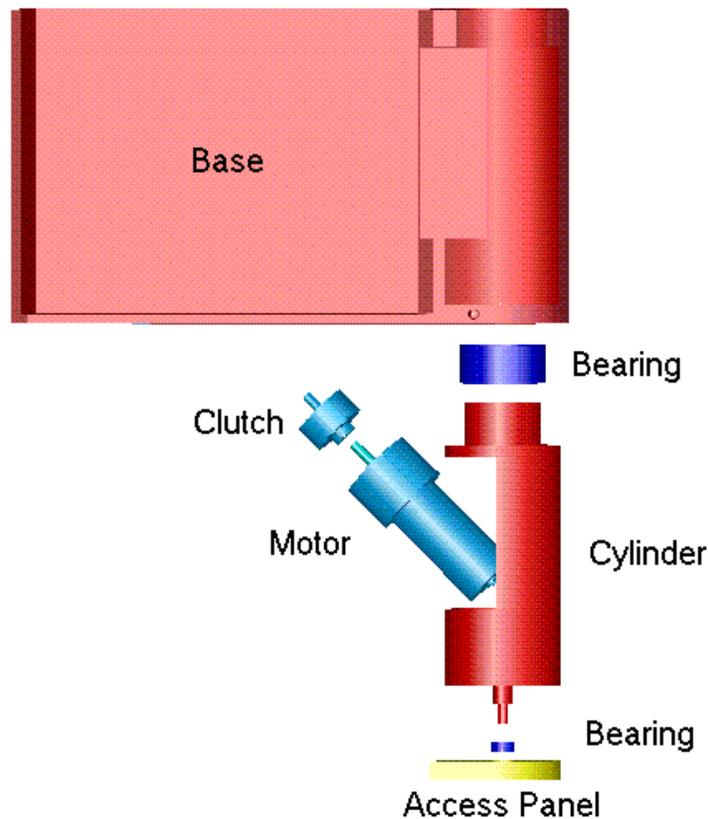


Figure 35: Base detail

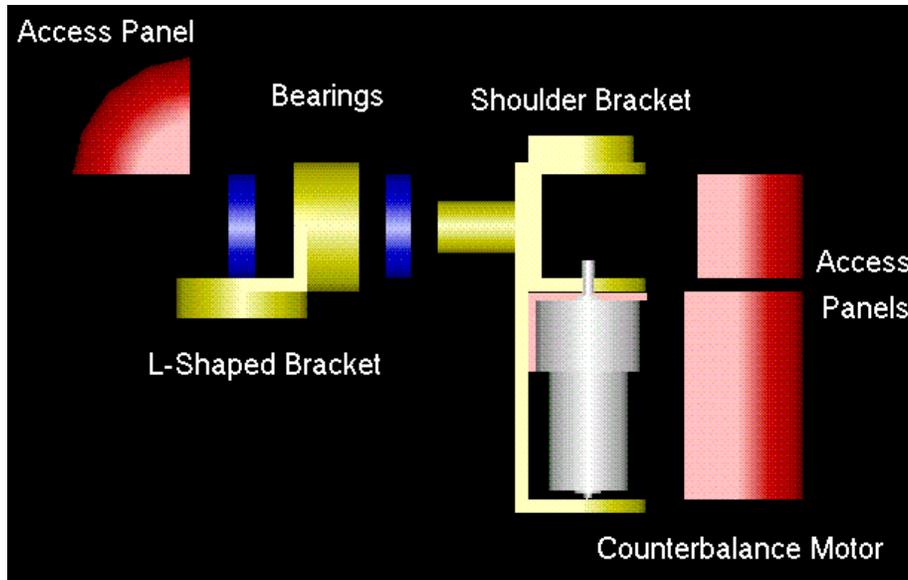


Figure 36: Shoulder detail

The piece of the arm referred to as the upper arm is 22 inches in length. The main portion of the upper arm is composed of ABS plastic. It is in the shape of a half circle on one side and two right angles at the other. This was chosen to allow extra space for mounting of gears and other mechanisms. The cross section dimensions are shown in Figure 37. These pieces are fitted on the joint and are fastened by screws around the perimeter.

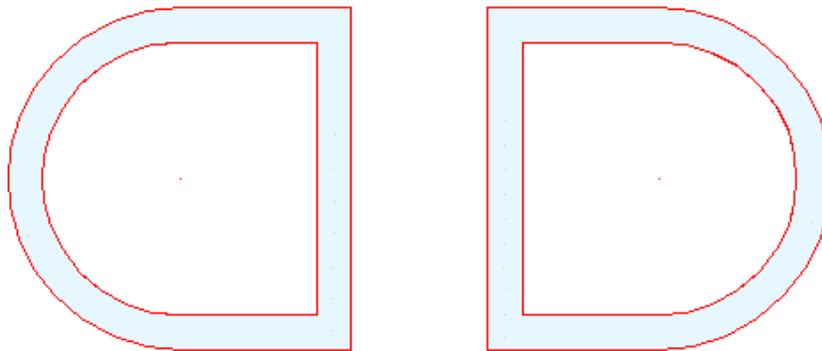


Figure 37: Cross-section of arms

At the shoulder end is the bracket that mates to the L-shaped shoulder bracket. Past that bracket the upper arm extends approximately 4 inches to house a fifth 2-function motor. This motor's primary function is to provide the motion for the elbow. The other function of this motor is to apply a counterbalancing weight for the entire arm. The housing for the counterbalancing motor has an access panel for the installation and removal of the motor.

At the elbow end there is an L-shaped bracket identical to the one used at the shoulder. This bracket mates with a similar L-shaped bracket that has a pin for rotation. This bracket is attached to the forearm. The set-up of the elbow is similar to the shoulder.

Two bearings provide smooth rotation of 360 degrees. Access panels are located on both brackets to conceal the shielded cable running out to the wrist and gripper. These details can be seen in Figure 38.

Wrist

The gripper is attached to the arm by a wrist as shown in Figure 39. The wrist rotates to aid in picking up objects. The rotation is controlled by a motor in the base via a shielded cable. The motor spins either clockwise or counterclockwise pulling the cable. The cables themselves are attached to a pair of mitered gears in the wrist to change the direction. This produces a full 180 degree rotation of the wrist and gripper.

The gripper has the difficult function of imitating the human hand. However, a simple, two finger, open and close gripper can perform many tasks, specifically picking up an object. Drexel has designed the length of the fingers to be the length of an average human finger and palm, about 6 inches. They contain rubber pads at the tips to reduce slipping and they are bent to the shape of a partially closed hand to better grip an object.

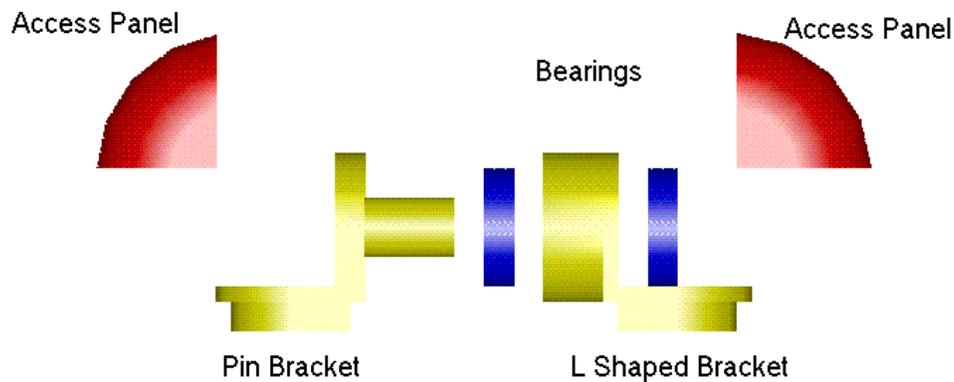


Figure 38: Elbow detail

Each finger is connected to the slide block by a link as shown in Figure 39. The slide block is connected to a motor in the base by a shielded cable fed through the arm. When the cable is pulled, the slide block slides toward the wrist, pulling the links and thus closing the gripper. When released, a spring returns the fingers to their original open position. A clutch on the motor keeps the gripper from breaking an object or hurting a person, but allows for the required one kilogram gripping force.

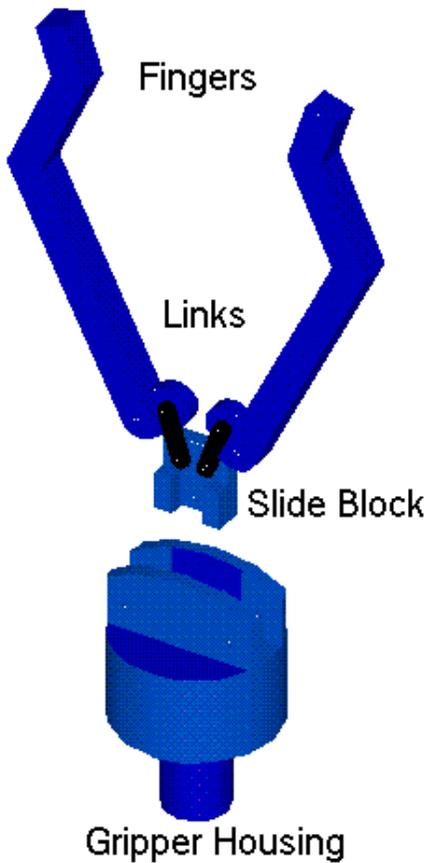


Figure 39: Gripper detail

Alternate Mounting

The optimal mounting for the arm as shown before is beneath the seat toward the front of the chair. The mounting places the majority of the arm's weight in the center of the chair, allowing the chair to steer normally. An alternate mounting for cases when space does not exist below the chair would be outside the wheels of the wheelchair with the arm extending from the rear of the box. While the alternate mounting does not effect the performance of the arm itself, it may effect the operation of the chair. Studies show that heavy arms mounted to the side of a chair tend to pull the chair to that side during normal driving.

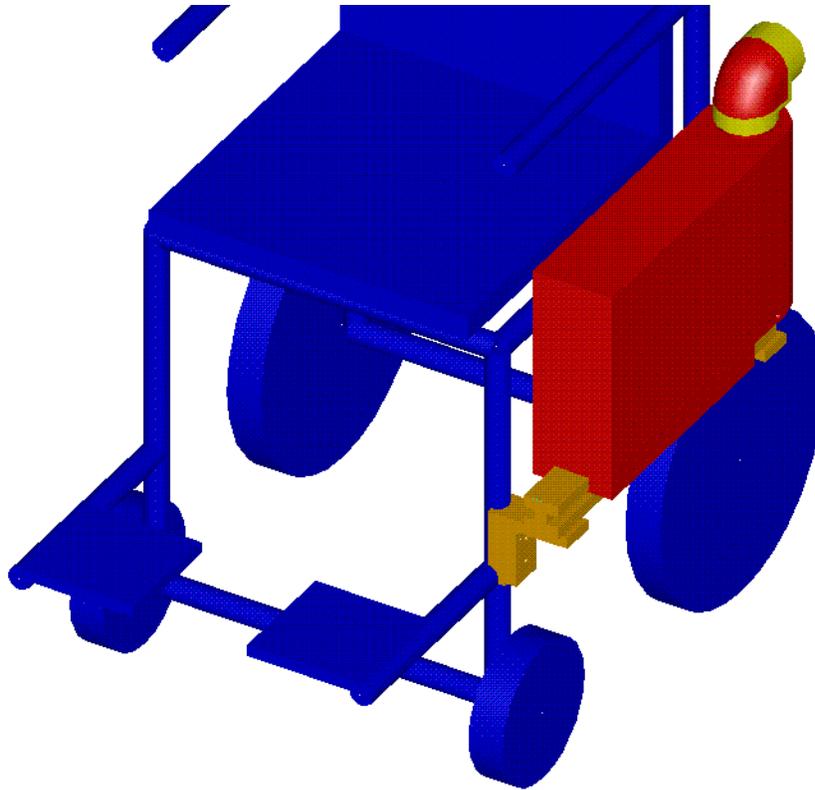


Figure 40: Alternate mounting

Controller Interface

The robotic arm is best controlled by its endpoint. This requires more programming, but makes it much easier for a child to control. Instead of moving each part of the arm individually, the user simply guides the end of the arm where they want it to go. Since the arm moves in a 3-dimensional area, Drexel has chosen a 3D interactive controller for use with the arm. Gaming companies, such as Sega and Nintendo make such controllers for use with their games. However, the world leader in RealLife 3D interactive motion control input devices and software is the Spacetec IMC Corporation. They make the SpaceOrb 360, the first true 3D controller. Simultaneous three-degrees, four-degrees, and six-degrees of freedom happen on the fly at brainspeed, all through light fingertip control. The result is natural, intuitive, smooth, and continuous 3D motion. Where joysticks restrict movement to one direction at a time (usually front, back, side-to-side), the SpaceOrb 360 provides the freedom to move in any and every direction, just as one would move in the real world. By simply pushing, pulling, and twisting the PowerSensor, the user can perform moves impossible with any other control devices, including the mouse and keyboard. The digital PowerSensor senses both the direction and amount of pressure applied to it, allowing the user to micro-adjust their speed and acceleration. It can be customized for use on DOS, Windows, and Macintosh personal computers. This model has six buttons, since it is a gaming controller, however, only one would be necessary for Drexel's design. It would be used for opening and closing the gripper. The controller will be mounted to the wheelchair, either on the left or right side, depending on the child. It

easily plugs into a port on the PC board and could be unplugged, allowing the arm to be removed.

C. A Power Wheelchair Mounted Robotic Arm (New Jersey Institute of Technology)

Design Objective

The primary objective is to design a Rehabilitation Robotic Arm (R2) mounted on a wheelchair to assist people who are unable to use their hands and arms in normal daily activities.

User Requirements

- **Intended User:** The arm will be designed for people with disabilities. The operation will be simple and easy for a non-sophisticated user to interface with such as joystick or keyboard.
- **Aesthetics:** The device will be mounted on an electric-wheelchair. The arm will be housed in an enclosure to make it as unobtrusive as possible. The enclosure that will be positioned in the side of the wheelchair will also make the arm aesthetically appealing to use. When not in use the arm will practically not be visible. Further, the robotic arm will be compact and not add in any significant way to the overall dimensions of the wheelchair. This is particularly important since the wheelchair is designed typically to pass through doorways and aisles.
- **Safety:** The arm must be used very easily and children should be able to operate it without much effort. The training required to use the arm effectively will be kept to a minimum. Complex movements are avoided to prevent any physical and mental discomfort to the user . The user will also have full control of the arm in using the device.
- **Cost:** The main objective is to provide a useful device at minimum cost. The device must be affordable for the average user.

Functional Requirements

- **Weight:** The weight of the arm must be kept to a minimum to ensure smooth and quiet operation. Aluminum is the material of choice due to its light weight, mechanical properties, and cost effectiveness.
- **Degrees of Freedom:** The robotic arm will have six degrees of freedom to provide desired maneuverability for picking and placing objects. Smooth and quiet operation is also a factor in determining the degrees of freedom.
- **System control:** The arm will move with endpoint control based on a program written in C programming language.
- **Object Size, Weight and Location:** The arm is designed to lift an object having a maximum length of 7.5cm (3.15 inches) and having a weight of 1 Kilogram (2.27

pounds). The arm must be able to reach out to an object up to a distance of 1m (39.4 inches) from its base.

- **Speed and Gripping Force:** The speed of the arm in reaching the destination point will range from 0.2 to 0.25 meters per second (7.9 inches per second to 9.8 inches per second). The gripping force is 5.8 KgF (13.2 lbF).
- **Care and Maintenance:** The Robot arm must be simple and easy to maintain, clean, and store when not in use. Most importantly, it must provide the user with reliable and robust operation.

Workspace Design Data

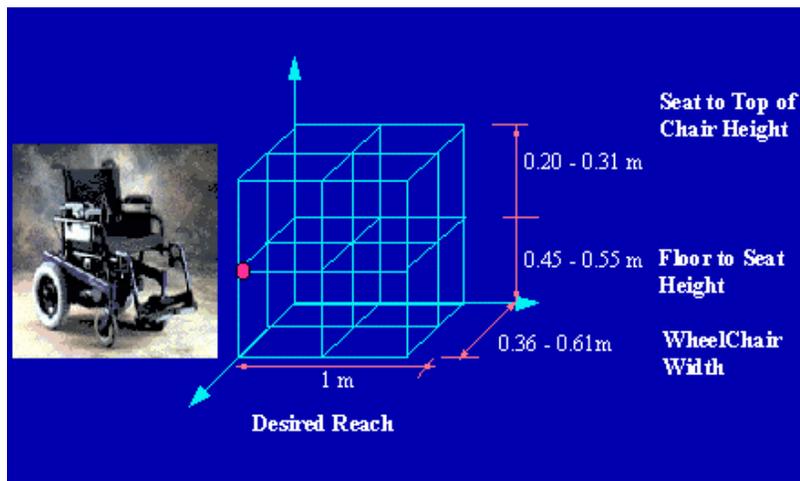


Figure 41: The dimensions of power wheelchair for children and the robot arm reach length requirements

(Source: The powered wheel chair dimensions are obtained from Invacare Corp. products)

Assuming the user is right handed, the control device is located at the resting position of the right hand. The reference position of the robot arm is accordingly under this position on the right hand side of the wheel chair. The side mounted robot arm is particularly desirable by users rather than the front mounted arms which are very obtrusive. Given that it is best if the robot arm is out of the user view, the arm will be located approximately at .55 m height from the floor.

Robot Arm Design

The design must be such that it does not imbalance the chair due to locating heavy weight on one side of the chair. Locating the arm as close as possible to the floor provides a more balanced chair than if it were located at the top of the chair especially above the seat height. This also makes it possible to have a shorter arm with less weight and less power requirement.

Locating the robot arm at the mid-point between the user's table and floor can provide the minimum arm length required. The arm then must reach 1 m from the floor. The item picked must be brought to the user, possibly to a table in front of the user.

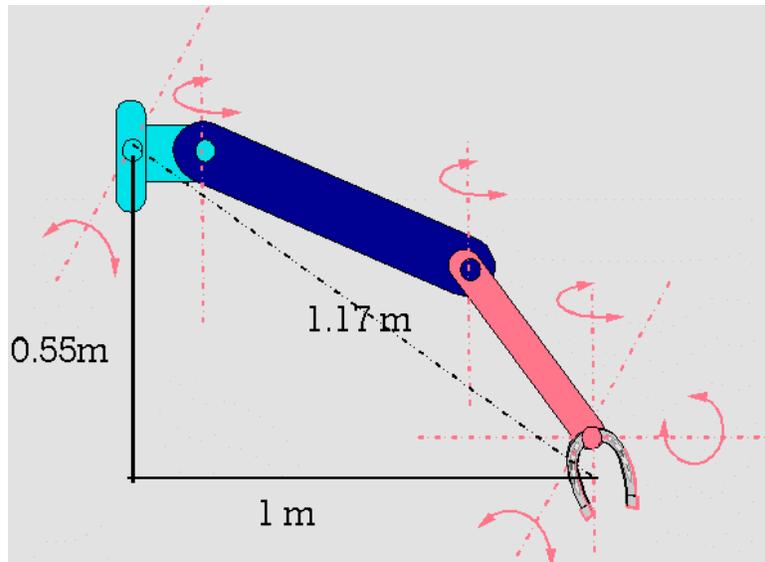


Figure 42: Robot arm length calculations

Given that the arm length reach is a maximum of 1 m and the height of the chair is approximately .55m, the total arm length can be shown to be 1.17m. Having an arm of length 1.17 m requires careful positioning of the arm when it is not in use. Positioning the arm vertically on the side of the chair was an option but was ruled out since it is likely to upset the balance of the chair as well as make the arm very obtrusive. The design team hence decided to make the arm foldable and locate it on the side of the wheelchair.

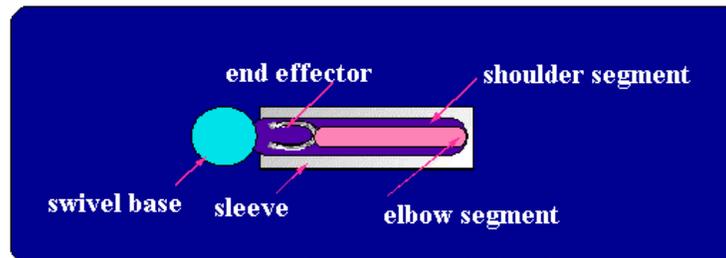


Figure 43: Folded robot arm design

The two segments of the robot arm, along with the base and the end effector, or gripper, define the shoulder the elbow and the wrist joints of the arm. The shoulder and the elbow segments are roughly equal in length to facilitate the folding of the arm. The sleeve serves as a casing for the arm. The arm moves on a track on the side of the wheelchair so that it can be withdrawn for use and retracted for storage while not in use.

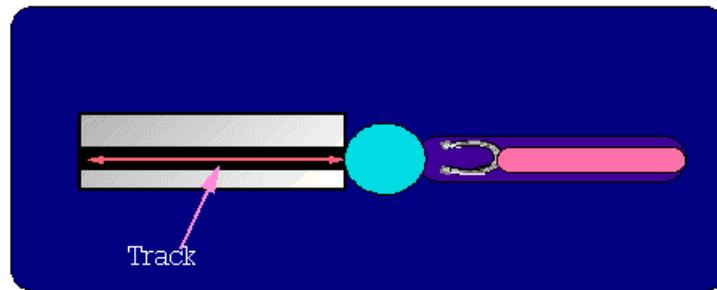


Figure 44: Retractable robot arm

The length of the sleeve must equal the depth of the chair so that the chair can be operated safely and unhindered by the sleeve. Another constraint is the width of the chair being well within the dimensions of doorways and aisles. Hence, the width of the chair including the sleeve must not be larger than standard minimum door width which is assumed to be 30 inches . The width of the wheelchair can vary between .36-.61m which leaves the sleeve approximately .12 meters in diameter in the worst case of being paired with the largest wheel chair. Since the wheelchair is for children, it is likely that it will be less than .61 m leaving more room for the sleeve. To provide more balance to the wheelchair, the design team decided to place the motors below the seat and close to the center of gravity of the wheelchair.

Components of the Robot Arm

The robot arm has the following components: swivel base, shoulder segment, elbow segment, end effector, and the sleeve.

- 1. Swivel Base:** This is the base of the arm and is fixed to the side of the seat of the wheelchair. It moves on tracks within the sleeve to retract and withdraw the folded arm. The base also has a rotary motion and can rotate in both clock-wise and counter-clockwise directions.
- 2. Shoulder segment:** The shoulder segment is connected to the swivel base. The shoulder can move vertically. It can also move laterally when the base rotates.
- 3. Elbow Segment:** The elbow segment is connected to the shoulder. The elbow segment can move laterally. The elbow segment when not in use moves 180 degrees and rests on top of the shoulder in the folded position of the arm.
- 4. End Effector:** The end effector, or gripper, plays a major role in the robot assembly. The gripper is connected to the end of the elbow segment to form the wrist joint. The gripper has roll, pitch, and yaw motions and has two fingers to grip objects. They can hold objects that are 7.5cm long and weighing 1 Kg.

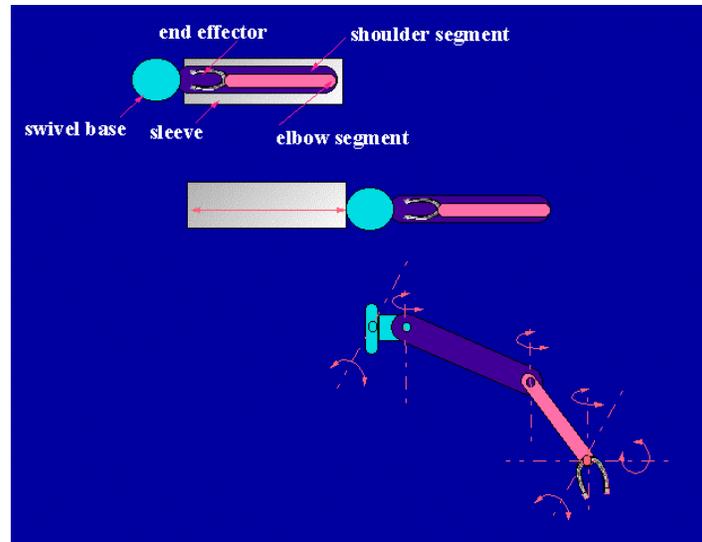


Figure 46: Sequence of motions of the robot arm

The top of the figure illustrates the robot arm folded and in resting position within the sleeve. When the user activates the arm, the arm moves forward on the track within the case. When the arm is out of the sleeve it unfolds itself and is ready to receive further commands. The movement of the arm is similar to a human arm movement such as the swivel base provides 360 degree movement and rotates the entire arm up and down. The shoulder moves in and out as well as the elbow. The gripper opens and closes, rotates 360 degrees, and moves up and down.

The picture of the robot arm (R2) mounted on a wheelchair is shown in Figure 47.



Figure 47: Overall view of robot arm

User Activation Site: Interface Units

The design objective is to have an end-point controlled manipulator system. The inputs of the user should map in an integrated manner to the outputs of the manipulator; three dimensional position mapping of the user's input position signal to the position of the arm's gripper is desired. This provides an interface system, which is intuitive and easy to use. It is also desired to have a Force Feedback system such as forces felt at the gripper are

transmitted back to the user's activation site via cables. A direct connection between the user and the arm can facilitate a system, which is easy to use if proprioception and force reflection are built into the control system. It has been shown that a system which compliments visual feedback with sensory channels is superior to visual feedback alone. Based on the market analysis it has been found that majority of the disabled user's are familiar (84%) and had used joysticks followed by 72% who had used a remote control unit. Since input device familiarity and acceptance is critical for the overall acceptability of the design it is preferred to choose a joystick system as major input device along with other support systems. Another issue in the choice of the input device is the involuntary arm movements and possible impact of these movements on the input device. The availability and decreasing cost of voice control systems make them a potentially strong contender in this area, however conventional (two and three degree of freedom) joysticks, due to their low cost and reliability are also likely choices. More exotic forms of input devices, e.g. ultrasonic and eye movement are not considered due to unfamiliarity of the user, and the unintuitive characteristics, and unknown reliability of the systems.

American Journal of Occupational Therapy, 41, 745-747.
Chameleon: A body powered Rehabilitation robot.

Wheel Chair Robot Safety

Gripper:

Definition: The gripper is used to pick up and hold the objects firmly and place them at another place.

The gripper should have rubber-faced V-grooves in each jaw for grasping cylindrical objects and the length of the linkage should be increased in order that the gripper would open wide enough to grasp an object. The wide end plates should be used for grasping flat objects, while being thin enough to insert an object into a confined space.

Primary safety should ensure that the robot is not able to hit the person. There are, however, situations where the object being held by the robot is required to come in contact with the user, e.g. feeding, and this must be allowed. The system should be sufficiently low powered and runs at such a low speed not to pose a serious safety hazard, but it is vital from the point of user confidence that the arm should not be able to hit the user.

Secondary safety is more important for a number of domestic tasks. Apart from any actual danger, the user is not able to take any remedial action such as wiping up a spill. Spills can occur if the gripper does not securely hold the vessel, allowing it to drop.

Wheelchair stability:

The stability of wheelchair changes when the user leans forward or backward. The stability

decreases in the direction of lean and increases in the opposite direction. So the center of gravity of the wheelchair and the user should fall in the same straight line to improve stability. Moreover the dimensions of the wheels should match the body of the chair.

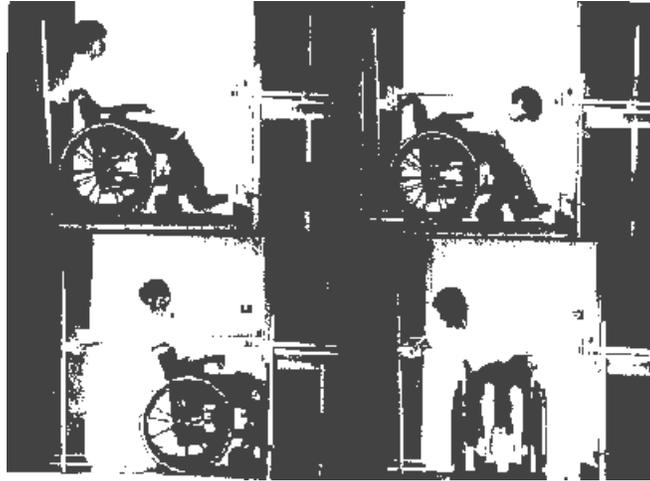


Figure 48: Wheel chair stability

Motor Safety:

The motors used should be servomotors or stepper motors with lightweight shafts and bearings to reduce friction. The transmission of power from motors to robot joints should be accomplished using friction free belts. Finally the motors should be installed at the base of the wheelchair.

Emergency Robot Braking

The speed should not exceed certain level. The speed of the motors should be optimum, which would be safe for the user. A separate braking system should be implemented which would stop the power transmission from motors to joints, whenever the speed increases.

Electronics Interface:

The most important aspect of the control is the hardware interface. So care must be taken to use only compact, lightweight, use of minimum power to operate. It consists of power supply, amplifier, signal generator ,motor drive and control, battery level sensing ,motion control chip, etc. The system should be prevented from malfunction when the battery voltage falls. The user should be informed or must be able to find the number of hours the battery can work. This should be informed to the user through a warning signal.

Emergency Stop Button:

As an extra safety feature there must also be a separate switch which stops the whole system when it is pressed. This switch must have the highest priority of all the keys.

Proper Handling of Equipment:

The equipment should be checked regularly. The transmission belts be replaced frequently. All the motors should be serviced regularly. All the interface circuits should be checked periodically.

General Robot Safety Standards:

The robot safety standard is a performance type standard. It's flexibility for the robot owner in designing a safe robot installation. Hazards results due to improper maintenance of the robots.

The Major Types of hazards are:

- Human errors.
- Mechanical errors.
- Electrical errors.
- Control errors.

Impact is the major type of accident. This includes being struck by either the robot itself, its parts or various items being carried.

a.) Flying objects: This includes parts or workpieces thrown from the end effector .

b) Cutting/shearing: This results in puncture type wounds as a result of sharp objects such as robotic tooling or edges of the robot running against the body.

c) Push/pull or caught: This involves a user becoming hung up on a industrial robot and then being thrown into other items or against the ground which results in injury.

Robot Safety Analysis:

The objective of safety analysis procedure is to specify the design and procedural factors which can:

- reduce the number of accidents/hazards
- reduce the size or dimensions of the danger zone.
- reduce the period the danger exists or minimize the time period the user is within the danger zone
- improve the perception and understanding of the risks involved.
- create escape possibilities at times of risk.
- train and create awareness among users for recognizing, controlling and eventually eliminating risks.

Robotic Safety Procedures and Methods

Software/Control System:

To control the various movements of the robot, software must be developed. This control software developed should be efficient and able to move to the specified directions only. It should be able to move to home position when given instructions.

Programming:

Programming to control the various positions of robot should be done correctly. Only experienced people must be allowed. During operation the program should not give any fatal error, which may result in a serious accident.

Ergonomics:

The design should be simple and the cost should be minimized. The robot should be of light weight and it should be in a position to move from one place to another. The control circuits should not be complicated.

Maintenance

The equipment should be checked regularly. Motors should be serviced regularly. Both motors and chains will need to be replaced after their useful lives have been exceeded. Care must be taken to check interface circuits periodically. Power supplies and voltage units should give required power and voltage. If not, they should be repaired or replaced.

Liability Issues:

Policies & Standards

a. Legislation Related To Disabled People

Legislation that takes the needs of people with into consideration is often broader in scope than the general legislation, typically specifying such matters as financial obligations and reimbursement. Among the activities behind initiatives for new legislation on disabilities, we find the associations and institutes for disabled people and, for example, in the case of Finland, the National Disability Committee, which has representatives from several different disability organizations and from the national agency for social welfare. The normal procedures for creating a new law in this field in Finland are:

First, a disability organization or the Disability Committee sends a letter or memorandum to a ministry, to members of parliament, or even to the press, pointing out a need for legislation in some area. Second, a ministry working-group prepares a first draft act,

which is sent to the relevant organizations for comments. Third, a final draft act is then prepared for passage by the legislative assembly (parliament). Last-minute changes may be proposed by organizations and by individual members of the legislative assembly before the act is passed.

Although there are no formal type approval organization or type approval procedures for technical aids, some checking is performed by social welfare agencies or medical rehabilitation centers etc., since most aids are paid for by the municipal or state agencies. The municipality may listen to users' expressed preferences when acquiring new aids, but often they will not, letting only economic reasons dictate their purchasing policy, since they have the power to decide whether or not to pay for a particular aid.

b. Adaptation & Interaction

The British amended Disabled Persons Act (1986) reinforces some of the provisions of the Disabled Persons Act (1970) and the Housing Act (1974) which allow for carrying out adaptations in the dwelling.

The British Education Act also states conditions for providing the technical equipment necessary for disabled school children to allow them equal participation in education, while local Education authorities will furnish technical aids within the school environment.

Through the British Manpower Services Commission, the Employment and Training Act (1973) assesses disabled persons' needs within the work place and provides for giving technical aids, enabling disabled people to be employed.

The British Department for Health and Social Security provides, among other things, various environmental controls. Local Health Authorities provide orthotic devices, hearing aids, surgical appliances and low vision aids.

British Local Housing authorities can fund housing adaptations (10-25 % to be carried by the client) and local Social Services will furnish technical aids for disabled people to allow independent living.

D. A.R.M. (Assistive Robotic Manipulator) (The Ohio State University)

The OSU Arm

In addition to the parameters established in the preliminary meeting in Philadelphia, the Ohio State team incorporated its own goals into the success of this project. Keeping in mind simplicity, a five degree of freedom (DOF) arm was decided upon as optimal for the manipulation of the required object and compatibility for other tasks. Also, base rotation, shoulder pitch, elbow pitch, wrist pitch, and wrist rotation were chosen as the desired motions for these same reasons. In order to resolve the large moment arms created in the

individual links, the motors were to be located at the base. Finally, for Ohio State's individual task a working simulation of the programming was desired.

Design

Optimization of the links is first analyzed in the design. The largest workspace is established with the upper and lower arm lengths set equal. This is verified by looking at the human arm, which consists of two links of approximately equal lengths. Using this idea and basic geometry, the link lengths are then calculated to be 31 inches for both the upper and lower arm. This calculation assumes that the base of the lower arm will be located 30 inches off the ground and the lower arm can not go below horizontal (the worst case scenario). Also, the shorter the wrist the greater the workspace, so a wrist length 2.5 times shorter than the arm links is then chosen.

Several options for power transmission systems are possible for this linkage: timing belts, chains, shafts, and cables. Cable and pulley systems are selected because they offer zero backlash and minimal friction characteristics. In addition, cables are compact, lightweight, tolerable of some degree of misalignment, and able to withstand tensions of nearly 2000 lbf. with a diameter of only one-eighth of an inch.

The detailed design of the arm begins with a definition of important design criteria including: low cost, the placement of all motors in the base, a cable tensioning system, minimal weight, and mass counter balancing so that smaller motors can be used. With these parameters in mind, a design is conceptualized to connect the links with steel shafts. The links are made of aluminum for lightweight, and designed so that their centers of gravity are located as close to the rotational joint as possible. Complete counterbalancing is not achieved, but significant reduction of the required torque is accomplished by moving the center of gravity close to the rotating joint.

In order to locate all the motors at the base of the robot, the design requires five separate cable and pulley systems, one each for the shoulder, elbow, wrist joints, wrist roll, and one for the gripping force. Movement in the shoulder, elbow, and wrist joints is achieved by rigidly fixing a torque-transmitting pulley to the moving link. A torque-transmitting pulley is fixed the appropriate link so power can be transferred to that joint and linkage movement achieved.

The elbow and wrist joints are slightly more complicated in that the cable cannot run directly to the motor, but must be routed through the arm. This is achieved by the use of a transmission pulley, which spins freely on the shoulder joint, and acts as a link between the elbow joint and the motor. A transmission pulley is placed at the shoulder joint in order to move the elbow joint. Similarly, the wrist movement requires a transmission pulley at both the shoulder and elbow joints. The arm transmission system can be seen in Figure 49.

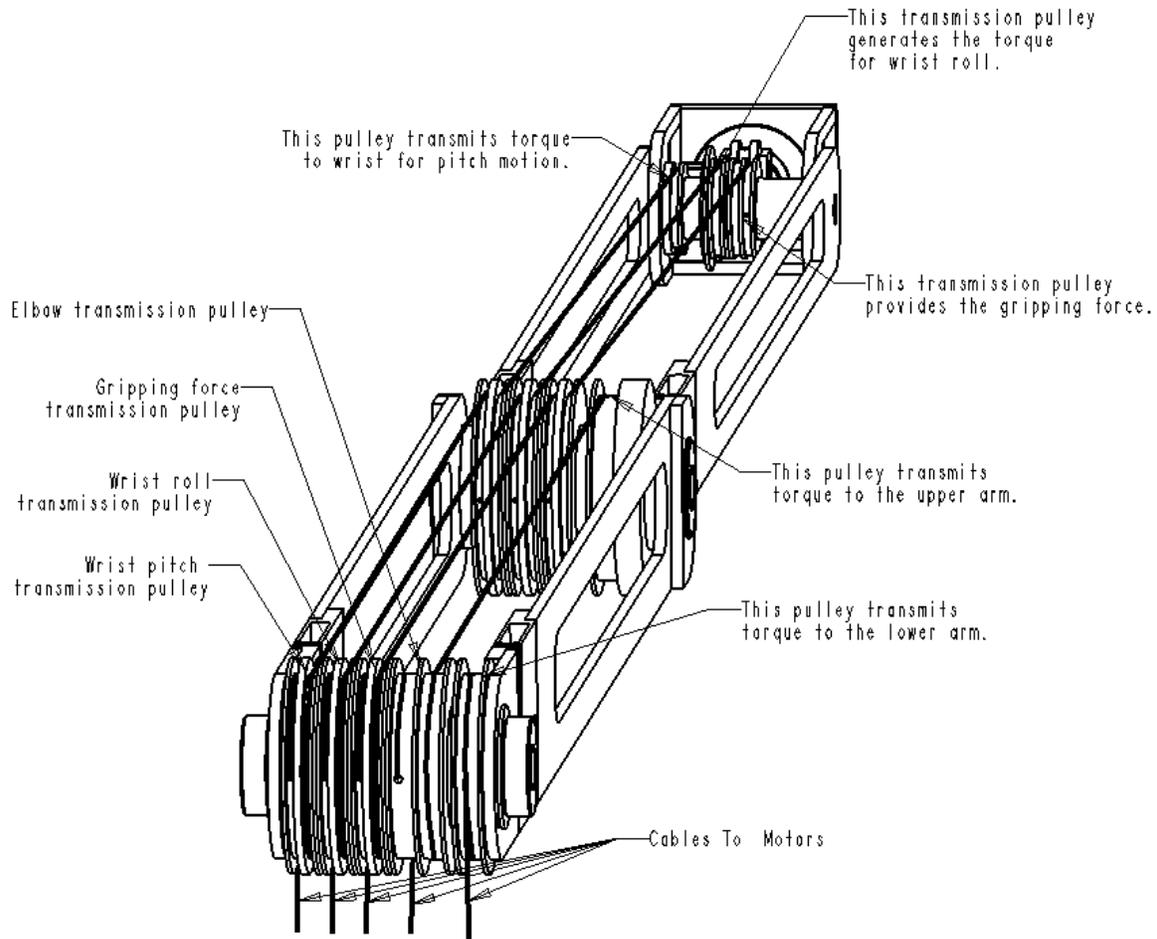


Figure 49: Arm transmission system

Wrist roll is similar to the wrist joint in that it also requires transmission pulleys at both the shoulder and the elbow. In addition, the wrist roll system has a transmission pulley that spins freely on the wrist shaft. This transmission pulley is rigidly fixed to a bevel gear, which then transmits the torque in the desired orientation. The wrist transmission system can be seen in Figure 50.

Furthermore, the necessary gripping force is created by three transmission pulleys, one each at the shoulder, elbow, and wrist. The transmission pulley at the wrist is connected to each finger on the gripper. When the transmission pulley turns, the cable is wound which causes the gripper fingers to close.

A problem that arises with this method of power transmission is the careful attention needed for the alignment of the pulley grooves to avoid the rubbing of cables and other difficulties. Perfect alignment is obtained in all but two cables; the wrist roll and gripper force cables, however, the misalignment of these cables is insignificant.

Finally, a cable tension system is incorporated to remove any slack from the cables.

When the cables are assembled to their respective pulleys a small amount the slack cannot be removed. The arm is then lengthened (up to one inch) until the cables are tight and is then fixed in that position with the use of machine screws. The cable tensioning system can be viewed in Figure 51, where it has been attached to the rest of the link.

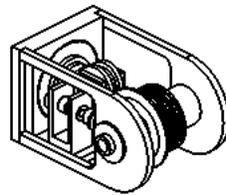
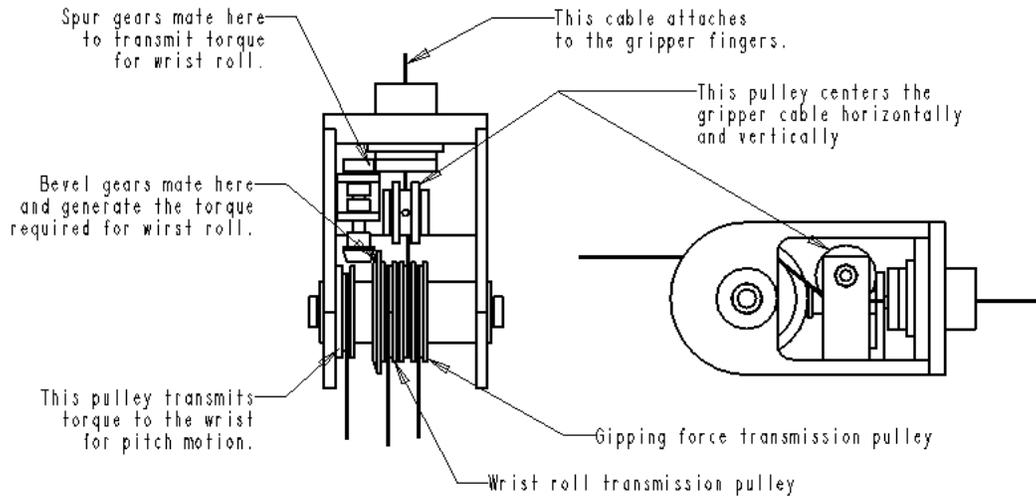


Figure 50: Wrist transmission system

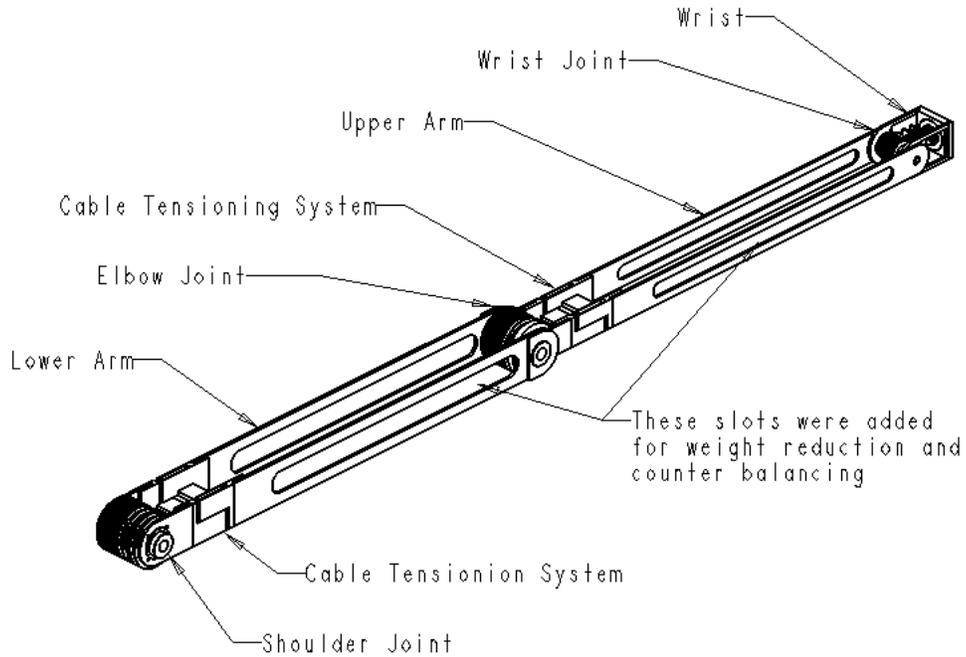


Figure 51: Structural frame of arm linkage

Optimization of the system also includes a stress analysis on each link to assure that failure does not occur. The stresses in the beams are partially generated by the weight of their own masses. At the same time, the mass of each link is a function of the cross-sectional area and strength, presenting an iterative process for this analysis. The arm is positioned in horizontal alignment (greatest stresses are developed in this position) and the masses and centers of gravity are assumed for each link, allowing for the heights of barstock required to support the assumed weight to be calculated. Using this height, the mass assumption can then be checked for validity, and the process repeated with the new values as needed.

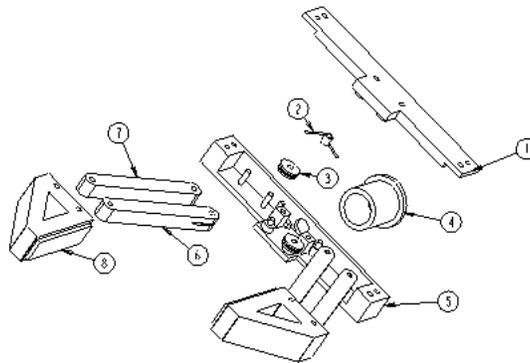


Figure 52: Gripper assembly

A slot is created in the right half of each of the frame beams. The slot not only reduces

the overall weight, but shifts the center of gravity to the left, reducing the moment in the beam and the torque required at each joint. With these recommendations in mind, a solid model of the link is generated as shown in Figure 53. This skeleton is then encased for safety, aesthetics, and its own protection. After performing a dynamic torque analysis it is evident that the dynamic components are entirely insignificant, and only the static torques need to be considered. The torque required in the motors never exceeds the values calculated in the horizontally aligned position. Therefore, if a motor is able to produce the torque required to move the manipulator from this critical position, it can move the manipulator at any other position. Permanent magnet (PM) DC motors are chosen because they are smaller, lighter, and more efficient than their wound counterparts. Additionally, the torque-speed characteristics of these motors are nearly linear, simplifying their control. Gearboxes are also used in conjunction with the motors to increase the torque and reduce the speed as required by this system. The speed is further reduced, without affecting the torque, by varying the voltage.



Figure 53: Final linkage design

The gripper design implements torsional springs and pulleys for opening and closing. The springs force the gripper ends open, while the chord, driven by a motor at the base and aligned by pulleys, closes the ends. This spring and pulley system is also encased for safety, aesthetics, and its own protection. Also, the parallel linkage design allows for the gripper faces to be maintained in the same direction during opening and closing motions. The assembly without the encasement can be seen in Figure 52.

A plate with three aligned holes prevents twisting of the cable from the wrist rotation motion. The cable from the wrist attaches to center hole, and allows the plate to freely

rotate about it. The gripper cables are similarly attached to the outer holes to transmit tension from the motor to the gripper. This motor is placed in series with a resistor, configured to cause the motor to stall at a the maximum torque necessary to exert a 1 Newton force on an object. This allows the gripper to maintain this force as long as the ends are being closed.

The component for attachment of this arm to the wheelchair is designed to affix to the chair on the opposite side of the joystick controller, in a similar fashion to the joystick. This design was made to be flexible enough to work with all different types of wheelchairs and able to be quickly removed with little difficulty.

The completely assembled arm with the gripper mated to the wrist can be seen in Figure 53. The entire design is estimated to have a production and assembly cost of \$4180.00.

User Interface Programming

In order for a computer to control the motion of the linkage system mathematical relations for the positioning of the system must be established. More specifically, it is necessary to define the end effector position in terms of the joint angles. The transform matrices for the linkage are described and solved with the gripper as the end effector. Each transform matrix is a description of how each link is positioned relative to the following and preceding links. An *embedded* coordinate system is created for each link with the Z-axis aligned with the axis around which the joint rotates. The *embedded* coordinate systems can be seen in Figure 54.

After the coordinate systems are defined, the relative translations and rotations between links are described by four parameters in a special notation, which then allows the transform matrices to be formed. The final link end effector position is then found relative to the base link by multiplying the matrices. The only components of this product matrix that are relevant to this system are those that define the x, y and z coordinate in respect to the base.

Since the type of input device for this system has not been established, it is assumed that the input is an incremental Cartesian coordinate for the end effector. The output to the motors is a specific angular position. End point control is a necessary parameter for this project, but is impractical for the wrist pitch, wrist roll, and gripper motions. Therefore, it is necessary to determine the corresponding angles for the specified coordinate, but only for the first three joints. Inverse kinematics can then be implemented for this.

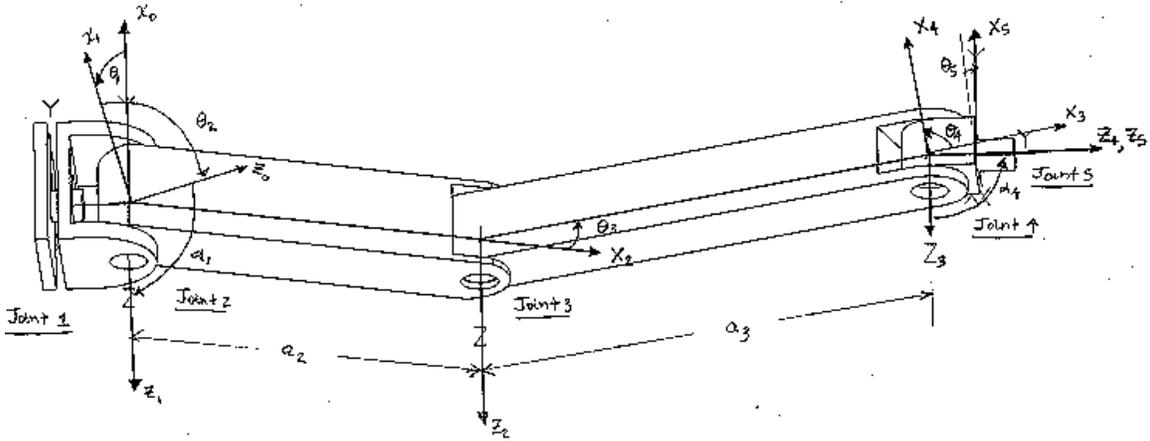


Figure 54: *Embedded coordinate system*

The inverse kinematics results in two separate arctangent relations for each of the first two joints. The third joint has a single arctangent relation, and the calculations for the decoupling of the fourth joint are solved knowing the first three angles. It is also assumed that the fifth joint, the gripper, is directly controlled by the user.

Matlab programming is now used for a graphical simulation of the system to test the relations derived from the transform matrices and inverse kinematics. It uses these calculations to define the end points of each link, and proceeds to plot those points with lines connecting them, representing links. Since Matlab's 3-D graphical capabilities are very limited, the third dimension is incorporated into a 2-D view by changing the lengths of the links to show depth as the robot turns in and out of the plane.

The user interfaces with Matlab to input a coordinate for a new location. The movement is then broken up into smaller increments, similar to the assumed input in the real system. The correct angle must be specified of the two found in the domain from zero to 360 degrees. This is accomplished by specifying the angle closer to the previous angle of this joint. This method has its limits, as the angles exceed the limits of this domain, the wrong solution is chosen. If the robot uses a limited range of motion within the specified domain this problem is solved. Although, a limited range of motion is not used in this program, it needs to be established for this reason, and for safety.

C can now be implemented to test these calculations in the programming language defined for this system, and can be easily modified into an *absolute move* command practical for the prototype. Since no graphics are available numerical results corresponding to those in the Matlab program are needed.

C++ is used for this preliminary prototype. This language was chosen because of its object-oriented characteristics. C, unlike Matlab, cannot handle operations on matrices. C++, however, allows for data abstraction, and the operators to be modified, or *overloaded*, for such processes. If the final program does not use the transform matrices, just the inverse kinematics, the object oriented capabilities will not be necessary, and C can

be used.

The C code is basically a translation of the commands in Matlab. This program's success is based on the same limited range of motion needed in Matlab. This program is not incremented like Matlab, which causes the wrong angles to be calculated. Therefore, it is still necessary to increment this movement. This is accomplished by incrementing in the input from the interface.

For the actual system an onboard computer is required to perform the mathematical computations and control the motor-feedback systems. The embedded PC is chosen because of its use of familiar software packages for development and debugging purposes (*i.e.* Borland C/C++ or Microsoft Visual C++). In addition, numerous hardware vendors provide commercial solutions.

Specially designed interface boards are also required. These interface boards will attach to the PC/104 computer to control the motors. The interface boards generate motor control signals, provide power for the motors and keep track of the shaft encoders. Commercial integrated circuits are available and provide a packaged solution.

Each permanent magnet DC motor is equipped with an incremental shaft encoder. The shaft encoder is physically mounted on the motor, and its motion is directly coupled with that of the motor.

E. W.A.R.M. (Wheelchair Assistive Robotic Manipulator) (University of Pennsylvania)

In addition to the design specifications set forth in October, there are five additional design considerations which were necessary in the development of our manipulator: minimal internal forces, low cost, low weight, high functionality, and aesthetics. Minimizing the internal forces is crucial. The smaller the forces that need to be endured by the motors, the less chance there is of motor inaccuracy, or failure. Furthermore, it is desirable to keep the forces on each motor as consistent as possible. Obviously, a primary consideration is the cost of the robot. None of the available manipulators are affordable. The intent is to create a superior robot for less than \$4000. Since the robot will be wheelchair mounted, it is essential to minimize the weight of the robot. The larger the weight of the robot, the harder it is to transport. Furthermore, heavy robotic arms will lead to unstable, jerky motions. The goal of the robot, as set forth in October, is to be able to pick up a one kilogram object off the floor. It is necessary to design our robot not only for this task, but for a variety of other tasks, such as turning on/off lights, opening doors, etc. These additional tasks do not involve any further detailed design. They are easily accomplished by a clever selection of the end effector. And lastly, since the target consumers are children, aesthetics is a concern. We want our users not only to have a functional robot that is going to make their lives easier, but also something that may be seen as fun and enjoyable to use.

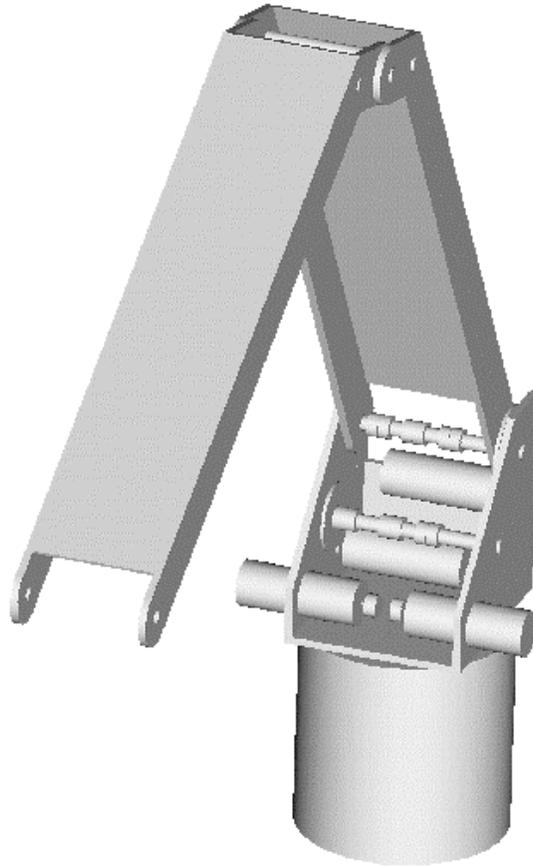


Figure 55: The W.A.R.M

After developing these design considerations, it was necessary to define the two general design criteria for the robot, the degrees of freedom and the mounting. Since the device needs to be child compatible, it is important to maximize the ease of use. Thus, the fewer degrees of freedom maintained by the manipulator, the easier it would be to use. It is considered using a four degree of freedom manipulator, 3 for the robot, and 1 for the gripper. By keeping the gripper vertical to the ground at all times, pitch could be eliminated. However, if any task other than picking something of the ground is to be accomplished, pitch of the gripper was necessary, thus adding a fifth degree of freedom (six including the opening and closing of the gripper).

Mounting of the robot is quite simple. The robot rotates on a circular base, that is about six inches high. This base will fit into a bracket which may be directly bolted onto the wheelchair. The robot will be able to be mounted to either side of the wheelchair. The mounting point will be at the front left, or front right corner of the wheelchair. The top of the base will be about level to the wheelchair seat. This mounting position will allow the robot to fold backwards into a position such that it is parallel, and directly alongside, the wheelchair arm.

After defining the general features of the robot manipulator, it was necessary to begin the

detailed design of the robot. The first design feature to consider was the robot geometry. Since the primary function of the robot is essentially to "reach out", a two arm manipulator was chosen (Figure 55). In deciding on the arm lengths, the location as to where the robot would be mounted had to be considered. After measuring several wheelchairs, it was decided that the shoulder joint would be a maximum of 20 inches above the ground. Using 23 inch arms (Figure 56) allows the manipulator to reach just over a one meter radius from the base point.

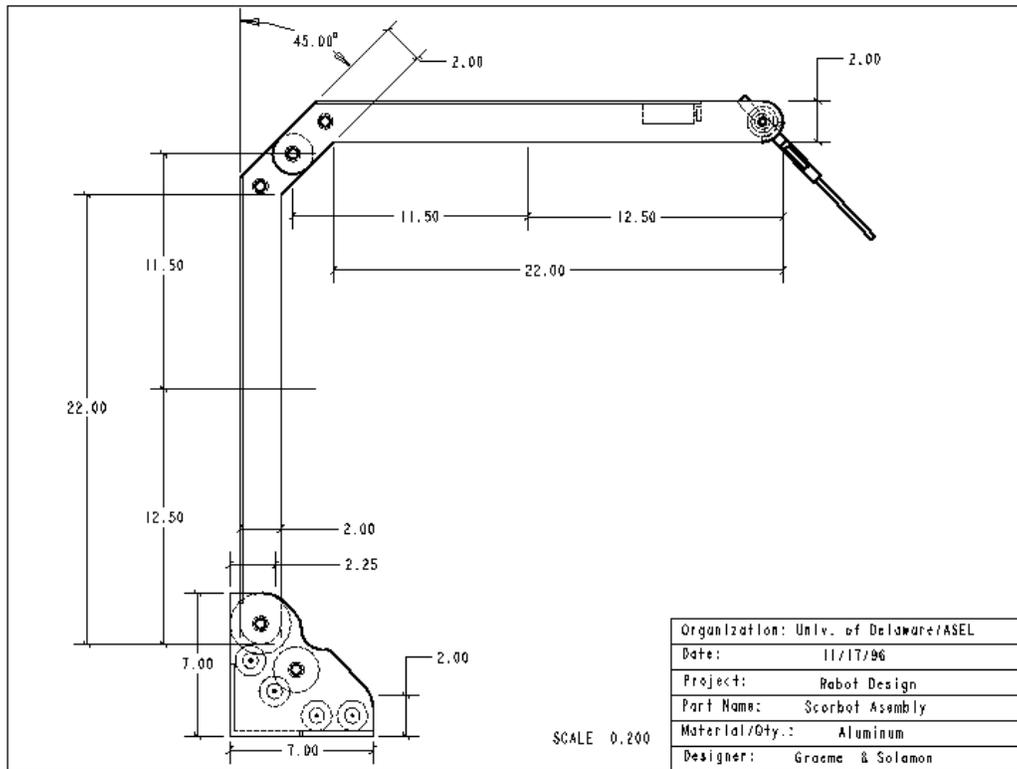


Figure 56: W.A.R.M. assembly drawing

Thus, two degrees of freedom are maintained throughout rotation of the two arms, the shoulder, and the elbow (Figure 55). The other three degrees of freedom are the rotating base, and the pitch and roll of the gripper. The gripper is also capable of opening and closing.

The next consideration was the mounting of the motors on the robot (Figure 55). Since our robot uses two 23 inch long arms, placement of the motors on the robot is very important. For the robot, this means that two motors would be mounted at the base, one at the joint between the shoulder and the elbow, and three at the gripper to control pitch, roll, and opening/closing.

This placement of the motors has two severe disadvantages. Spreading the weight of the motors throughout the manipulator shifts the center of mass of system farther away from the base. Since the design contains a serial linkage, the torque that each motor needs to

exert to keep the arms from moving increases as the base is approached. Thus, the shoulder motor would be responsible for exerting a torque to withstand the resulting moment of the entire robot. As mentioned earlier, a major design criterion is to minimize internal forces. Furthermore, concentrating a large load in the vicinity of the gripper increases its inertia, and thus would lead to jerky motions of the gripper.

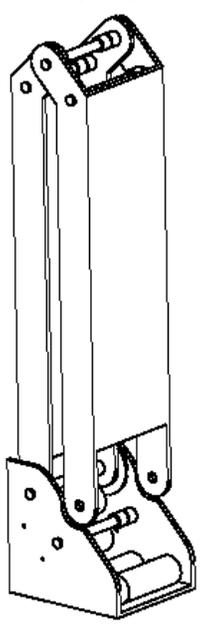


Figure 57: W.A.R.M. in tucked position

In order to see how this problem has been addressed in the past, research was done on existing robotic arms. The two we considered closely are the Scorbot, and the Puma 560. The Puma 560 was found to be inconsistent with the design purpose. The motors are indeed mounted at various points along the linkage. However, this can be done since both the arms and the motors are substantially smaller than the ones made necessary by our robot.

The Scorbot, however, uses motors that are mounted in the base. These motors drive a series of belts and pulleys which extend up to their respective joints. The only motor that is not mounted at the base is a small motor that controls the opening and closing of the end effector. This motor is mounted on the gripper itself.

Thus, all the motors will be mounted at the base, minimizing motor torques, and positioning the center of mass closer to the base. The shoulder motor is geared directly to the shoulder arm. The other three motors use pulleys and belts to control the elbow, and pitch/roll of the end-effector. The two motors that control pitch and roll are coupled. Using a differential gear, the gripper pitches when the two operate in the same direction, and rolls when the motors operate in opposite directions. It was decided to mount the gripper motor at the end of the elbow, rather than on the gripper itself. Thus, it remains out of sight, and does not create a safety hazard.

It was also decided to employ a basic open/close gripper. The gripper is comprised of two

arms, each representing a four-bar linkage. The motor described above drives a lead screw, which translates an assembly up and down. At the end of the assembly is a small link that is connected directly to the four-bar linkage. As the link translates up and down, it rotates the four-bar linkage, opening and closing the gripper. The motor is connected to the lead screw with a flexible shaft. This gripper gives us added versatility, allowing one to perform the tasks mentioned earlier such as opening doors, turning on/off lights, and reaching for different objects.

In order to minimize the weight of the robot manipulator, and to keep the inner workings of the robot from creating a safety hazard, the robot is made of aluminum U beams (square ends). An attractive added feature of this construction is that Plexiglas may be placed over the open face of the beam, such that the operator may actually view the inner workings of the device. Furthermore, the two arms are joined by an offset axis, such that the arms may be folded parallel to each other (Figure 57). This is an efficient default position for the robot when not in use, and also maximizes safety since the arms do not protrude out into the workspace.

The last major design criterion concerned aesthetics. The Plexiglas feature adds to the aesthetics of the robot, allowing the user to view the inner workings of the robot. Furthermore, the aluminum beams could be painted a variety of colors, customized to each user. Adding to the pleasant look of the robot manipulator is the offset elbow and shoulder arms, as described above. The arms, in the default position, are completely parallel to each other. Furthermore, they may be folded down when not in use, such that they are parallel, and at the same level as the armrest. Thus, when the robot is not in use, it is not readily seen. This adds to the safety of the device, allowing the user to travel easily through doorways without getting stuck, and around rooms without knocking over objects.

After our detailed design of the wheelchair mounted robot was complete, it was necessary to build a prototype in order to test the feasibility of our robot. The robot skeleton was built using an erector set. The robot is roughly on a 1:2 scale. All the structural components of the robot were captured, including the base, shoulder and elbow arms with offset axis, and the gripper. The only component that was not built is the wheelchair mount, which has not yet been fully detailed. Using off-the-shelf pulleys and pulley cable, we were able to accurately control each required joint, except roll of the gripper which would require a differential. Furthermore, it was demonstrated how all the joints may be controlled from the base of the robot.

Computer Interface Design

Phase I on the University of Pennsylvania's responsibilities involved developing a proposed design for the computer interface and motor control. A proposed design was developed by determining a method of real time control and selecting hardware for the computer interface. Real time control refers to the method of accomplishing a task using C code to control the robot's motors. As agreed upon in the initial meeting all the code is

written in C++ and uses English units to define constants. The final selections of the are based on experimental and market research.

A) Real Time Control:

Real time control implies tasks are completed at a particular rate of time. The tasks can be executed on a computer or on a motion controller chip. The real time control of the robot will be simplified with use of the LM629 motion controller. Traditionally motion control involved writing code that will send either digital or analog signals to motor driver that in turn sends current to the motor. The real time control code was written in C++ to execute tasks in the computer at a specific rate. When using a computer, real time control involves writing code to control the motion of each motor. The speed and accuracy of the robots motion will be determined by the rate at which the data is transmitted, the gearing and the type of motors used in the design.

The C code time stamps each piece of information received from the board. The purpose is to associate a data set with a particular time. In C this accomplished by writing an Interrupt Service Routine (ISRs), which allows the programmer to interrupt the normal flow of a program to run a specified function. The data collected in this program is received from the input/output board, which is connected to a decoder counter chip. The ISR also sends a pulse width modulated signal (PWM) signal to the motor driver chip.

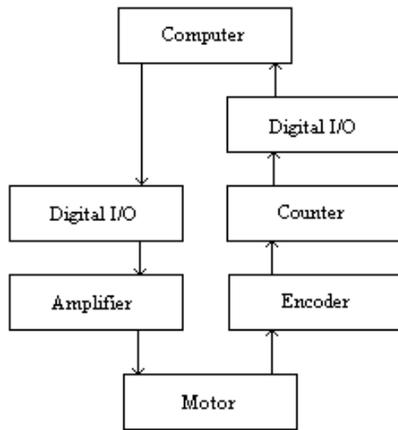


FIGURE 1

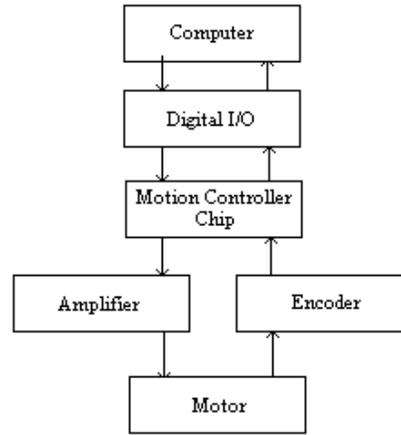


FIGURE 2

Figures 58 and 59: Motor control systems

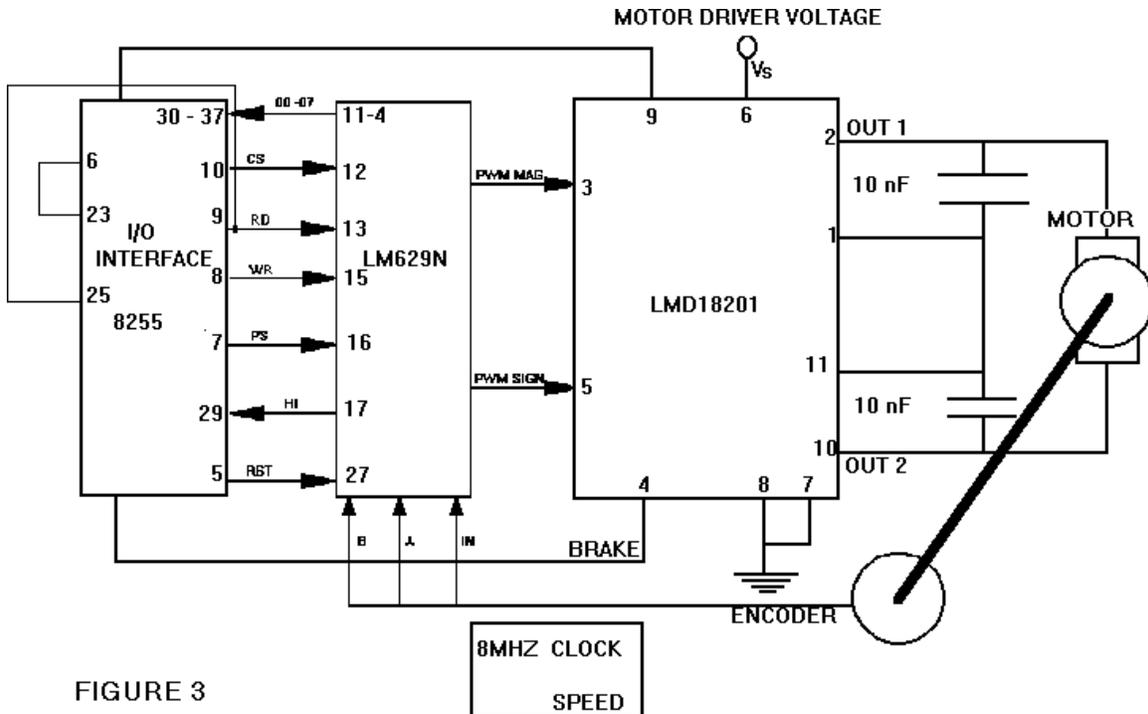


FIGURE 3

Figure 60: Hardware setup

Prior to the January meeting a motor control system was developed by writing C code and using Simulink to send data to the motor. The Real Time Workshop (RTW) module made the code generation easier by creating the ISR's. The various components were setup as shown in Figure 58.

Although responsive this setup produced several problems with noise in the PWM signal sent to the motor. This problem was solvable, but we chose to experiment with the LM629 chip because of its capabilities. The next step in our experimentation involved making the system shown in Figure 59 with the help of a Scott Doyle and Tom Sugar, an electrical technician and graduate student at Penn respectively.

B) Proposed Design:

The proposed design can be seen in Figure 60.

C) Code:

There are several methods to achieve motion control in the robot. However, based on the research a system that saves time and money was developed. A conventional computer interface set comprises of a I/O board, possibly a D/A or A/D chip, motor driver, encoder, decoder, clock and several lines of code to do the time stamping and the computation of desired trajectories. Through experimentation, it was verified that the use of the LM629 motion controller chip by National Semiconductor eliminated the interrupt service routine in the code to compute new trajectories, it also replaced the decoder, D/A converter chip hardware. Functions were written for the LM629 chip to read velocity and position data, send trajectory data, initialize PID gain values using the manuals accessible on the World

Wide Web.

D) Hardware:

The overall purpose of the computer interface design was to stay within the time and financial constraints of the project. In regard to these constraints we suggested the setup shown in Figure 60. This setup consists of the LM629 microcontroller chip, LMC8255 I/O board, cabling, breakout board and a 196 bit I/O board. The robot design contains six motors, therefore each motor will require one breakout board, one cable, one motor driver chip, LM629 chip and a clock. The chip would be powered by a 5 volt non-switching DC power supply. The next was to integrate the motor driver chip, I/O lines and the LM629 chip on to a motor board.

VI. SPECIFICATIONS FOR PROTOTYPE DESIGN

Design Conference at The Cooper Union

Following the design presentations of each university at the January 1997 meeting, the five teams sat down to discuss the key features of each team's kinematic design configuration. After an open forum critique of each university's design criteria, the final plans and specifications were determined for the prototype manipulator. Also, it was decided that each school's team would assume a particular responsibility to assist in the fabrication of the finalized design configuration.

Design Configuration of Prototype

The final design configuration of the prototype was based on several key elements presented by each university at the design conference. The manipulator is to operate using six degrees of freedom (including the open/close actuation of the gripper). The robotic arm will simulate the action of the human arm. It will consist of three joints: a shoulder joint, an elbow joint, and a wrist joint, and two links (upper and lower arms). The six degrees of freedom will represent pitch and roll in the shoulder joint, pitch in the elbow joint, and pitch and roll in the wrist joint. The servomotors will be mounted inside the base of the manipulator to decrease weight in the arm itself. The prototype will be made out of a durable yet lightweight material for this first application.

Transmission System

The transmission system as presented by the Ohio State team in their conceptual design was chosen for the prototype manipulator for simplicity and for the purpose of achieving six degrees of freedom. It will consist of a system of a pulley at each joint and a series of chain running along each arm linkage so that each degree of freedom can be performed from the servomotors, which are mounted in the base.

VII. PHASE II

After noting the best features of each presentation, the students assigned each university a specific task for the second phase of the project - the manufacture of a prototype manipulator. Cooper Union is generating the design for the user interface in controlling the manipulator along with required programming, and is to perform graphical simulation and analysis on the final design of the arm using ADAMS software. Drexel University is to design the skeleton of the prototype manipulator including the design for housing the components and a tensioning system for the chains. Drexel is designing a base for the arm in which the servomotors will be mounted. New Jersey Institute of Technology is concentrating on an extensive market analysis for the prototype manipulator. NJIT is also assisting in the manufacture of the arm since they have solid manufacturing capabilities. Ohio State University is heading the design of the transmission system for the manipulator that will provide the motion to the arm. This task includes choosing the proper cable chains and belts, sprockets, and bearings to satisfy motion requirements. Other responsibilities include assisting Drexel in designing a base for the manipulator. Finally, University of Pennsylvania is choosing the correct servomotors for the proposed design and designing the software to control the motors. All five universities will work closely and communicate frequently to accomplish these design tasks and goals.

A. Arm Skeleton (Drexel University)

Since the design meeting at Cooper Union in January, 1997, the Drexel University team worked extensively in designing the arm skeleton, tensioning system and base for the manipulator. Based on the preliminary transmission system agreed upon by the universities at the January meeting, Drexel began designing the skeleton around this system. The team worked closely with the Ohio State team in developing the final design of the skeleton as OSU was responsible for the transmission system of the manipulator. The Drexel team designed detailed components for the arm skeleton and also for housing OSU's transmission system.

Links

The arm pieces of the skeleton are referred to as the links of the system. There are four links in the design of the skeleton, two for the upper portion of the arm (shoulder) and two for the lower portion of the arm (forearm). The links are made of standard aluminum 6061 for the prototype design and all links are of equal length and thickness (25 inches, 0.25 inches respectively). These dimensions are governed by Ohio State's transmission system of cable chains, chains, and sprockets. The links of the upper and lower portions of the arm are connected using different structural elements such as support blocks, tensioning blocks and covers, as well as portions of the transmission system such as the joint members containing the sprockets. The upper arm is slightly wider than the lower arm to house the system of motion. The key to the link design are the cutouts found in each of the four members. These cutouts serve to minimize the weight of each link, as

weight is a primary concern for the overall design of the manipulator. The team studied different configurations for the links in addressing the weight issue, and upon completion of each trial design, a preliminary one-dimensional beam finite element verified the structural integrity of each possible configuration.

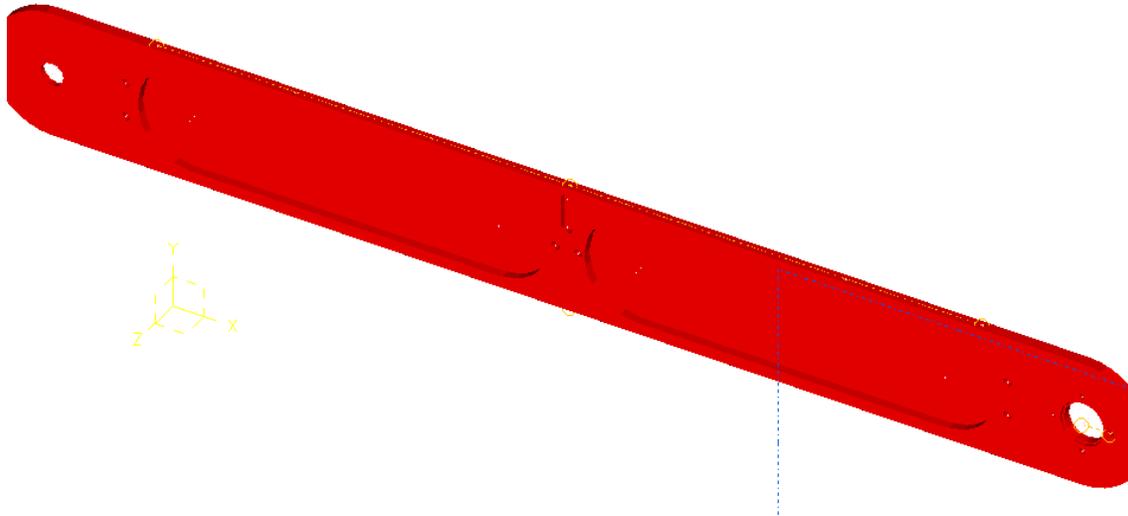


Figure 61: Arm link

Structural Elements

Support blocks found in the upper and lower portions of the arm are referred to as structural elements because they serve as structural supports for the overall skeleton. There are two support blocks for each portion of the arm, connecting the left and right links. In helping to minimize the weight of the arm, the blocks are made out of lightweight aluminum 6061 in the shape of an eye beam. In the middle of each part of the arm, there is an additional support block to serve as the foundation of the tensioning system as well as provide extra support to the arm. Also, the tensioning system of both parts of the arm fit through an additional shaft containing tensioning components, and these shafts also provide structural support to the arm. The shafts found in each joint of the entire arm (shoulder joint, elbow joint) also serve as support mechanisms, constructed out of lightweight aluminum and containing components of the transmission system.

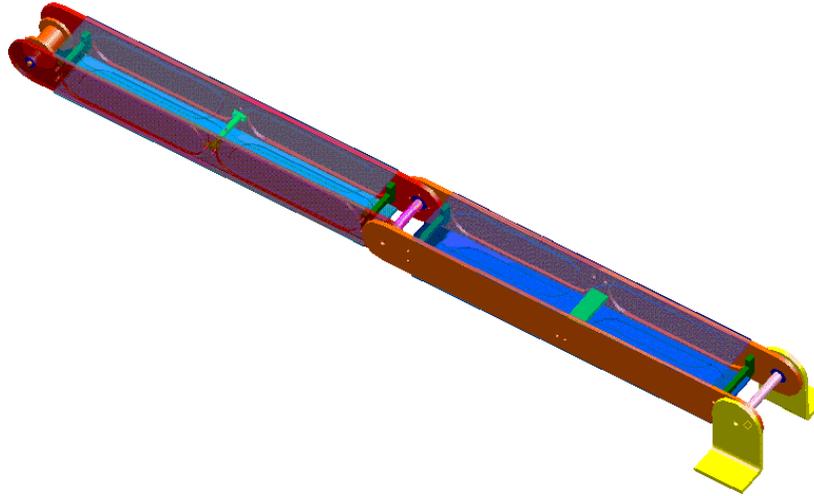


Figure 62: Structural elements of arm linkage

Bearings

There are numerous bearings of different sizes which provide the user with fluid motion control. Bearings are used in various locations, including the differential gearing system to provide pitch and roll to the manipulator's gripper, the joint shafts, the arm tensioning system, and the gearing design connecting the servomotors to the transmission system. Figure 63 shows the detail of the bearings at the elbow joint.

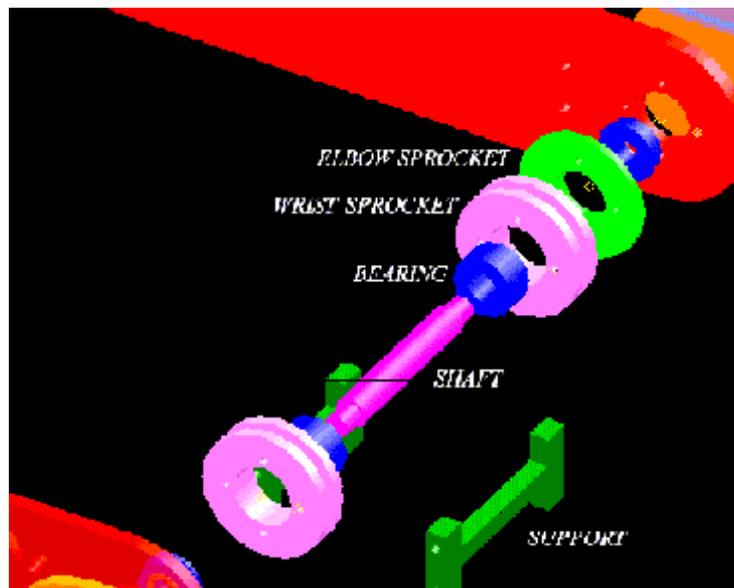


Figure 63: Elbow joint components

Tensioning System

To ensure proper tension on the various cable chains and belts as governed by Ohio State's transmission design, the Drexel team designed a unique tensioning system to accomplish this goal. There are two systems of tensioning, each of which is particular to a portion of the arm assembly. The lower arm system consists of an aluminum shaft containing a strategically located sprocket and bearing to match the location of the belt. There will be a through hole in the shaft to allow for a tensioner screw to fit through which threads in the tensioning support block. The links of the lower arm contain a slot at this location to allow for vertical tensioning in this portion of the arm. The idea behind the system is that as tension is required in the chain, the tensioner screw can be adjusted vertically and the shaft containing the sprocket tightens the cable as it moves vertically downward. The upper arm tensioning system operates on the same principle except there are two shafts and three tensioner screws (two on the same shaft) to tension the appropriate number of chains. There are two slots in the upper arm links to accommodate the two tensioning shafts. The tensioner screws are adjusted as needed to provide correct tensioning to the chains. Figure 64 shows a detail of the lower arm tensioning system and an overall view of the entire configuration.

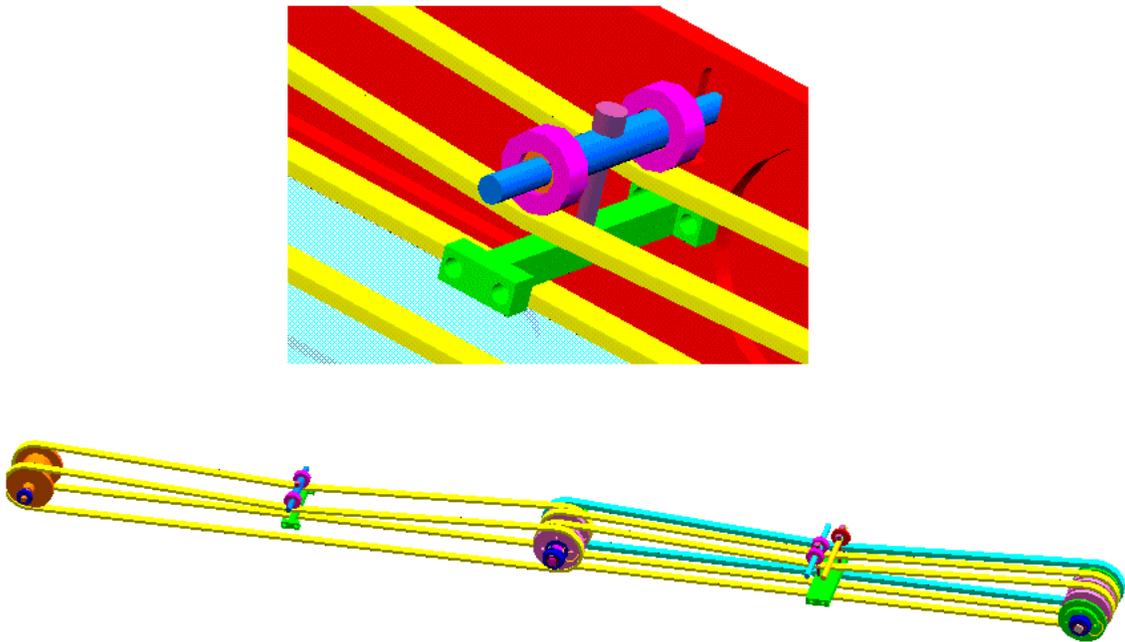


Figure 64: Lower arm tensioning system

Covers

The upper and lower portions of the arm contain covers on the top and bottom sections of

the links to enclose the transmission system. The covers are important to the design of the arm skeleton because they prevent collapsing of the structure if a lateral load should hit the side of the arm during use. They are made out of lightweight aluminum 6061 and also contain cutouts to save overall weight in the arm. The covers are fastened to the links along their outer perimeter. They also serve as a protective device for the user as they conceal the moving parts of the arm and can prevent potential injury. They are easily removed from the links for routine maintenance of the arm.

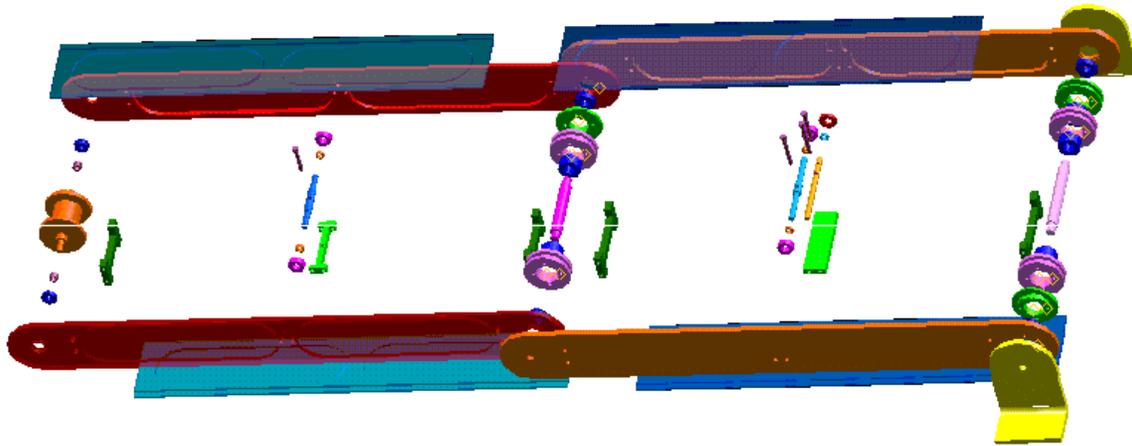


Figure 65: Arm covers

Base

Along with the arm skeleton and tensioning system for the manipulator, Drexel also designed the base of the manipulator to house the servomotors, their corresponding transmission system for the arm pieces, and servomotor power supplies. The base is made of lightweight aluminum 6061 in the shape of a box. The box is assembled using sheet metal varying in thickness from 0.250" to 0.375" and connected with machine screws along the perimeter of each sheet. Within this box, there is another box, constructed in the same way, smaller in size, that has the ability to rotate, allowing the manipulator to move freely about the roll axis at its shoulder. Inside this second box, there are four servomotors, interconnected by a gear train designed by Ohio State University, which provide movement along the pitch axes of the shoulder and elbow, and also motion along the pitch and roll axis of the wrist. Outside this box, within the outer box, there are two additional servomotors that provide the rotation to the inner box and the open/close motion of the gripper.

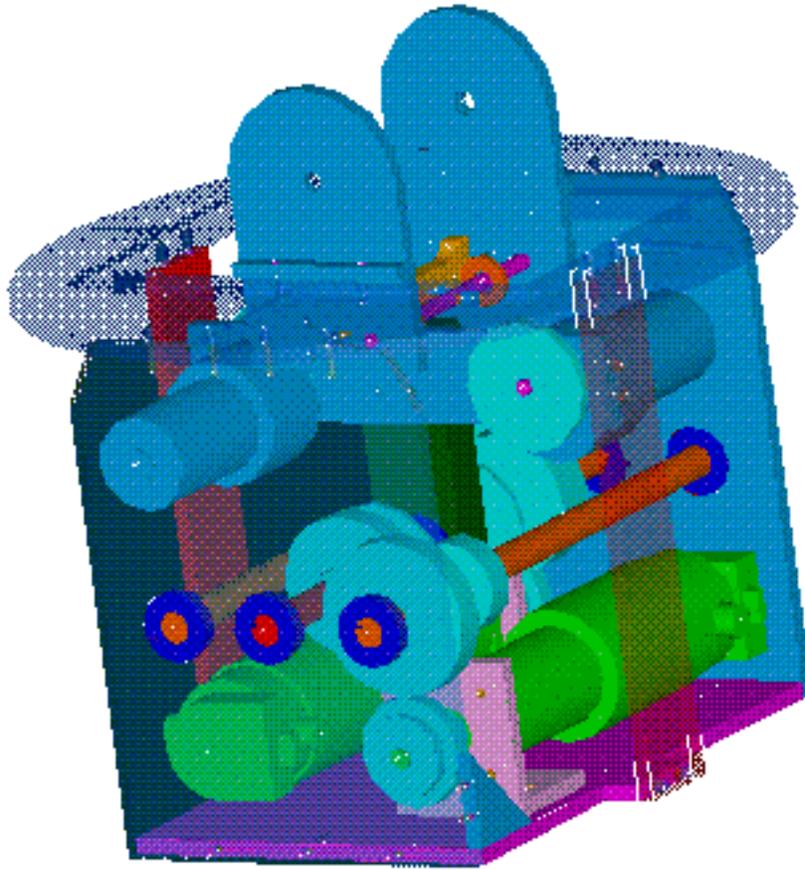


Figure 66: Rotating base

On top of the outer box, there is a circular cutout through which the belts attach to the shoulder sprockets from the motors. In addition, a circular piece of aluminum mounts to the inner box with machine screws. There is a rectangular cutout in this circular piece to allow for the cable chains and belts to transmit power from the servomotors to the manipulator. To provide fluid motion between this circular piece and the outer box (only the inner box rotates), there is a nylon ring to serve as a bearing around the perimeter of the circle. The top of the outer box contains a step at the circular cutout for application of the ring between the inner and outer boxes. Within the rectangular cutout of the inner box, there are two mounting brackets, known as “saddles”, to which the arm attaches to the base by a shaft and also cable belts. There are a number of “L” brackets used in both boxes for the mounting of the servomotors within the base. Also, in the outer box, there are two rectangular brackets, strategically placed on the opposite sides of each other, to provide support to the inner box. Figure 67 shows a side view of the entire base showing the rotation mechanism, saddles, circular piece of aluminum, and the shoulder roll motor.

Similar to the tensioning systems found in both arm pieces of the manipulator, the base also contains a tensioning system to ensure proper tensioning of the belts that connect the gear train to the shoulder sprocket. Like the other systems, the base tensioning system is

made up of sprockets, a shaft, a tensioning screw, and a support block. As tensioning is required in the belts, the screw is adjusted horizontally, unlike the vertical adjustments found in the arm pieces. This tensioning system is found in the “saddles” of the base.

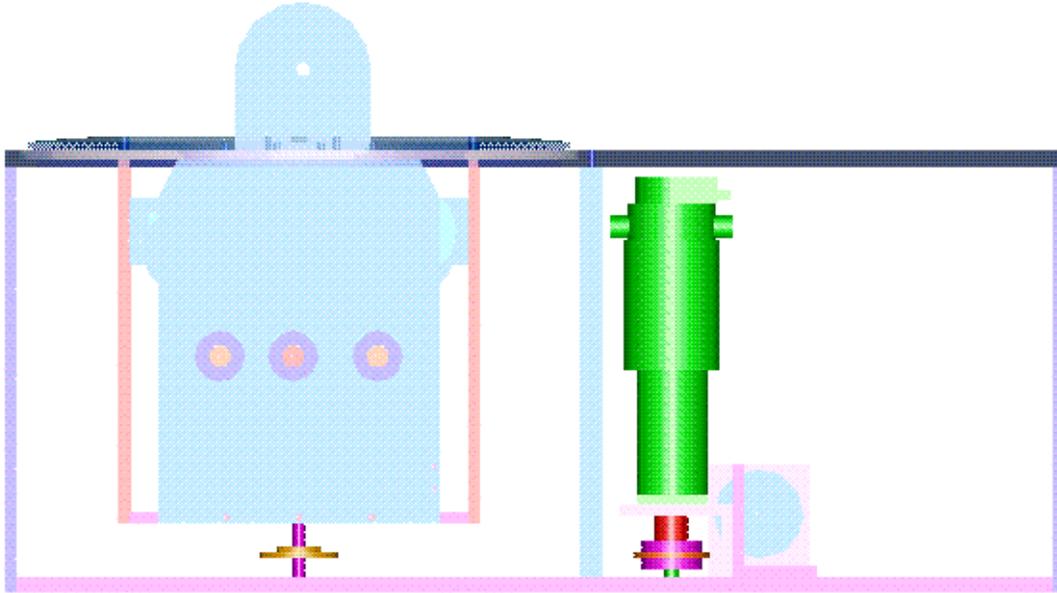


Figure 67: Side view of entire base

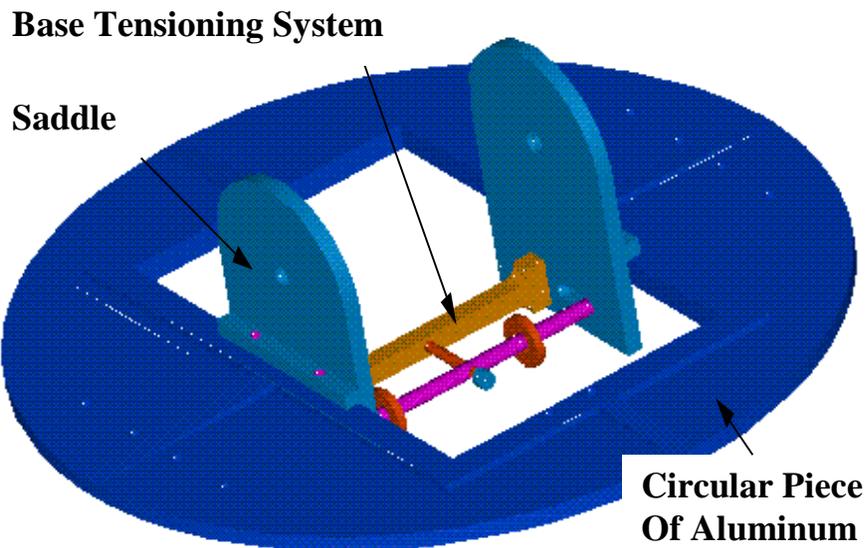


Figure 68: Base tensioning system

B. Power Transmission System (The Ohio State University)

This quarter, the OSU Gateway Team has been faced with many challenges in the design and manufacture of the power transmission system. Specifically, the task has been the specification of all parts needed to transmit power from the motors located in the base of the manipulator to each of the joints and actuation of the base rotation. This task has been broken into three categories: actuation of joints and gripper, gear reduction of motors and routing of power to shoulder joint, and base rotation. Trade-offs have been made which reflect decisions based on cost, reliability, and manufacturability of the entire manipulator.

The first major task of our team was to select a flexible machine element to transmit power from the motors to the joints. The preliminary design selected from Phase I, which can be seen in Figure 69, incorporated high strength cables which were attached to a pulley then wound by the motor, causing torque to be transmitted.

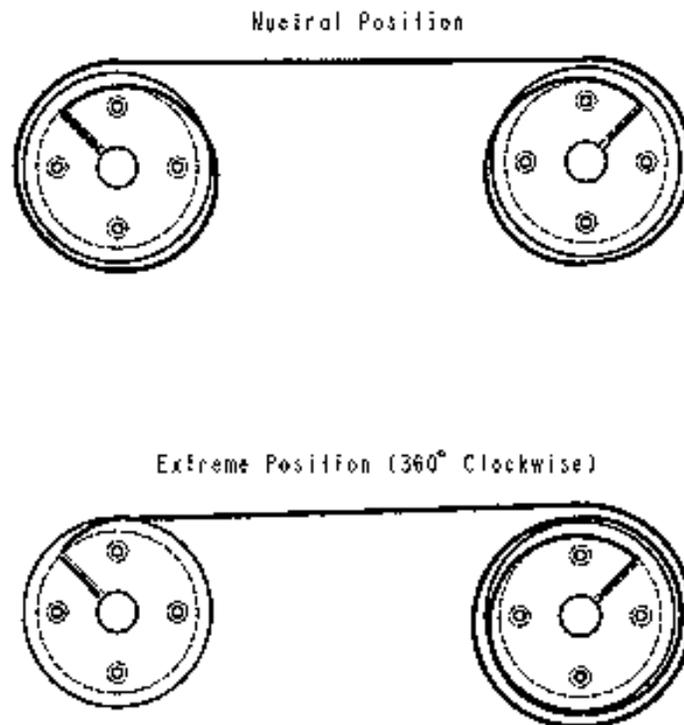


Figure 69: Original cable design

This seemed like a clever design on paper, in that it was lightweight and very compact; however, after thorough investigation of cables, they were eliminated from our design for reliability and control concerns. At the same time, other flexible machine elements such as timing belts, pitch link roller chains, cable chains, and shielded cables were being researched. Timing belts were quickly eliminated because of the width required to transmit the torque. Pitch link roller chains, similar to those used on bicycles, seemed quite promising for the elbow and shoulder joints with attributes such as long life, minimal

stretching, and zero backlash. Furthermore, pitch link chains were not much heavier or larger than cables, weighing only 0.09 lb/ft and measuring a little over one-quarter of an inch wide, and carry a tensile strength twice that of cables. Clearly, pitch link chains (Figure 71) offer a practical design solution for this application. With similar attributes, cable chains proved to be viable option for the roll and pitch of the wrist. Cable chains consist of two cables parallel to one another and connected by polyurethane coating (See Figure 70).

Figure 70: Cable Chain



Figure71: Pitch link chain

Using this information, a final recommendation was made to use 0.25 pitch link chain to control the shoulder and the elbow, 0.1475 pitch cable chain to control wrist roll and pitch, and a 1.8 mm shielded cable to actuate the gripper. Clearly, this new design will be more expensive than the Phase I cable design; however, manufacturing time will be reduced. Furthermore, greater confidence in the control and life of the transmission system has been insured.

Next, bearings and shafts were chosen. Due to the significant forces applied to the shaft and the bearings, special care was taken in selection of these elements to insure long and reliable operation. For the shafting, a hardened, yet machineable 0.5” diameter steel, which will provide a minimum safety factor of three at the shoulder joint, was chosen. Needle bearings with an inner race were chosen for the transmission pulleys because of their excellent load, friction, and size characteristics.

With chains, sprockets, shafts, and bearings chosen, the formidable design task can be attacked. The design had to incorporate a torque-transmitting pulley at the shoulder and elbow that would transfer the torque from the sprocket/chain to the link. A design was proposed (See Figure 72) in which the sprocket attached directly to the link itself.

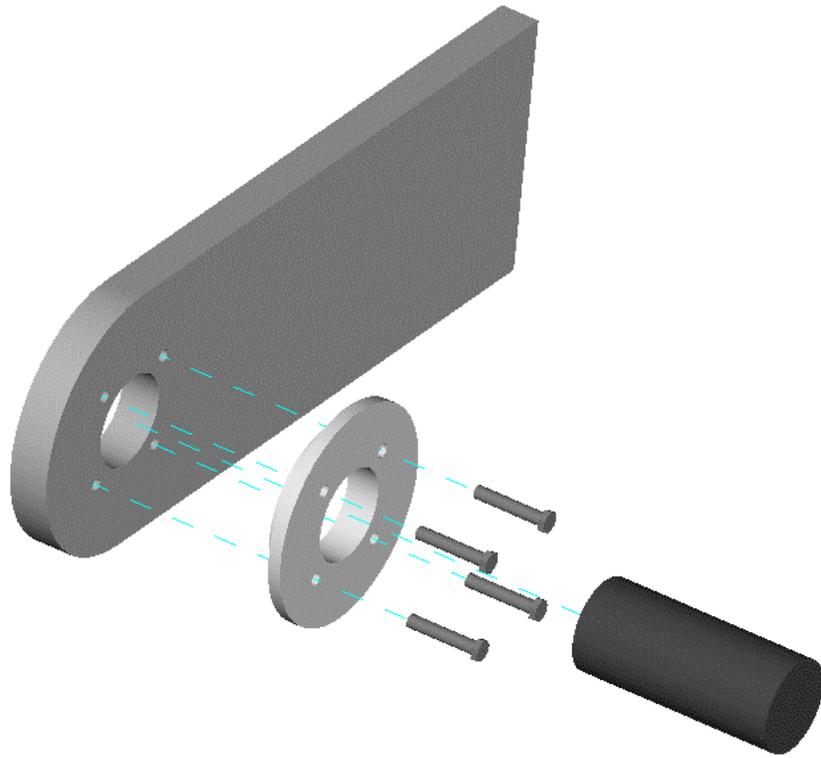


Figure 72: Torque transmitting pulleys

In this way torque is transmitted to the link by both the shear of the socket head screws and the friction force generated between the link and the sprocket. Transmission pulleys were needed at each joint to route the chains to their respective torque pulleys. A design was proposed that transmitted torque via four #4-40 socket head screws and a spacer between the sprockets. This design can be seen in Figure 73.

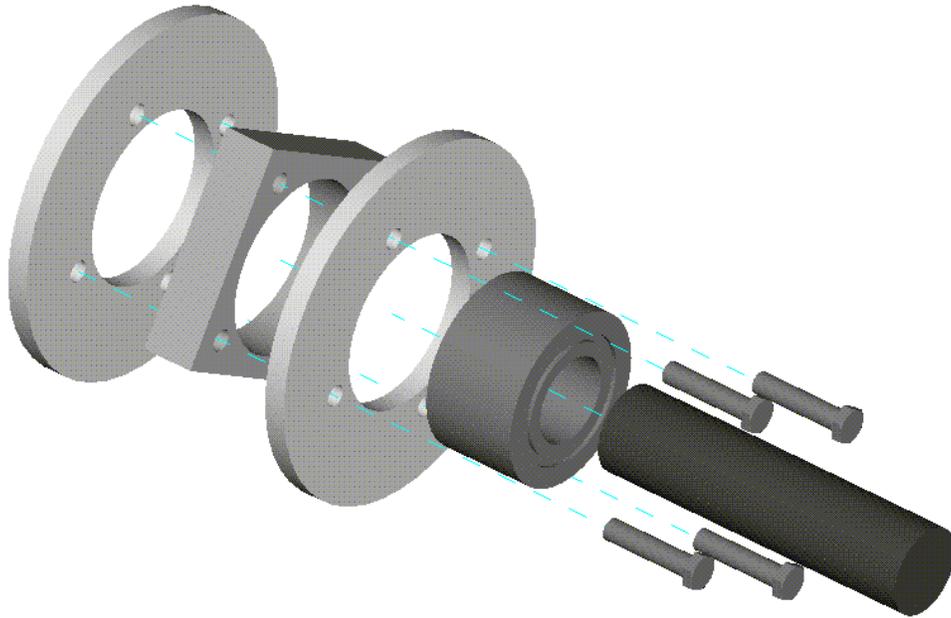


Figure 73: Transmission pulley design

Again, torque is transmitted through both the shear area of the screws and the frictional force created between each side of the spacer and the sprockets. Motion in the wrist is achieved with the use of a differential (See Figure 74).

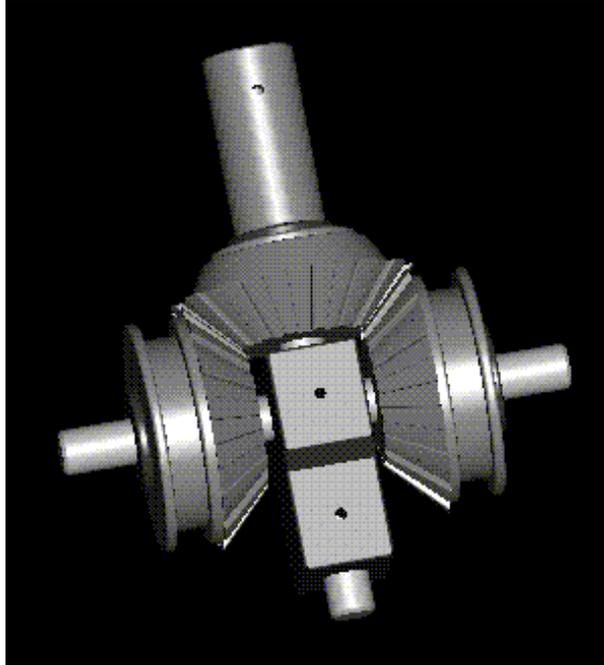


Figure 74: Differential

The differential forces the wrist to pitch if the end gears are actuated in the same direction while causing the wrist to roll if the end gears are actuated in opposite directions. This wrist design is somewhat more expensive than the design of the Phase I wrist (See pg. 59, Figure 50) specified by OSU but offers a part reduction of nearly 20 parts and will significantly reduce the manufacture and assembly times. By using shielded cables for the gripper, a direct and flexible connection is made from the motor to the gripper. The shielded cable is attached to a pulley connected to a motor at the base of the arm and runs through the arm up to the gripping device. A spring at the base of the gripper forces it to stay open. The actuation of the motor shortens the shielded cable length, which forces the gripper to close. Figure 75 is a picture of the gripper and shielded cable, which is used to activate the gripper to a closed position. A complete arm transmission parts list can be seen in the Bill of Materials section of the Appendix.

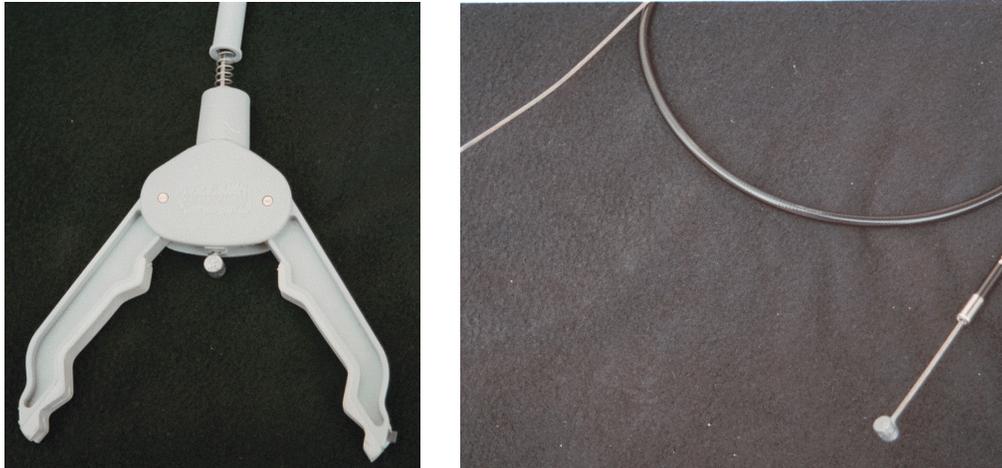


Figure 75: Gripper with shielded cable

Finally, a tensioning system is devised to take slack out of the chains. Drexel University designed the tensioning system seen in Figure 64, which is nicely integrated into the design.

Another major task is to take the torque from the motor shaft to the arm in a feasible manner. The main constraints when dealing with the transmission are to increase the torque as well as reduce the speed, keep the motor housing size small, and to be able to rout and align the transmission chains properly. The most important function of the transmission system is to increase the torque output of the motor by a factor of three in the case of the shoulder motor and a factor of two for the elbow motor. The simplest and most efficient means to accomplish this are by the use of gears. Different types of gears were investigated and spur gears were found to be the best and simplest for our purpose. With the use of spur gears the load is entirely in the tangential direction with minimal losses. Also, the use of a gear train was chosen to minimize the width of the motor housing. Based on the minimum gear size of 1.75 inch pitch diameter for the clutching used, the gears selected are shown in Table 5.

Table 5: Transmission gear sizing

	Shoulder Transmission (Pitch Diameter)	Elbow Transmission (Pitch Diameter)
Clutch	1.75 in	1.75
Gear 2	2.50	2.50
Gear 3	1.00	1.50
Gear 4	2.25	2.25
Total Reduction	3.21:1	2.15:1

A slightly larger gear reduction than necessary is made to account for losses from the motor to the joints. The torque is transmitted along the shafts by the use of keys and keyways. Gear 4 is attached to the same shaft as the sprocket that carries a .25 pitch chain up

to the joints. This design is represented in Figure 76.

The main concern when selecting the gear sizes and materials is breaking of the gear teeth due to high stresses. In order to correctly select the sizes and materials the modified Lewis equation is used for analysis (see sample calculations in the Appendix). It is then found from this analysis that the standard .50 inch gear face will suffice using a steel gear and give a sufficient factor of safety.

The remaining transmission problems were in dealing with the gripper pitch and roll as well as the gripper opening and closing. Since the torque requirements were not that large for either, no reduction is necessary. For the pitch and roll, a sprocket is directly attached to the motor shaft using set screws. The same configuration worked for the open/close except that instead of a sprocket, a grooved pulley and bike cable were used.

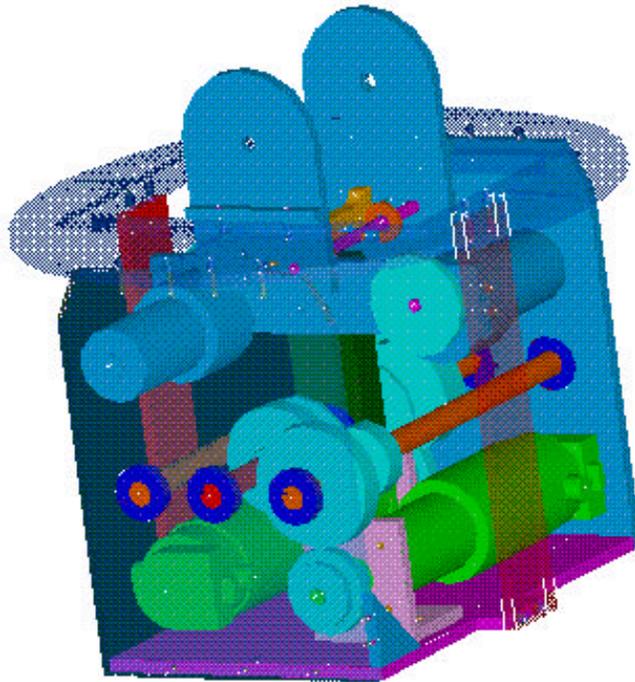


Figure 76: Gear train architecture

Finally, the base rotation is accomplished by attaching a sprocket to the motor and one to the bottom of the base. The entire transmission assembly is then placed inside a larger housing supported at the top by a thrust bearing. The base is then rotated about the center axis inside the housing as seen in Figure 77. Supporting the base at the top enabled the use of a smaller motor and shaft needed to rotate the base at the bottom.

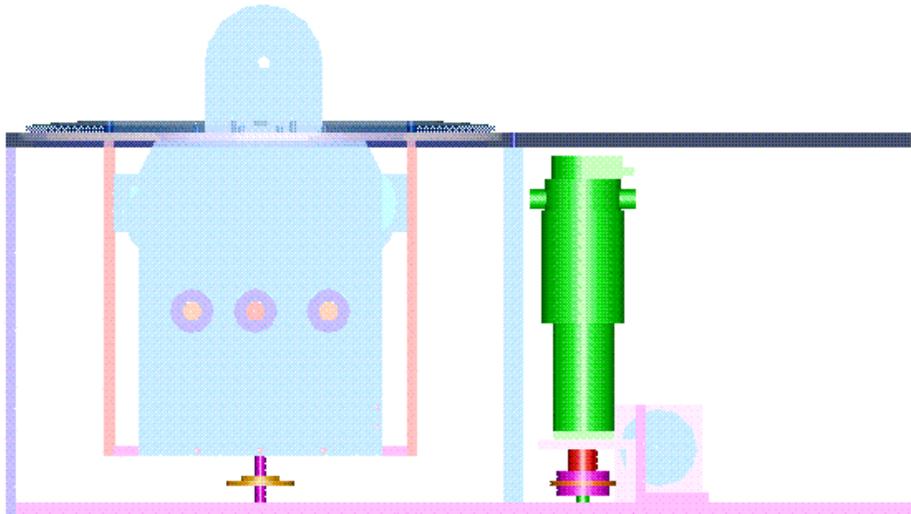


Figure 77: Rotation of Arm

C. Motor/Gearbox Selection and Sizing/Design of Computer Interface (UPenn)

Perhaps the most crucial components to the proper functioning of any robot are the motors. Motors are expensive, and thus they must be carefully assessed in order to assure that they will satisfy the given requirements. The first step in motor selection was to choose the type of motor that was to be used. After studying various robots, including the Scorbot, the Manus Manipulator, and the Helping Hand, it was decided that DC servo motors would best suit our application. Once this was decided, several brands of DC servo motors were considered- namely Maxon, MicroMo, and Pittman. Maxon motors were limited in selection, while MicroMo motors were unusually large and very expensive. The Pittman line of DC servo motors offered a wide variety of different size and power motors at a reasonable price. Furthermore, the motors could be conveniently pre-assembled and purchased with gearboxes and optical encoders to suit our needs.

The primary factor in motor sizing is the amount of torque that is generated at the joint which the motor will rotate. Thus, before beginning, it was necessary to determine the maximum torque at each of the six joints (base, shoulder, elbow, pitch, roll, open/close). These torques take into account the weights of the arms, the gripper, the transmission, and the end-point load. These were determined (by OSU) to be

- shoulder = 5000 oz-in ---> with 3:1 reduction at joint ---> 1667 oz-in
- elbow = 2000 oz-in ---> with 2:1 reduction at joint ---> 1000 oz-in
- pitch, roll, open/close = 150 oz-in ---> with no reduction ---> 150 oz-in

Thus, it is necessary that the output gear shaft of each motor be able to exert the amount of force underlined above, respectively.

Also of key importance is the required speed of each arm of the robot, including the speed at which the entire robot rotates. Considering the maximum velocity requirement of .2 m/s in the Cartesian Frame, it was determined (by OSU) that each gear shaft should rotate at no more than 10 rpm. Thus, considering the reductions at the shoulder and elbow joints, the arms should actually rotate at a maximum speed of 3.33 and 5.00 rpm respectively. Since pitch, roll, and open/close do not significantly translate the end effector, we are less concerned with these speeds. Thus, a maximum of 10 rpm is sufficient. It is also possible to vary the gear shaft speeds through each motor driver, thus slowing down the motors if necessary.

Thus, we have the two basic requirements for each joint/motor; output load and output speed (output is always synonymous with gear shaft). The first two equations to be considered are the required motor torque and motor speed, and are given by

$$\mathbf{S} = \mathbf{G}_s \times \mathbf{R} \quad \mathbf{T} = \frac{\mathbf{T}_L}{\mathbf{R} \times \mathbf{E}} \quad (1)$$

where \mathbf{G}_s is the required gear shaft speed, \mathbf{R} is the gear ratio, \mathbf{T}_L is the gear shaft load, and \mathbf{E} is the efficiency of the gearbox.

The actual torque that a motor can apply is not a given specification. It must be calculated based on various other motor specifications, including the required motor speed, \mathbf{S} . Thus, the torque that a motor will actually apply is given as

$$\mathbf{T}_{\text{cont}} = \sqrt{\frac{155 - \mathbf{T}_{\text{amb}}}{\mathbf{TPR}} - \frac{\mathbf{T}_m \times \mathbf{S}}{\mathbf{C}}} \times \mathbf{PKO} \times \mathbf{K} - \mathbf{T}_m \quad (2)$$

where \mathbf{T}_{amb} is the ambient temperature, \mathbf{T}_m is the motor friction, \mathbf{TPR} is the thermal impedance, \mathbf{PKO} is the motor constant, \mathbf{C} is the rationalization constant, and \mathbf{R} is the thermal derating factor. Thus, this equation represents the total torque minus the torque due to friction. Intuitively, this equation makes sense. The higher the operating temperature, the more heat the motor needs to reject. This obviously results in decreasing continuous torque. Similarly, the faster the motor rotates, the less continuous torque the motor is able to apply.

If the continuous motor torque, \mathbf{T}_{cont} , is lower than the required motor torque, \mathbf{T} , then the motor may automatically be eliminated. If $\mathbf{T}_{\text{cont}} > \mathbf{T}$, then the motor will be able to exert enough torque to rotate the joint at the given speed, and the analysis continues. The next step is to select a motor winding. This is done by calculating the desired torque constant for the given motor. The desired torque constant is given by

$$K_T = \frac{E}{\frac{T_L + T_m}{PKO^2} + \frac{S}{K}} \quad (3)$$

where E is the power source. The torque constant, K_T , varies with different windings. Thus, once the desired K_T has been calculated, the winding with the closest K_T may be chosen.

It is probable that the motor winding used has a slightly different torque constant than the one calculated. Thus, it is possible that this motor will require more than the given power supply, E , in order to operate as desired. Thus, we must double check what the exact voltage is under the desired requirements. If the calculated voltage, E_C , is less than the given voltage, E , then everything is okay. The calculated voltage is given by

$$E_C = \frac{T_L + T_m}{K_T} \times R_T + K_E \times \omega \quad (4)$$

where R_T is the terminal resistance, K_E is the back emf constant, and w is the motor speed. If one is limited in terms of the current that the motor driver can handle, it may be advantageous to calculate the current required by the motor. The current is simply expressed as the first component of the voltage equation. Thus

$$I = \frac{T_L + T_m}{K_T} \quad (5)$$

Since the current is a function of the torque constant, choosing a higher winding reduces the current used by the motor. This may be an alternative if the current needed by a given winding exceeds the maximum output of the motor driver.

However, before this sizing process could begin, it is necessary to decide on the types of gearboxes that were to be used. Pittman has four types of gearboxes:

- spur gearboxes (regular) - max. torque output of 150 oz-in
- spur gearboxes (extra-strength gears) - max. torque output of 175 oz-in
- spur gearboxes (wide-face gears) - max. torque output of 500 oz-in
- planetary gearboxes- max. torque output of 2000 oz-in

Thus, for the base, shoulder, and elbow motors, the planetary gearboxes were selected. For the other three motors, the spur gearboxes with wide-face gears were selected to insure proper performance.

In order to select the gearbox ratios, and the appropriate motor models, spreadsheets were created to compute the aforementioned equations. These specifications were calculated for 12 motors, from two Pittman motor series (14000 and 9000 series), each with five

different gearbox ratios, for a total of 60 possibilities (Appendix B). From these spreadsheets, the appropriate motor/gearbox pairs were selected. The final motor/gearbox selections are as follows:

Table 6: Final motor/gearbox selections

<i>Joint</i>	<i>Motor #</i>	<i>Torque (req.)</i>	<i>Torque (cont.)</i>	<i>Safety Fac.</i>	<i>Winding</i>	<i>Battery</i>	<i>Gearbox Type</i>	<i>Ratio</i>
base	GM14603	12.49 oz-in	17.94 oz-in	1.44	19 V	24 V	Planetary	325.47:1
shoulder	GM14603	12.49 oz-in	17.94 oz-in	1.44	19 V	24 V	Planetary	325.47:1
elbow	GM14603	7.49 oz-in	17.94 oz-in	2.40	19 V	24 V	Planetary	325.47:1
pitch	GM9413	1.16 oz-in	2.87 oz-in	2.47	24 V	12 V	Spur (wide-face)	218.42:1
roll	GM9413	1.16 oz-in	2.87 oz-in	2.47	24 V	12 V	Spur (wide-face)	218.42:1
open/close	GM9413	1.16 oz-in	2.87 oz-in	2.47	24 V	12 V	Spur (wide-face)	218.42:1

Although slightly smaller motors could have been used for the base and the elbow, the same motors were used as the shoulder joint to save costs. This simply has the effect of increasing the factor of safety. Since most standard motorized wheelchairs are powered by two 12-volt batteries, it is possible to use either 12 or 24 volt motors.

Clutch Selection and Sizing

In order to prevent injury or damage due to collisions of the robot arms, clutches are necessary. The function of the clutch is to allow the motor to rotate freely, without rotating the output gear shaft, or the transmission element coupled to the output gear shaft. Although the motor rotates, the corresponding joint does not. There were many considerations in selecting clutches. The only joints necessitating the clutches are the base, shoulder, and elbow. Clutches for the pitch, roll, and open/close joints are not necessary. Since all motors are mounted in the base, within a limited area, the clutches must be as small as possible. They must be able to re-engage automatically after disengaging, if the torque is alleviated. This allows for simple use, and eliminates the need for an assistant during robot use. If possible, the clutches should allow a wide variety of transmission elements, such as pulleys, gears, or sprockets, to be used with the clutch. Furthermore, they must be easily connected to the output gear shaft of each motor.

In consideration of these issues, Dalton torque limiters were selected. The Dalton torque limiters satisfy all of the considerations listed above. The torque limiters are comprised of two friction plates, and a sleeve. The sleeve slides over the output gear shaft, and is secured to the shaft with two set screws. Over the sleeve, a gear fits between the two friction plates. The amount of friction (torque) applied by the plates may be adjusted with a torque wrench. When the torque on the gear exceeds the torque applied by the friction plates, the clutch disengages, and the gear spins around the sleeve freely. As soon as the torque drops below the maximum value, the friction plates engage once again, coupling the motor with the joint.

The torque limiter specifications are as follows:

Table 7: Torque limiter specifications

Joint	Dalton Model	Torque (max.)	Length	Bore
base	OSD-131D	2304 oz-in	1.275 in	.315 in
shoulder	OSD-131D	2304 oz-in	1.275 in	.315 in
elbow	OSD-131	1652 oz-in	1.275 in	.315 in

Theoretical Velocity Analysis for a General 6 Degree of Freedom Manipulator

The goal of the velocity analysis for any manipulator is to express the end effector velocities in the x, y, and z directions in terms of the joint angle velocities, and vice-versa. We may express the end effector velocity in terms of the joint velocities as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dots \end{bmatrix} = \mathbf{J} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dots \end{bmatrix} \quad (6)$$

where the matrix, J, is called the Jacobian matrix. Thus, to present the joint velocities in terms of the end effector velocities, we may simply invert the Jacobian, rewriting the above equation as

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dots \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dots \end{bmatrix} \quad (7)$$

These end effector velocities are usually specified in the end effector reference frame, or in the base reference frame. For a six degree of freedom robot, this would create an extremely complicated Jacobian matrix. Thus, it is to our advantage to calculate the Jacobian in some intermediate reference frame, such as reference frame 3. The Jacobian, in reference frame 3, is expressed as

$$J = \begin{bmatrix} {}^3u_1 & {}^3u_2 & {}^3u_3 & {}^3u_4 & {}^3u_5 & {}^3u_6 \\ {}^3\rho_1 \times {}^3u_1 & {}^3\rho_2 \times {}^3u_2 & {}^3\rho_3 \times {}^3u_3 & {}^3\rho_4 \times {}^3u_4 & {}^3\rho_5 \times {}^3u_5 & {}^3\rho_6 \times {}^3u_6 \end{bmatrix} \quad (8)$$

where

$$\begin{aligned} {}^3u_1 &= V_3^T U_3^T V_2^T U_2^T V_1^T U_1^T [0 \ 0 \ 1]^T & {}^3\rho_1 &= -V_3^T U_3^T V_2^T U_2^T V_1^T s_1 - V_3^T U_3^T V_2^T s_2 - V_3^T s_3 \\ {}^3u_2 &= V_3^T U_3^T V_2^T U_2^T [0 \ 0 \ 1]^T & {}^3\rho_2 &= -V_3^T U_3^T V_2^T s_2 - V_3^T s_3 \\ {}^3u_3 &= V_3^T U_3^T [0 \ 0 \ 1]^T & {}^3\rho_3 &= -V_3^T s_3 \\ {}^3u_4 &= [0 \ 0 \ 1]^T & {}^3\rho_4 &= 0 \\ {}^3u_5 &= U_4^T V_4^T [0 \ 0 \ 1]^T & {}^3\rho_5 &= U_4^T s_4 \\ {}^3u_6 &= U_4^T V_4^T U_5^T V_5^T [0 \ 0 \ 1]^T & {}^3\rho_6 &= U_4^T s_4 + U_4^T V_4^T U_5^T s_5 \end{aligned} \quad (9)$$

U_i , V_i , and s_i may be obtained from the D-H Parameters $(\theta_i, \alpha_i, d_i, a_i)$, where

$$U_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 \\ s\theta_i & c\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad V_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i \\ 0 & s\alpha_i & c\alpha_i \end{bmatrix} \quad s_i = \begin{bmatrix} a_i \\ 0 \\ d_i \end{bmatrix} \quad (10)$$

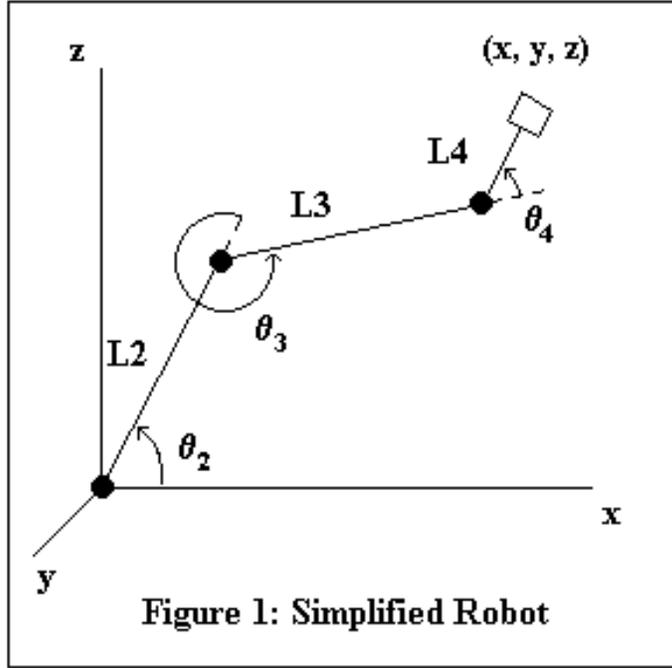


Figure 78: Simplified robot

Conversion of X, Y, Z, Pitch, and Roll Velocities into Motor Velocities for Our Manipulator

It is evident that this can be quite a complicated problem. The kinematics of our 5 degree of freedom robot enable us to solve for the Jacobian matrix in a much simpler fashion. Our robot may be broken up into two parts. The first part (Figure 78) consists of the base, the upper arm (L2), the lower arm (L3), and the gripper (L4). With the use of a gear differential, the gripper pitch and roll are decoupled and can thus be analyzed separately. Thus, from Figure 1, the position of the end effector, (x, y, z), can be written as

$$\begin{aligned}
 x &= [L2\cos(\theta_2)+L3\cos(\theta_2+\theta_3)+L4\cos(\theta_2+\theta_3+\theta_4)]\cos(\theta_1) \\
 y &= [L2\cos(\theta_2)+L3\cos(\theta_2+\theta_3)+L4\cos(\theta_2+\theta_3+\theta_4)]\sin(\theta_1) \\
 z &= L2\sin(\theta_2)+L3\sin(\theta_2+\theta_3)+L4\sin(\theta_2+\theta_3+\theta_4) \\
 \textit{pitch} &= \theta_2+\theta_3+\theta_4 \\
 \textit{roll} &= \theta_5
 \end{aligned} \tag{11}$$

where θ_1 is a rotation of the linkage about the z axis, and θ_5 is rotation of the gripper about its own axis. We may then take the derivative of these five equations in order to get the end effector velocity in the x, y, and z directions, as well as the pitch and roll velocities. Taking the derivatives, and separating the terms,

$$\begin{aligned}
 \mathbf{dx} &= [-L2\cos(\theta_2)\sin(\theta_1) - L3\cos(\theta_2+\theta_3)\sin(\theta_1) - L4\cos(\theta_2+\theta_3+\theta_4)\sin(\theta_1)]\mathbf{dq}_1 + \\
 &\quad [L2\cos(\theta_1)\sin(\theta_2) + L3\cos(\theta_1)\sin(\theta_2+\theta_3) + L4\cos(\theta_1)\sin(\theta_2+\theta_3+\theta_4)]\mathbf{dq}_2 +
 \end{aligned}$$

$$\begin{aligned}
& [L3\cos(\theta_1)\sin(\theta_2+\theta_3) + L4\cos(\theta_1)\sin(\theta_2+\theta_3+\theta_4)]\mathbf{dq}_3 + \\
& [L4\cos(\theta_1)\sin(\theta_2+\theta_3+\theta_4)]\mathbf{dq}_4
\end{aligned} \tag{12}$$

$$\begin{aligned}
\mathbf{dy} = & [L2\cos(\theta_2)\cos(\theta_1) + L3\cos(\theta_2+\theta_3)\cos(\theta_1) + L4\cos(\theta_2+\theta_3+\theta_4)\cos(\theta_1)]\mathbf{dq}_1 + \\
& [L2\sin(\theta_1)\sin(\theta_2) + L3\sin(\theta_1)\sin(\theta_2+\theta_3) + L4\sin(\theta_1)\sin(\theta_2+\theta_3+\theta_4)]\mathbf{dq}_2 + \\
& [L3\sin(\theta_1)\sin(\theta_2+\theta_3) + L4\sin(\theta_1)\sin(\theta_2+\theta_3+\theta_4)]\mathbf{dq}_3 + \\
& [L4\sin(\theta_1)\sin(\theta_2+\theta_3+\theta_4)]\mathbf{dq}_4
\end{aligned} \tag{13}$$

$$\begin{aligned}
\mathbf{dz} = & [L2\cos(\theta_2) + L3\cos(\theta_2+\theta_3) + L4\cos(\theta_2+\theta_3+\theta_4)]\mathbf{dq}_2 + \\
& [L3\cos(\theta_2+\theta_3) + L4\cos(\theta_2+\theta_3+\theta_4)]\mathbf{dq}_3 + [L4\cos(\theta_2+\theta_3+\theta_4)]\mathbf{dq}_4
\end{aligned} \tag{14}$$

$$\mathbf{dpitch} = \mathbf{dq}_3 + \mathbf{dq}_4 + \mathbf{dq}_5 \tag{15}$$

$$\mathbf{droll} = \mathbf{dq}_5$$

Thus, these equations can be easily represented in matrix form in order to obtain the Jacobian and its inverse (Appendix C).

The joint angle velocities are not the velocities that the motors need to obtain to achieve some Cartesian velocity. It is necessary to convert the joint angle velocities into actual motor velocities (see inverse kinematics, section C). The matrix which relates the motor velocities to the joint angle velocities is simply made up of gear ratios, and mechanical couplings, and is expressed as

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \\ \dot{\phi}_5 \end{bmatrix} = \begin{bmatrix} r1 & 0 & 0 & 0 & 0 \\ 0 & r2 & 0 & 0 & 0 \\ 0 & r3 & r3 & 0 & 0 \\ 0 & r4 & r4 & r4 & r4 \\ 0 & r5 & r5 & r5 & -r5 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \end{bmatrix} \tag{16}$$

where $\mathbf{r1}$, $\mathbf{r2}$, $\mathbf{r3}$, $\mathbf{r4}$, and $\mathbf{r5}$ are the total gear ratios between the joints and the respective motors. Since both pitch and roll are being operated through a differential, $\mathbf{r4} = \mathbf{r5}$.

Computer Interface Design

Phase two of the computer interface design involved implementing the hardware selection and method of real time control decided upon in Phase 1. The focus of the second part of the project is to optimize the capabilities of the chip to our design specifications and order any parts that are necessary for the final design or testing. We also took the first step towards interfacing Cooper Union's work on the user interface.

A) Hardware:

The first step of this section is to order the digital I/O board, cabling, breakout board and chips specified in the parts list. Scott Doyle, an Electronics Engineer at PENN designed a inexpensive motor control circuitry board that contained the LM629 and motor driver chip. By ordering large quantities of the boards and the chips, the cost of all the components is greatly reduced. Mr. Doyle's board contains easy to snap on connections with labels, which made assembly easier than soldering parts. The board also allowed our setup to be mobile. The board itself consisted of 16 connections to interface with the breakout board and the LM629 chip, a motor driver, 8 MHz crystal oscillating clock, and connections for the motor and power supplies (5 and 24 volts DC).

At the Cooper Union meeting, the group determined that six motors are necessary to control the base, shoulder, elbow, pitch, roll, and gripper. For each motor we needed one setup consisting of a breakout board, completely wired motion controller board and a motor with a power supply connection. Each of these setups are connected to the 196 bit digital I/O board, which contains 8 ports to control 8 different components. A complete wiring diagram is outlined in the appendix. Each port on the I/O board has a computer base address associated with it that we referenced to in our C code to write and receive code from the LM629 chip. Once the two setups were completed, we rewrote the code to control more than one motor independently.

B) Code:

Writing code for the project experienced an evolution of its own as Cooper Union's joystick code was interface with the motor control code and the capabilities of the robot were redefined. In the original specifications the motor code was written to send and receive position data from the motor. This was accomplished by writing the code to receive encoder values for the motor, based on user inputted theta values. The final code in position mode was able to accept independent angle values for each motor and complete each trajectory independently. Before writing the code to control more than motor we worked with the Cooper Union team to control one motor with their joystick and the motor control code. We were able to finish the code in time for the March meeting at the University of Pennsylvania. After the March meeting, it was determined that associating a velocity with the joystick's position and executing inverse kinematics matrix to generate relevant angular velocity values for each motor would increase the user friendliness of the robot. In the velocity mode if the user moves the joystick to one extreme in the x direction he/she will generate a large angular velocity value thus completing the movement faster. Conversely if the user is moving the robot and would like to stop instantaneously in position mode, the motors would continue to move at the same velocity until the trajectory is complete. In the velocity mode, if the user re-centers the joystick, the velocity will equal zero and the code will stop the motion in a smooth and controlled manner.

C) Inverse Kinematics:

Due to the configuration of the final mechanical design it became apparent that we needed to develop inverse kinematics code to generate angular velocities for each motor when a

particular function is requested from the user. The inverse kinematics matrix takes in previous motor angle values from the encoder of each motor and desired velocity data from the joystick. The desired velocity data is calculated by associating a velocity with the ratio of the change in position to the total range in the x or y direction calculated from the joystick.

$$\text{Ratio} = (\text{Change in the } x \text{ or } y \text{ direction}) / (\text{Total range in the } x \text{ or } y \text{ direction}) \quad (17)$$

The ratio for each x and y joystick movement is calculated to determine an accurate range of motion of the coordinates, which are based on the user's calibration. This method is more effective than inputting a numerical range for the joystick movements based on the calibration of a particular run. The x and y direction's are broken down into four sections to vary the velocities given to the inverse kinematics during as the joystick is moved further in one direction. After going through the calculations the function outputs an array that contains angular velocity data for each motor. The velocity data from the joystick is generated. The maximum velocity is based on the design specifications for the maximum speed of .2 m/s or 7.87 encoder counts per second. The velocity values for each motor are input into a function to start each motor with the correct velocity. The final code consists of joystick functions, inverse kinematics calculations and motor control code with several functions to utilize the capabilities of the Lm629 chip and perform relevant algorithms.

D. Design of User Interface (The Cooper Union)

Simulation

A mechanical simulation program for the wheelchair-based robotic arm is developed in MATLAB Version 4.2b, because of MATLAB's matrix manipulation and interactive graphical display attributes.

The simulation initially displays side, front and top views of the wheelchair-based robot arm. Initially, the robotic arm is in a fully extended position, consequently, the distance between the shoulder joint and the end effector joint is 1.2 m. A 3D Joystick interface, which detects points clicked within its boundary, is subsequently displayed by pressing "Enter".

In essence, the robotic arm is to be controlled by a 3-D Joystick Interface, which is made up of an xy plane (divided into quadrants by 2 orthogonal green lines), two z-axis toggle-buttons, and an exit button. By clicking on the xy-plane of the 3D Joystick interface, the user establishes the endpoint of a vector emanating from the plane's origin (the intersection of the orthogonal green lines). Such a vector will provide the direction that the end effector must move, for incremental values of x, y, and z and is fed into the Denavit-Hartenberg matrices to update the display as the arm moves in space. The speed of motion

in the xy plane is directly proportional to the length of the vector.

Motion in the z-axis may be achieved by clicking onto the z-axis toggle-buttons on the 3D Joystick Interface, thereby activating the z-axis. Once activated, increments in the negative z-direction are obtained by pressing the left mouse button, and increments in the positive z-direction are obtained by pressing the right mouse button. The program enables the user to choose between big and small increments in the z-direction, by clicking on the "z-fast" and "z-slow" toggle-buttons on the 3D Joystick Interface respectively. The user may stop motion in the z-direction, by clicking onto any of the z-axis toggle-buttons. If the user moves the robotic arm to a point that the arm is not capable of reaching (a point located farther than 1.2 m from its shoulder joint), the user will be informed that the robotic arm is fully extended and cannot reach this particular position. Therefore, the aforementioned simulation enables the designer to determine the work envelope of the wheelchair-based robotic arm by providing animated graphical output of its front, side, and top view.

Joystick Interface

The Game Port

IBM defined the game port as it exists on IBM PCs and compatibles. Their definition includes a connector and a PC bus interface. Several PC and compatible product vendors provide game port hardware such as multi-I/O cards, sound cards, and game cards.

Devices as joysticks or game pads are generally used to control the game port hardware, and are used in conjunction with a wide diversity of software such as games or amusement programs in general. Joysticks or game pads plug to a 15-pin female connector which have a predetermined configuration, and pass all circuit information onto the game port PC bus as illustrated in Figure 79.

Gameport 15-pin Female Connector

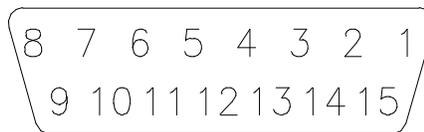


Figure 79: D15 Connector

The electronic assignments of the D15 connector are shown below with respect to each pin number:

Table 8: Electronic assignments of the D15 connector

<u>Pin #</u>	<u>Standard Joystick</u>
--------------	--------------------------

1	+5V
2	Joystick A, Button #1
3	Joystick A, X-Axis
4	Ground
5	Ground
6	Joystick A, Y-Axis
7	Joystick A, Button #2
8	+5V
9	+5V
10	Joystick B, Button #1
11	Joystick B, X-Axis
12	Ground
13	Joystick B, Y-Axis
14	Joystick B, Button #2
15	+5V

The portion of the game port interface of interest to our project is the PC bus interface. The game port appears to the PC bus as a single “read and write” location at I/O address 201 hex, and is 8 bits wide as shown in Figure 80. Bit 0 through bit 3 represent the axes. Bit 4 through bit 7 represent buttons. Button positions can be determined at any time by reading location 201 hex and testing bits 4,5,6, and 7 for high or low logic states. A high state being pressed and a low being not pressed.

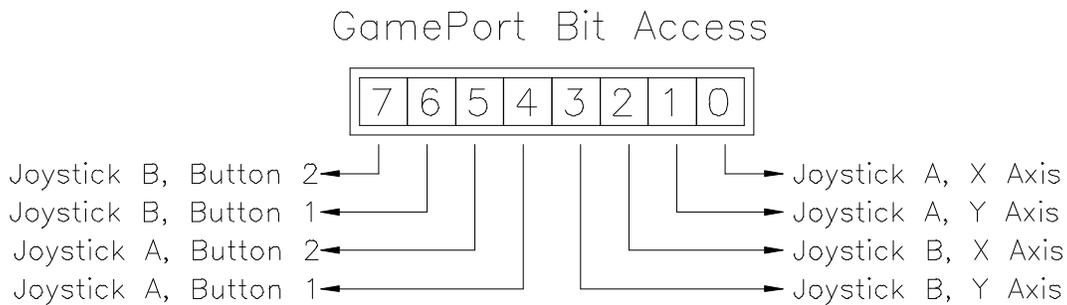


Figure 80: Game port - PC bus

Reading game port axis information is more complex. Within the game port circuitry exist “one-shot” timers (flip-flop type logic device with a resistor & capacitor time out circuit) for each of the four axes. Since joysticks and other positional input devices are resistive transducers, varying an axis position causes a varying resistance. Therefore, the game port connects to one end of this varying resistance. The joystick axis resistance determines the charge-up time of the capacitor. As shown in Figure 81, the axis bits are read as high or set (1) when the capacitors are charging up, and the bits go low or clear (0) when the capacitor charges to a predetermined level. Consequently, game port axis information is represented by capacitor charge up time, and is determined by reading the axis timer status bits at address 201 hex. Thus, the high or low times of the axis bits directly represent the

axis resistance or position. The game port can output axis position information into a program by repetitively looping and reading the axis bits while at high state, and counting loop iterations. The higher the loop count, the higher the axis resistance. Note that a single dummy bit must be written to the game port (write any value to address 201 hex, 0 value recommended) in order to start a time-out cycle.

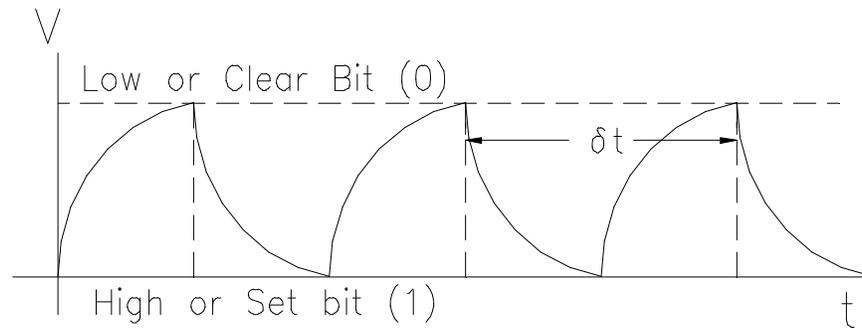


Figure 81: Joystick axis bit definition

A typical game port may have a “time-out” period of approximately between 1.3 and 2.0 milliseconds for an axis resistance of 100k ohms. A disconnected or defective game port would have a much greater or infinite time out period. The programmer should take this into account and limit the maximum number of axis read loop iterations to take no more than several milliseconds. Generally, if a “time-out” takes too long, zero should be returned. Looping should not continue as the computer will appear hung.

The Joystick Working Space

Based on the fact that the conventional game port is an 8-bit wide port, and responds to two XY-axis joysticks, each with two buttons, as shown in Figure 80, it is intended to make good use of all of these I/O available locations for the controlling of the robotic arm. In fact, the main positive aspect of the joystick interface hardware is the advantageous use of a single joystick that supports three of the four axis, and all of the four buttons with which the game-port can interact. However, the crucial task has been to assign these controlling capabilities to the robotic arm in a smart and effective way.

First of all, it is necessary to determine the number of actual elements that the joystick was going to control. Evidently, the minimum number of elements to be control would be six: one for each degree of freedom. Since it is implied that each servomotor corresponds to a single degree of freedom, in effect, the six functions to control are the X, Y, and Z axes, pitch, roll and gripper. Taking into account motion efficiency, it was determined that inverse kinematics would be used to control the XYZ-displacement. In the mean time, the remaining functions are controlled directly from the joystick.

Mainly, the joystick controlling code is embedded in the main user interface C program. It consists of a continuous loop that accesses the game port every time it goes around, getting the complete status information of all eight bits. Meanwhile, it exchanges

information with the I/O board, which at the same time communicates with the chip controller that controls the motion of each individual motor. Initially, after the program has initialized all the circuitry and chips, it goes on with the calibration of the joystick. The calibration of the joystick is an essential stage of the user interface because it defines the working space of the joystick according to the user's criteria. After the axes in the game port have been detected, the user is prompted to pull the joystick to the upper left corner and to press any button while in that position in order to record the lowest values for X and Y axis. Subsequently, the user is asked to move the joystick to the lower right corner to record the highest positions of both axes. Finally, in order to record the resting position of the joystick, the user is asked push any button while in the center location. Therefore, in this manner the joystick will respond accordingly to this working space as shown in Figure 82.

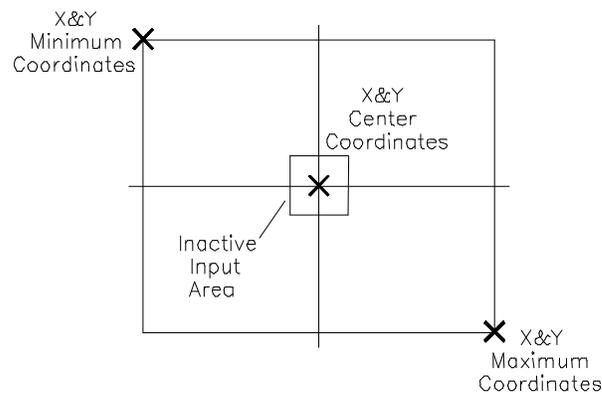


Figure 82: Joystick working area and calibration parameters

Once the joystick calibration is complete, the program enters into the controlling mode, as any variation in the position of the joystick or the status of the buttons originates motion in the arm. The joystick is considerably sensitive in general, but it is mostly axis sensitive. That is the reason why it is necessary to allow some neutral area around the resting position of the joystick, so it will not cause any motion in the arm from small deviations from center. Indeed, whenever the joystick is moved out of the neutral area, it automatically creates an imaginary vector in the X-Y plane that is inputted into the controlling program. In the case when the Z-axis also gets activated, an imaginary three-dimensional vector r is created proportional to the deviation of the joystick from the center position, as shown in Figure 83.

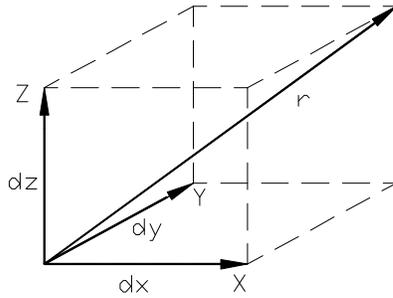


Figure 83: Imaginary three-dimensional input vector

Thus dx , dy , and dz go directly into the inverse kinematics, which outputs a set of velocities proportional to the magnitude of each of them. The goal of the inverse kinematics is to convert the imaginary vector r into a set of velocities that would originate motion in a rectilinear form and with the same orientation as r . Upon the accomplishment of motion, a new cycle may begin according to the position of the joystick.

The Joystick

The joystick used for the controlling of the robotic arm is the CompUSA joystick controller. It has a wide number of advantages that make it appropriate for this project. Among the most significant features of this joystick is the fact that it has the capacity to control all four buttons in the game port. These buttons are conveniently located around

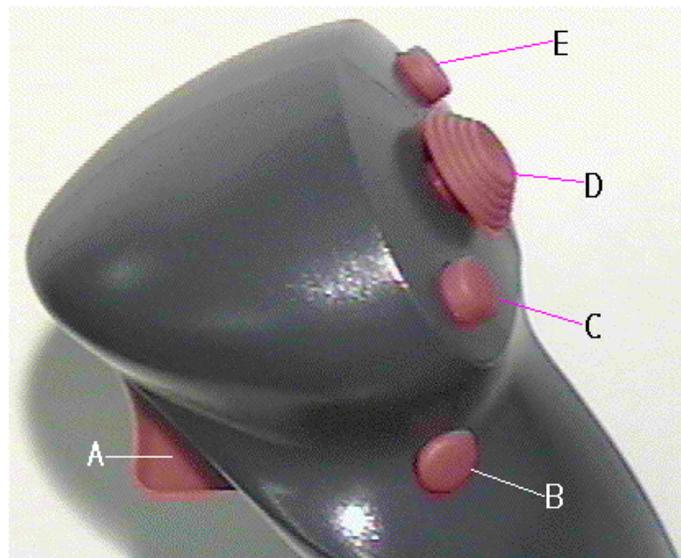


Figure 84: CompUSA joystick controlling elements

the body of the joystick, which makes it a lot easier to push than if all of them were at the base of the joystick. In addition, this joystick is equipped with a third axis that is located at the head of the joystick. Figure 84 shows the labeling of the joystick buttons. A, B, C and E are buttons and D is the extra axis, which corresponds to the Y-axis of joystick B in the game port.

In order to make the joystick interface user friendly, the X and Y axis motion of the robotic arm are to be controlled by the joystick's XY-axis controlling feature, and will respond to any slight motion of the joystick shaft. A neutral area is defined in the central working joystick space, which prevents any motion while the joystick moves within it, as illustrated in Figure 82. In addition, three of the four buttons are assigned to the three functions of pitch, roll, and gripper. Namely, the buttons assigned to these three motion functions are B, C, and D. Likewise, the D label has been assigned to control the Z-axis motion though the third axis controller in the joystick. Furthermore, given that the assignments of B, C, D, and E are exclusive for those controlling functions, and that they are permanent, the pushing of any of those will actuate that corresponding degree of freedom in one direction. Thus, the opposite direction will be actuated by pressing button A at any time. In other words, A is the element in charge of reversing the motion of any of those four degrees of freedom. Button A will not do anything by itself unless it is pressed in combination with B, C, D, or E.

The efficiency and reliability of the joystick interface is improved by the fact that the computer is able to run the main loop of the C user interface program at a high speed, relative to the speed expected from the user. Finally, it can be concluded that in addition to the main features of the joystick interface explained above, a general crucial advantage of it is that at no time will the user be unable to control any of the degrees of freedom of the arm. Given that the joystick code is able to read the entire status of the joystick port at all times, all of these input functions can constantly be translated into motion of the arm on a one-shot basis. And yet, there is not complexity in the controlling of the arm as far as joystick interface is concerned. The actuation of each of the six degrees of freedom of the robotic arm is done by merely pressing two buttons at most, or by just moving the joystick shaft in the desired direction that the user wants the arm to move.

E. The Final Product

The schools involved in the 1997 Gateway Coalition Project convened on May 17 at Drexel University to give the final presentation to members of the National Science Foundation. Representatives from the participating schools are shown in the figure below.



Figure 85: Gateway Coalition student representatives

The presentation included contributing pieces from each school a video of the finished product (seen below) in use.

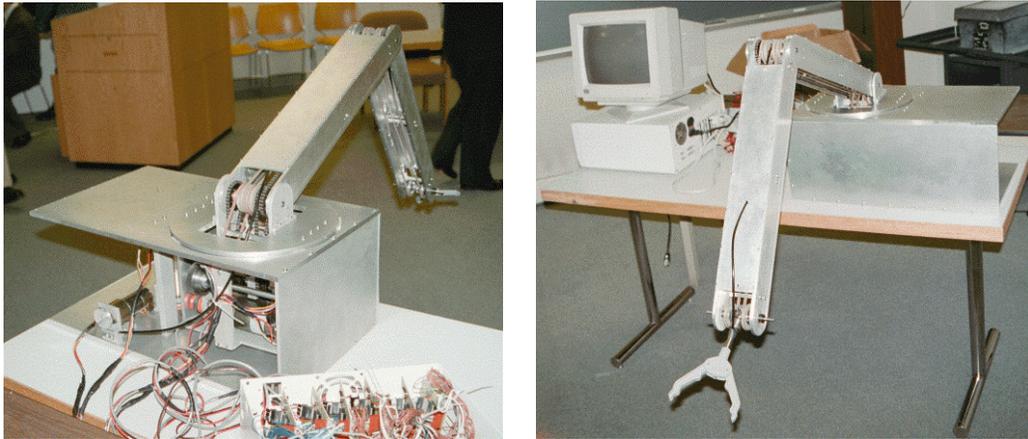


Figure 86: Final product - a) rear view of robotic arm and b) front view of robotic arm

VIII. PROJECT LOGISTICS

Communication

The nature of this design and manufacturing project constitutes massive and detailed communication, considering that five different universities, separated geographically, must complete individual aspects of the project, yet work as a single unit to accomplish one common goal. Communication between schools has been and continues to be one of the major concerns when considering the restraints on the completion of the project.

Communication between schools was attempted and was moderately successful using several different methods. Messages and ideas were shared through video-conferencing, the worldwide web, e-mail, telephone calls, and facsimiles.

Videoconferences using computers were one of the most successful forms of discussion. All team members could be seen and heard, as opposed to a closed, one on one phone conversation. As the Gateway teams quickly discovered, everyone could not be heard at once, as the transfer of sound by videoconference is very sensitive to interference. The most successful conferences were achieved by assigning one particular person from one particular school to host or control the conference. Activating the “mute” button when another party is speaking also seemed to be successful in minimizing interference. Drexel and Ohio State conferenced at least once a week, and were often joined by University of Pennsylvania, as each of these school’s responsibilities were interrelated. Cooper Union and N.J.I.T. joined in on a few as well. In a couple instances, all five Gateway schools conferenced at the same time. This did not go as well as hoped and resulted in a lot of dead time for the individual



Figure 87: Ohio State and Drexel in a video-conference

schools at times, but it allowed everyone to be involved and interject in the discussion. The main problem with getting all of the schools together simultaneously was obviously satisfying schedules so that all or most members of each team could meet at a specific time each week. The picture above illustrates one such video-conference between Ohio State and Drexel.

The internet and worldwide web were also used quite effectively. Each school developed a web page, and one of the major goals established at the January meeting was to have each school update their page on a weekly basis. Several drawings of parts and concepts were posted for the Gateway Coalition's use of updated information as well as the individual schools. Schedules were also posted as to when each school was available for video - conferencing. A sample of one of the web sites is seen on the in Figure 88.

Messages through e-mail were also used, serving as a method of sending simple information, posing questions and answers, and setting up videoconferences and concurrent meetings. Dimensioned drawings and sketches were also transferred by e-mail through IGES files.



Figure 88: Drexel University's Gateway Web Page

Telephone calls were an obvious method to get in touch with one another, proving to be more effective than e-mail at times and less effective at others. This depended on whether or not the requested party was available or near his or her phone. Fax machines proved to be useful in sending drawing printouts and part lists and numbers. Phones and fax machines were also the main method by which parts and materials were inquired and ordered.

In conclusion, it is obvious that the communication of ideas was one of the most important and demanding aspects of this project. Time and patience are vital, as well as having an open mind to alternative ideas and suggestions. Communication was given an honest attempt, but needs to be further assessed in future projects. In deciding upon task responsibilities in the future, the Gateway coalition may want to consider assigning the most closely related tasks to the schools in close proximity to one another. The communication aspect of the project was a helpful learning experience and allowed for engineering students from some of the finest colleges and universities in the country to share ideas and contribute to such a worthy project.

IX. CONCLUDING REMARKS

Problems and Future Recommendations

After the recent assembly of the robotic manipulator, it has become clear to the Gateway Team that many improvements can be made to streamline the design, manufacture, and assembly of the A.R.M.

Beginning at the end point, the gripper itself is very efficient in both cost and function. However, the attachment of the gripper to the tube on the differential presented a difficult problem. The gripper was initially designed to press fit onto the end of the differential which would have provided the gripper's pitch and roll movements. With the simple press fit design, it was found that while trying to lift heavy objects the gripper would actually slip due to a poor press fit. In order to prevent this from happening a hole was drilled through the gripper end and the tube of the differential. Taking care not to disrupt the spring in the gripper end, a cotter pin was then placed through these holes. This made for a secure way of fastening the gripper temporarily, but needs to be accounted for in the redesign. Possibilities include a bonding agent or an integral fastening system designed into the differential.

The differential should also be a major target for improvements. From an engineering standpoint, the differential is excellent in that it provides the pitch and roll motions in a single package that can be ordered from the supplier ready to assemble. On the other hand, at a price of \$450.00, the cost of the differential makes it a very unattractive option, economically. The reason for the high price is the special tube that the supplier places on the differential to accommodate the gripper. This price can be eliminated by either ordering in larger quantities or doing the modifications in-house. The latter of these two options should be investigated. Another option would be to optimize the original design, which is much less costly but would add greatly to part count and assembly time.

A great deal of difficulty was encountered with the chains that are used to actuate the sprockets that drive the differential. Cable chains were selected for pitch and roll because they are lightweight and do not require lubrication. The cable chain was ordered in bulk and was to be cut to the required length and crimped into a loop on sight. It was quickly realized that the tools to aid in the crimping of the cables were insufficient. To begin, it was nearly impossible to thread two cables through the bushing without fraying the cables. If one is successful with this task, the bushings can be crimped to hold the cables together but the tensile strength of the joint is significantly less than the supplier's specifications. The most cost efficient answer to this problem is to order the cable chain to the correct dimensions from the supplier. It is important to note that the chain must have some slack in it to allow the assembler to get the chain around the sprockets. A tensioning system can then take the slack out of the chain. The current tensioning system for the cable chains may be inadequate to take this amount of slack out of the chain. A possible solution to this problem is to use two of the current tensioning systems on each chain rather than just

one. This allows for more slack to be taken out of the chain. From a manufacturability standpoint, off-the-shelf chain tensioning systems are available and may cut down on manufacturing time enough to offset the extra cost.

Similar problems were experienced with the roller chain, which actuated the shoulder and elbow joint. Again, the chain was ordered in bulk lengths and assembled on sight. During the design phase, it was not realized that every other link is different, a sort of male-female relationship, which means the smallest amount of chain that can be removed at one time is a half of an inch, instead of the quarter of an inch assumed in the design. Therefore, the chain was either too loose or not quite long enough. This was overcome by stretching the chain by heating it. In the redesign, the chain itself can not be changed, but the tensioning system could be designed such that it would take up the necessary slack. One concern that must be considered is the reduction of chain life due to the reverse bending from the tensioning system. Additionally, the supplier recommended lubrication after every eight hours of use for long life. These factors may or may not have a significant effect on the life, but nonetheless, should be investigated.

In order to more efficiently build a transmission system from the motor to the joints, a few improvements can be made. First off, based on the current design for the base rotation, the width of the motor housing could be made as large as the length due to the fact that the entire subassembly rotates about the center axis and is hidden within another assembly. Based on this assumption, a gear train would not be necessary and only two gears and one shaft would be needed per shoulder motor and elbow motor. If this improvement is not feasible due to a design change in the base subassembly, then the minimum improvement to be made would be to reduce the shafts to the minimum size needed, especially the last shafts in the gear train. Another improvement to be made is to use lighter and cheaper materials for the gears. A simple way to reduce the weight would be to use spoked or slotted gears. Reducing the hub diameters would also help, as would reducing or eliminating the hubs altogether.

Another area for improvement is in the base rotation. Now that a complete solid model is available, the mass moment of inertia can be found for different positions. Using these numbers a complete torque analysis can be made and a proper motor selected, which should turn out to be smaller than the current motor in use.

X. APPENDIX

A. Sample Calculations

Torque Calculations: (put in Appendix)

For elbow: input torque from motor = 1000 in-oz
Power = constant = 0.0099 hp

$$T_{OUT} = \left(\frac{w_1}{w_2}\right)\left(\frac{w_2}{w_3}\right)\left(\frac{w_3}{w_4}\right)(T_{IN})$$

where w_1 = speed of clutch = 10 rpm
 w_2 = speed of gear 2 = 7.0 rpm
 w_3 = speed of gear 3 = 7.0 rpm
 w_4 = speed of gear 4 = 4.67 rpm

$$T_{OUT} = \left(\frac{10rpm}{4.67rpm}\right)(1000 \text{ in-oz}) = 2143 \text{ in-oz}$$

$$\text{Reduction} = \frac{T_{OUT}}{T_{IN}} = \frac{2143}{1000} = 2.143$$

Gear Tooth Analysis:

Modified Lewis Equation: $S = \frac{W \cdot P}{F \cdot Y} \cdot \left(\frac{600}{600 + V}\right)^{-1}$

$$\text{where } W = \text{tooth load} = \frac{2T}{D} = \frac{2(1000 \text{ in-oz}) / 16 \text{ in} / \text{lb}}{1.75 \text{ in}} = 71.43 \text{ lbs}$$

$$V = \text{pitch line velocity} = 0.262 \times D \times \text{RPM} = 4.59 \text{ ft/min}$$

T = torque (in-lbs)

F = face width = 0.50 in

Y = tooth form factor (for 35 teeth) = 0.373

P = diametral pitch

D = pitch diameter = 1.75 in

S = stress (psi)

$$\text{For elbow clutch to gear 2: } S = \frac{(71.43 \text{ lbs})(20)}{(0.50 \text{ in})(0.373)} \times \left(\frac{600}{600 + 4.59}\right)^{-1} = 12349 \text{ psi}$$

B. Simulation Code

```
% *****
% Manip2.m
% This program will calculate the points
% needed for the animation and animates the
% manipulator given the final coordinates.
% *****
clear;clc;

disp('    Robotic Arm Animation and Transform Matrix Calculation')
disp('        OSU and Cooper Union Gateway Design Group 1997')

fprintf('\n\n\n\n\n\n\n')

% Modifications for figure insertion

figure(1);set(1, 'position', [70 0 650 380], 'menubar', 'none');

% This defines the positions of axis of subplots
% This also sets up the handles
% of the linkages, gripper, and joints within each axes
h1=axes('position',[.6 .15 .38 .35],'box','on','xcolor',[1 1 1],'ycolor',[1 1 1]);
title('Side View','color',[1 1 1],'fontweight','bold');
xlabel('x');
ylabel('z');
axis('equal');
hinge1=line('xdata',[],'ydata',[],'erasemode','none','color',[0 1 0]);
pivot1=line('xdata',[],'ydata',[],'erasemode','none','color',[0 1 0]);
line1a=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[1 0 0]);
line2a=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[1 1 0]);
line3a=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[0 0 1]);
grip1a=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[.49 1 .83]);
grip1b=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[.49 1 .83]);
grip1c=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[.49 1 .83]);
grip1d=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[.49 1 .83]);
jointsa1=line('xdata',[],'ydata',[],'linestyle','o','erase','xor','markersize',5);
jointsa2=line('xdata',[],'ydata',[],'linestyle','o','erase','xor','markersize',5);
jointsa3=line('xdata',[],'ydata',[],'linestyle','o','erase','xor','markersize',5);
jointsa4=line('xdata',[],'ydata',[],'linestyle','o','erase','xor','markersize',5);

h2=axes('position',[.1 .15 .38 .35],'box','on','xcolor',[1 1 1],'ycolor',[1 1 1]);
title('Front View','color',[1 1 1],'fontweight','bold');
xlabel('y');
```

```

ylabel('z');
axis('equal');
hinge2=line('xdata',[],'ydata',[],'erasemode','none','color',[0 1 0]);
pivot2=line('xdata',[],'ydata',[],'erasemode','none','color',[0 1 0]);
line1b=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[1 0 0]);
line2b=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[1 1 0]);
line3b=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[0 0 1]);
grip2a=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[.49 1 .83]);
grip2b=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[.49 1 .83]);
grip2c=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[.49 1 .83]);
grip2d=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[.49 1 .83]);
jointsb1=line('xdata',[],'ydata',[],'linestyle','o','erase','xor','markersize',5);
jointsb2=line('xdata',[],'ydata',[],'linestyle','o','erase','xor','markersize',5);
jointsb3=line('xdata',[],'ydata',[],'linestyle','o','erase','xor','markersize',5);
jointsb4=line('xdata',[],'ydata',[],'linestyle','o','erase','xor','markersize',5);

h3=axes('position',[.1 .6 .38 .35],'box','on','xcolor',[1 1 1],'ycolor',[1 1 1]);
title('Top View','color',[1 1 1],'fontweight','bold');
xlabel('y');
ylabel('x');
axis('equal')
hinge3=line('xdata',[],'ydata',[],'erasemode','none','color',[0 1 0]);
pivot3=line('xdata',[],'ydata',[],'erasemode','none','color',[0 1 0]);
line1c=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[1 0 0]);
line2c=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[1 1 0]);
line3c=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[0 0 1]);
grip3a=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[.49 1 .83]);
grip3b=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[.49 1 .83]);
grip3c=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[.49 1 .83]);
grip3d=line('xdata',[],'ydata',[],'linewidth',2,'erasemode','xor','color',[.49 1 .83]);
jointsc1=line('xdata',[],'ydata',[],'linestyle','o','erase','xor','markersize',5);
jointsc2=line('xdata',[],'ydata',[],'linestyle','o','erase','xor','markersize',5);
jointsc3=line('xdata',[],'ydata',[],'linestyle','o','erase','xor','markersize',5);
jointsc4=line('xdata',[],'ydata',[],'linestyle','o','erase','xor','markersize',5);

%Set the axis limits for each subplot
axes(h1);
axis([-1.6 1.6 -1.6 1.6]);
axes(h2);
axis([-1.6 1.6 -1.6 1.6]);
axes(h3);
axis([-1.6 1.6 -1.6 1.6]);

%set(hinge1,'xdata',[.6],'ydata',[0])

```

```

set(jointsa1,'xdata',.6,'ydata',0)
set(line1a,'xdata',[0 .6],'ydata',[0 0])
%set(hinge1,'xdata',[1.2],'ydata',[0])
set(jointsa2,'xdata',1.2,'ydata',0)
set(line2a,'xdata',[.6 1.2],'ydata',[0 0])
%set(hinge1,'xdata',[1.4],'ydata',[0])
set(jointsa3,'xdata',1.4,'ydata',0)
set(line3a,'xdata',[1.2 1.4],'ydata',[0 0])
set(grip1a,'xdata',[1.4 1.5],'ydata',[0 0])
set(jointsa4,'xdata',1.5,'ydata',0)
set(grip1b,'xdata',[1.5 1.5],'ydata',[.1 -.1])
set(grip1c,'xdata',[1.5 1.6],'ydata',[.1 .1])
set(grip1d,'xdata',[1.5 1.6],'ydata',[-.1 -.1])

```

```

%set(hinge2,'xdata',[0],'ydata',[0])
set(jointsb1,'xdata',0,'ydata',0)
set(line1b,'xdata',[0 0],'ydata',[0 0])
%set(hinge2,'xdata',[0],'ydata',[0])
set(jointsb2,'xdata',0,'ydata',0)
set(line2b,'xdata',[0 0],'ydata',[0 0])
%set(hinge2,'xdata',[0],'ydata',[0])
set(jointsb3,'xdata',0,'ydata',0)
set(line3b,'xdata',[0 0],'ydata',[0 0])
set(grip2a,'xdata',[0 0],'ydata',[0 0])
set(jointsb4,'xdata',0,'ydata',0)
set(grip2b,'xdata',[0 0],'ydata',[.1 -.1])
set(grip2c,'xdata',[0 0],'ydata',[.1 .1])
set(grip2d,'xdata',[0 0],'ydata',[-.1 -.1])

```

```

%set(hinge3,'xdata',[0],'ydata',[.6])
set(jointsc1,'xdata',0,'ydata',.6)
set(line1c,'xdata',[0 0],'ydata',[0 .6])
%set(hinge3,'xdata',[0],'ydata',[1.2])
set(jointsc2,'xdata',0,'ydata',1.2)
set(line2c,'xdata',[0 0],'ydata',[.6 1.2])
%set(hinge3,'xdata',[0],'ydata',[1.4])
set(jointsc3,'xdata',0,'ydata',1.4)
set(line3c,'xdata',[0 0],'ydata',[1.2 1.4])
set(grip3a,'xdata',[0 0],'ydata',[1.4 1.5])
set(jointsc4,'xdata',0,'ydata',1.5)
set(grip3a,'xdata',[0 0],'ydata',[1.5 1.5])
set(grip3a,'xdata',[0 0],'ydata',[1.5 1.6])
set(grip3a,'xdata',[0 0],'ydata',[1.5 1.6])

```

```
drawnow;
```

```
%calculates position and size of hinges
```

```
r=.03; %this changes how big the hinge looks on plot
```

```
x0=0; %this sets the x coordinate of hinge position
```

```
y0=0; %this sets the y coordinate oh hinge position
```

```
ninc=20;
```

```
thta=0;
```

```
npoints=ninc+1;
```

```
coord=circle(r,x0,y0,ninc); %calls circle.m
```

```
for t=1:1:npoints
```

```
    xcoord(t)=coord(t,1);
```

```
    ycoord(t)=coord(t,2);
```

```
end;
```

```
binc=ninc/2;
```

```
coord2=bushing(r,x0,y0,binc,thta); %calls bushing.m
```

```
bpoints=ninc/2+16;
```

```
for t=1:1:bpoints
```

```
    xbush(t)=coord2(t,1);
```

```
    ybush(t)=coord2(t,2);
```

```
end;
```

```
% This is where the hinges are drawn
```

```
set(pivot1,'xdata',xcoord,'ydata',ycoord); %this draws the pivot circle in axes(h1)
```

```
set(hinge1,'xdata',xbush,'ydata',ybush); %this draws everything else axes(h1)
```

```
set(pivot2,'xdata',xcoord,'ydata',ycoord); %this draws the pivot circle axes(h2)
```

```
set(hinge2,'xdata',xbush,'ydata',ybush); %this draws everything else axes(h2)
```

```
set(pivot3,'xdata',xcoord,'ydata',ycoord); %this draws the pivot circle axes(h3)
```

```
set(hinge3,'xdata',xbush,'ydata',ybush); %this draws everything else axes(h3)
```

```
theta = [0,0,0,pi/2,0];
```

```
% For Joint 1:
```

```
alpha1=-pi/2;
```

```
a1=0;
```

```
s1=0;
```

```
% For Joint 2 ( the theta will vary) :
```

```
alpha2=0;
```

```
a2=.6;
```

```
s2=0;
```

% For Joint 3 (the theta will vary) :

```
alpha3=0;  
a3=.6;  
s3=0;
```

```
alpha4=pi/2;  
a4=0;  
s4=0;
```

```
alpha5=0;  
a5=0;  
s5=.2;
```

```
vector1=[0;0;0;1];  
vector2=[0;0;0;1];  
vector3=[0;0;0;1];  
vector4a=[0;0;.1;1];  
vector4b=[.1;0;.1;1];  
vector4c=[-.1;0;.1;1];  
vector4d=[.1;0;.2;1];  
vector4e=[-.1;0;.2;1];  
in=1;  
inp=0;  
t1=0;  
t2=0;  
t3=0;  
t4=0;  
t5=0;  
x=1.2;  
y=0;  
z=0;  
qq=0;  
j=[0,0,0];  
xglob=1.2;  
yglob=0;  
zglob=0;  
oldxglb=1.2;  
oldyglb=0;  
oldzglb=0;  
zstat=-1;  
mag=0;  
zopt=0;  
chk=0;  
stat=0;
```

```

while in>0;
if stat==0;
    stat=1;
    disp(' 0 = quit')
    disp(' 1 = move to new point')
    disp('What do you want to do? (0-1) [1]: ')
    resp=input('*>>');
    fprintf('\n\n\n\n\n\n')
    if resp==0;
        in=0;
    elseif resp==1;
    else;
        resp=1;
    end;
else;
    disp('Pick new motion direction.')
    resp=1;
end;
while resp==1;

% ginput loop

% Joystick Emulator Starts

% inserting another figure
figure(2);set(2, 'position', [0 460 250 180], 'menubar', 'none');
set(2, 'position', [0 480 250 180], 'menubar', 'none');
%clf reset ---- removing the clear figure
axes('position', [.1 .1 .8 .8], 'box','on','xcolor',[1 1 1], 'ycolor', [1 1 1]);
title('3D Joystick Emulator', 'color', [0 1 0], 'fontweight','bold');
xlabel('X', 'fontweight', 'bold')
ylabel('Y', 'fontweight', 'bold')
axis([-1.0 1.0 -1.0 1.0])
hold on

text(1.0,1.0,'*****','color',[1 0 0])
text(1.0,.975,'*','color',[1 0 0])
text(1.0,.95,'*','color',[1 0 0])
text(1.0,.925,'*','color',[1 0 0])
text(1.0,.9,'*','color',[1 0 0])
text(1.0,.875,'*','color',[1 0 0])
text(1.0,.85,'*','color',[1 0 0])
text(1.0,.825,'*','color',[1 0 0])

```

text(1.0,.8,'*****','color',[1 0 0])
text(1.2,1.0,'*','color',[1 0 0])
text(1.2,.975,'*','color',[1 0 0])
text(1.2,.95,'*','color',[1 0 0])
text(1.2,.925,'*','color',[1 0 0])
text(1.2,.9,'*','color',[1 0 0])
text(1.2,.875,'*','color',[1 0 0])
text(1.2,.85,'*','color',[1 0 0])
text(1.2,.825,'*','color',[1 0 0])
text(1.2,.8,'*','color',[1 0 0])
text(1.0,1.15,'Z slw')
text(1.0,.95,'on/off')

text(1.0,0.2,'*****','color',[1 0 0])
text(1.0,.275,'*','color',[1 0 0])
text(1.0,.25,'*','color',[1 0 0])
text(1.0,.225,'*','color',[1 0 0])
text(1.0,.3,'*','color',[1 0 0])
text(1.0,.375,'*','color',[1 0 0])
text(1.0,.35,'*','color',[1 0 0])
text(1.0,.325,'*','color',[1 0 0])
text(1.0,0.4,'*****','color',[1 0 0])
text(1.2,0.2,'*','color',[1 0 0])
text(1.2,.275,'*','color',[1 0 0])
text(1.2,.25,'*','color',[1 0 0])
text(1.2,.225,'*','color',[1 0 0])
text(1.2,.3,'*','color',[1 0 0])
text(1.2,.375,'*','color',[1 0 0])
text(1.2,.35,'*','color',[1 0 0])
text(1.2,.325,'*','color',[1 0 0])
text(1.2,0.4,'*','color',[1 0 0])
text(1.0,.55,'Z fst')
text(1.0,.15,'on/off')

text(1.0,-.8,'*****','color',[1 0 0])
text(1.0,-.825,'*','color',[1 0 0])
text(1.0,-.85,'*','color',[1 0 0])
text(1.0,-.875,'*','color',[1 0 0])
text(1.0,-.9,'*','color',[1 0 0])
text(1.0,-.925,'*','color',[1 0 0])
text(1.0,-.95,'*','color',[1 0 0])
text(1.0,-.975,'*','color',[1 0 0])

```

text(1.0,-1.0,'*****','color',[1 0 0])
text(1.2,-.8,'*','color',[1 0 0])
text(1.2,-.825,'*','color',[1 0 0])
text(1.2,-.85,'*','color',[1 0 0])
text(1.2,-.875,'*','color',[1 0 0])
text(1.2,-.9,'*','color',[1 0 0])
text(1.2,-.925,'*','color',[1 0 0])
text(1.2,-.95,'*','color',[1 0 0])
text(1.2,-.975,'*','color',[1 0 0])
text(1.2,-1.0,'*','color',[1 0 0])
text(1.01,-.877,'EXIT')

```

```

view(0,90)

```

```

xx=-1:1;
yy=(xx*0);
plot(xx,yy,'-');
plot(yy,xx,'-');

```

```

xmou=[];
ymou=[];
znc=0;

```

```

n = 0;
% Loop, picking up the points.
motion = 0;
but = 1
while motion == 0;

```

```

    [xi,yi,but] = ginput(1);

```

```

    disp(xglob)
    disp(yglob)
    disp(zglob)

```

```

    oldxglb=xglob;
    oldyglb=yglob;
    oldzglb=zglob;

```

```

    if but == 1;

```

```

if xi>1.0 & xi<1.2;
    if yi>0.8 & yi<1.0;
        zstat=zstat*(-1);
        zopt=1;
        disp(zopt)
        if zstat==1;
            disp('Z-axis motion is active.')
        else;
            disp('Z-axis motion is off.')
        end;
        elseif yi>0.2 & yi<0.4;
            zstat=zstat*(-1);
            zopt=2;
            disp(zopt);
            if zstat==1;
                disp('Z-axis motion is active.')
            else;
                disp('Z-axis motion is off.')
            end;
            elseif yi>-1 & yi<-0.8;
                chk=1
                motion=1;
                resp=0;
                in=0;
                disp('Exiting motion loop...')
            end;

else;
    if yi<1.0 & yi>-1.0 & xi>-1.0 & xi<1.0;
        xxx = [x; xi];
        yyy = [y; yi];
        plot(xi,yi,'go','era','back')
        n = n + 1;
        %text(xi,yi,[' ' int2str(n)],'era','back');
        fprintf('the value of x');
        disp(xi)
        fprintf('the value of y');
        disp(yi)
        motion = 1
    end;

end;
end;

if but == 2;

```

```

if xi>1.0 & xi<1.2;
    if yi>0.8 & yi<1.0;
        zstat=zstat*(-1);
        zopt=1;
        disp(zopt)
        if zstat==1;
            disp('Z-axis motion is active.')
        else;
            disp('Z-axis motion is off.')
        end;
    elseif yi>0.2 & yi<0.4;
        zstat=zstat*(-1);
        zopt=2;
        disp(zopt);
        if zstat==1;
            disp('Z-axis motion is active.')
        else;
            disp('Z-axis motion is off.')
        end;
    elseif yi>-1 & yi<-0.8;
        chk=1;
        motion=1;
        resp=0;
        in=0;
        disp('Exiting motion loop...')
    end;

else;

if yi<1.0 & yi>-1.0 & xi>-1.0 & xi<1.0;
xxx = [x; xi];
yyy = [y; yi];
plot(xi,yi,'go','era','back')
n = n + 1;
%text(xi,yi,[' ' int2str(n)],'era','back');
fprintf('the value of x');
disp(xi)
fprintf('the value of y');
disp(yi)
motion = 1
end;
end;

```

```

end;
% motion closes
end;
% emulator ends

```

```

ok=1;
while ok==1;
    xmou=xi;
    fprintf('\n\n\n\n\n\n')

    ymou=yi;
    fprintf('\n\n\n\n\n\n')

    if zstat==1 & but==1;
        znc=(0.1*zopt);
    elseif zstat==1 & but==2;
        znc=(-0.1*zopt);
    else;
        znc=0;
    end;

    s=sqrt((xmou^2)+(ymou^2));
    if s>0 & s<=0.4;
        mag=1;
    elseif s>0.4 & s<=0.8;
        mag=2;
    else;
        mag=3;
    end;

    slp=abs(xmou/ymou);

    if xmou>0 & slp>1 & ymou>0;
        xnc=(0.1*mag);
        ync=xnc/slp;
    elseif xmou<0 & slp>1 & ymou<0;
        xnc=(-0.1*mag);
        ync=xnc/slp;
    elseif xmou>0 & slp>1 & ymou<0;
        xnc=(0.1*mag);
        ync=-xnc/slp;
    end;
end;

```

```
elseif xmou<0 & slp>1 & ymou>0;
xnc=(-0.1*mag);
ync=-xnc/slp;
end;
```

```
if xmou>0 & slp<1 & ymou>0;
ync=(0.1*mag);
xnc=ync*slp;
elseif xmou>0 & slp<1 & ymou<0;
ync=(-0.1*mag);
xnc=-ync*slp;
elseif xmou<0 & slp<1 & ymou>0;
ync=(0.1*mag);
xnc=-ync*slp;
elseif xmou<0 & slp<1 & ymou<0;
ync=(-0.1*mag);
xnc=ync*slp;
end;
```

```
if motion==1 & qq==1;
    xglob=xglob+xnc;
    yglob=yglob+ync;
    zglob=zglob+znc;
end;
```

```
if qq==0;
    xglob=x+xnc;
    yglob=y+ync;
    zglob=z+znc;
    qq=1;
end;
```

```
if abs(xglob)>1.2;
xnc=0;
end;
```

```
if abs(yglob)>1.2;
ync=0;
end;
```

```
if abs(zglob)>1.2;
znc=0;
end;
```

```

limit= sqrt((xglob^2)+(yglob^2)+(zglob^2));
if limit<=1.2;
    ok=0;
    resp=0;
    ans='y';

elseif limit>1.2;
    if chk==1;
        fprintf('\n\nShutting program off...\n\n')
        ok=0;
        motion=0;
        ans='n';
    else;
        fprintf('\n\nThose numbers are out of the range of')
        fprintf('\nmovement. Try again.\n')
        x=oldxglb;
        y=oldyglb;
        z=oldzglb;
        disp(limit)
        ok=0;
        motion=0;
        ans='n';
    end;
end;

    % ok closes
end;

% resp closes
end;

if in>0;

xin=xglob;
yin=yglob;
zin=zglob;

limit= sqrt((xin^2)+(yin^2)+(zin^2));
if limit>=1.2;
in=0;
end;

```

```

px=x;
py=y;
pz=z;
for m=1:1:31;
    if m==1;
        px=x;
        py=y;
        pz=z;
    elseif m==31;
        px=xin;
        py=yin;
        pz=zin;
    else;
        pz=pz-(z-zin)/30;
        py=py-(y-yin)/30;
        px=px-(x-xin)/30;
    end;

```

% THETA(1) CALCULATION

```

theta_a=atan2(py,px);
theta_b=atan2(-py,-px);
if abs(theta_a-theta(1)) < abs(theta_b-theta(1));
    theta(1)=atan2(py,px);
else;
    theta(1)=atan2(-py,-px);
end;

```

% THETA(2) Calculation

```

pxy=px*cos(theta(1))+py*sin(theta(1));
theta_a= atan2(-pz, pxy) + atan2(sqrt((2*a2*pxy)^2 +
(2*a2*pz)^2 - (a2^2-a3^2+pxy^2+pz^2)^2), a2^2-a3^2+pxy^2+pz^2);
theta_b= atan2(-pz, pxy) - atan2(sqrt((2*a2*pxy)^2 +
(2*a2*pz)^2 - (a2^2-a3^2+pxy^2+pz^2)^2), a2^2-a3^2+pxy^2+pz^2);
if abs(theta_a-theta(2)) < abs(theta_b-theta(2));
    theta(2)= atan2(-pz, pxy) + atan2(sqrt((2*a2*pxy)^2
+ (2*a2*pz)^2 - (a2^2-a3^2+pxy^2+pz^2)^2), a2^2-a3^2+pxy^2+pz^2);
else;
    theta(2)= atan2(-pz, pxy) - atan2(sqrt((2*a2*pxy)^2
+ (2*a2*pz)^2 - (a2^2-a3^2+pxy^2+pz^2)^2), a2^2-a3^2+pxy^2+pz^2);

```

```
end;
```

```
% THETA(3) CALCULATION %
```

```
theta_a= atan2(-sin(theta(2))*pxy - cos(theta(2))*pz,  
cos(theta(2))*pxy - sin(theta(2))*pz-a2);  
% theta_b= atan2(-(-sin(theta(2))*pxy - cos(theta(2))*pz), -  
(cos(theta(2))*pxy - sin(theta(2))*pz-a2));  
% if abs(theta_a-theta(2)) < abs(theta_b-theta(2));  
theta(3)= atan2(-sin(theta(2))*pxy - cos(theta(2))*pz,  
cos(theta(2))*pxy - sin(theta(2))*pz-a2);  
% else;  
% theta(3)= atan2(-(-sin(theta(2))*pxy -  
cos(theta(2))*pz), -(cos(theta(2))*pxy - sin(theta(2))*pz-a2));  
% end;
```

```
theta(4)=-theta(2)-theta(3)+pi/2; % Since all the joints are  
parallel, just add.
```

```
A1=a_matrix(theta(1),alpha1,a1,s1);  
A2=a_matrix(theta(2),alpha2,a2,s2);  
A3=a_matrix(theta(3),alpha3,a3,s3);  
A4=a_matrix(theta(4),alpha4,a4,s4);  
A5=a_matrix(theta(5),alpha5,a5,s5);  
T2=A1*A2;  
T3=A1*A2*A3;  
T5=A1*A2*A3*A4*A5;  
point1=T2*vector1;  
point2=T3*vector2;  
point3=T5*vector3;  
point4a=T5*vector4a;  
point4b=T5*vector4b;  
point4c=T5*vector4c;  
point4d=T5*vector4d;  
point4e=T5*vector4e;  
x1(m)=point1(1);  
y1(m)=point1(2);  
z1(m)=point1(3);  
x2(m)=point2(1);  
y2(m)=point2(2);  
z2(m)=point2(3);  
x3(m)=point3(1);  
y3(m)=point3(2);  
z3(m)=point3(3);
```

```

x4a(m)=point4a(1);
y4a(m)=point4a(2);
z4a(m)=point4a(3);
x4b(m)=point4b(1);
y4b(m)=point4b(2);
z4b(m)=point4b(3);
x4c(m)=point4c(1);
y4c(m)=point4c(2);
z4c(m)=point4c(3);
x4d(m)=point4d(1);
y4d(m)=point4d(2);
z4d(m)=point4d(3);
x4e(m)=point4e(1);
y4e(m)=point4e(2);
z4e(m)=point4e(3);
theta*180/pi
end;
z=pz;
x=px;
y=py;
t1=theta(1);
t2=theta(2);
t3=theta(3);
t4=theta(4);
t5=theta(5);
% Animating the linkage
itotal=m-1;
ans='y';
while ans=='y'
    for i=1:1:itotal;
        %set(hinge1,'xdata',[x1(i)],'ydata',[z1(i)])
        set(jointsa1,'xdata',x1(i),'ydata',z1(i))
        set(line1a,'xdata',[0 x1(i)],'ydata',[0 z1(i)])
        set(hinge1,'xdata',[x2(i)],'ydata',[z2(i)])
        set(jointsa2,'xdata',x2(i),'ydata',z2(i))
        set(line2a,'xdata',[x1(i) x2(i)],'ydata',[z1(i) z2(i)])
        %set(hinge1,'xdata',[x3(i)],'ydata',[z3(i)])
        set(jointsa3,'xdata',x3(i),'ydata',z3(i))
        set(line3a,'xdata',[x2(i) x3(i)],'ydata',[z2(i) z3(i)])
        set(grip1a,'xdata',[x3(i) x4a(i)],'ydata',[z3(i) z4a(i)])
        set(jointsa4,'xdata',x4a(i),'ydata',z4a(i))
        set(grip1b,'xdata',[x4b(i) x4c(i)],'ydata',[z4b(i) z4c(i)])
        set(grip1c,'xdata',[x4b(i) x4d(i)],'ydata',[z4b(i) z4d(i)])
        set(grip1d,'xdata',[x4c(i) x4e(i)],'ydata',[z4c(i) z4e(i)])
    end
end

```

```

%set(hinge2,'xdata',[y1(i)],'ydata',[z1(i)])
set(jointsb1,'xdata',y1(i),'ydata',z1(i))
set(line1b,'xdata',[0 y1(i)],'ydata',[0 z1(i)])
set(hinge2,'xdata',[y2(i)],'ydata',[z2(i)])
set(jointsb2,'xdata',y2(i),'ydata',z2(i))
set(line2b,'xdata',[y1(i) y2(i)],'ydata',[z1(i) z2(i)])
%set(hinge2,'xdata',[y3(i)],'ydata',[z3(i)])
set(jointsb3,'xdata',y3(i),'ydata',z3(i))
set(line3b,'xdata',[y2(i) y3(i)],'ydata',[z2(i) z3(i)])
set(grip2a,'xdata',[y3(i) y4a(i)],'ydata',[z3(i) z4a(i)])
set(jointsb4,'xdata',y4a(i),'ydata',z4a(i))
set(grip2b,'xdata',[y4b(i) y4c(i)],'ydata',[z4b(i) z4c(i)])
set(grip2c,'xdata',[y4b(i) y4d(i)],'ydata',[z4b(i) z4d(i)])
set(grip2d,'xdata',[y4c(i) y4e(i)],'ydata',[z4c(i) z4e(i)])

```

```

%set(hinge3,'xdata',[y1(i)],'ydata',[x1(i)])
set(jointsc1,'xdata',y1(i),'ydata',x1(i))
set(line1c,'xdata',[0 y1(i)],'ydata',[0 x1(i)])
set(hinge3,'xdata',[y2(i)],'ydata',[x2(i)])
set(jointsc2,'xdata',y2(i),'ydata',x2(i))
set(line2c,'xdata',[y1(i) y2(i)],'ydata',[x1(i) x2(i)])
%set(hinge3,'xdata',[y3(i)],'ydata',[x3(i)])
set(jointsc3,'xdata',y3(i),'ydata',x3(i))
set(line3c,'xdata',[y2(i) y3(i)],'ydata',[x2(i) x3(i)])
set(grip3a,'xdata',[y3(i) y4a(i)],'ydata',[x3(i) x4a(i)])
set(jointsc4,'xdata',y4a(i),'ydata',x4a(i))
set(grip3b,'xdata',[y4b(i) y4c(i)],'ydata',[x4b(i) x4c(i)])
set(grip3c,'xdata',[y4b(i) y4d(i)],'ydata',[x4b(i) x4d(i)])
set(grip3d,'xdata',[y4c(i) y4e(i)],'ydata',[x4c(i) x4e(i)])

```

```
drawnow;
```

```
end;
```

```
ans='n';
```

```
% inp=1;
```

```
% ans closes
end;
```

```
resp=1;
motion=0;
```

```
% in closes  
end;
```

```
while inp>0;  
    disp(' 0 = next')  
    disp(' 1 = point_1')  
    disp(' 2 = point_2')  
    disp(' 3 = point_3')  
    disp(' 4 = point_4')  
    disp(' 5 = theta_1')  
    disp(' 6 = theta_2')  
    disp(' 7 = theta_3')  
    disp(' 8 = theta_4')  
    disp(' 9 = theta_5')  
    disp(xin)  
    disp(yin)  
    disp(zin)  
    disp(mag)  
    disp(s)  
    disp('What do you want to do? (0-9) [0]: ')  
    inp=input('*>>');  
    fprintf('\n\n\n\n\n\n')  
    if inp==1;  
        point_1=[point1(1),point1(2),point1(3)]  
    elseif inp==2;  
        point_2=[point2(1),point2(2),point2(3)]  
    elseif inp==3;  
        point_3=[point3(1),point3(2),point3(3)]  
    elseif inp==4;  
        point_4=[point4a(1),point4a(2),point4a(3)]  
    elseif inp==5;  
        theta_1=t1*180/pi  
    elseif inp==6;  
        theta_2=t2*180/pi  
    elseif inp==7;  
        theta_3=t3*180/pi  
    elseif inp==8;  
        theta_4=t4*180/pi  
    elseif inp==9;  
        theta_5=t5*180/pi  
    else;  
        inp=0;  
    end;  
end;  
end;
```

end;

C. Joystick Control Code

The program JOY.C was written using a Borland C++ Compiler; and the port reading procedure was achieved with the aid of the library file: DOS.H. Attached is the c program:

```
/*
 * non-standard functions:
 *   outp(port, byte)   - outputs a byte to an I/O port
 *   inp(port)         - inputs a byte from an I/O port
 *   disable()         - disables CPU interrupts
 *   enable()          - enables CPU interrupts
 */

#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <mem.h>
#include <stdlib.h>

/* standard I/O port address for the PC joystick */

#define JPORT      0x201
/*
 * max counts to wait for joystick. In normal usage, you don't
 * want this too large, so that the program doesn't hang up too long
 * if no joystick is attached. But if you make it too small, a
 * program that works on a slow machine may fail on a faster CPU
 * (unless you have a "speed adjustable" game card).
 */

#define CMAX      9999

/*
 * Read 4 joystick pots and buttons.
 * Disabling interrupts during the polling provides steadier
 * readings, but you may not want to do that in real usage.
 *
 * Joystick values are unchanged if nothing's there.
 */
```

```

* To "calibrate" a joystick, call this routine while asking the
* user to move the stick to the corners, and see what
* limits you get.
*/

```

```

int a, bb, c, d, i;
int v[4];
char b[4];
int stick_a_min_x=0;      /* Minimum high state duration for X axis of joystick A */
int stick_a_max_x=0;      /* Maximum high state duration for X axis of joystick A */
int stick_a_cen_x=0;      /* Joystick A - X axis center position */
int stick_a_min_y=0;      /* Minimum high state duration for Y axis of joystick A */
int stick_a_max_y=0;      /* Maximum high state duration for Y axis of joystick A */
int stick_a_cen_y=0;      /* Joystick A - Y axis center position */
int stick_b_min_x=0;      /* Minimum high state duration for X axis of joystick B */
int stick_b_max_x=0;      /* Maximum high state duration for X axis of joystick B */
int stick_b_cen_x=0;      /* Joystick B - X axis center position */
int stick_b_min_y=0;      /* Minimum high state duration for Y axis of joystick B */
int stick_b_max_y=0;      /* Maximum high state duration for Y axis of joystick B */
int stick_b_cen_y=0;      /* Joystick B - Y axis center position */

```

```

char Pause ()
{
    char c;
    printf ("\n\nPress Enter to continue...");
    while ((c = getchar()) != '\n') { }
    return c;
}

```

```

void joystick(int v[4], char b[4]) {

    int i, j;
    int mask = 15;

    i=j=0;

    disable();

    outp(JPORT, 0);

    for (i=1; mask && i<CMAX; i++) {

        j = inp(JPORT) ^ mask;

        if (j & 1) {

```

```

        v[0] = i;
        mask ^= 1;
        if (a==0) printf("Joystick A - X Axis Detected.\n");
        a=1;
    }

    if (j & 2) {
        v[1] = i;
        mask ^= 2;
        if (bb==0) printf("Joystick A - Y Axis Detected.\n");
        bb=1;
    }

    if (j & 4) {
        v[2] = i;
        mask ^= 4;
        if (c==0) printf("Joystick B - X Axis Detected.\n");
        c=1;
    }

    if (j & 8) {
        v[3] = i;
        mask ^= 8;
        if (d==0) printf("Joystick B - Y Axis Detected.\n");
        d=1;
    }
}

if (a==0) a=2;
if (bb==0) bb=2;
if (c==0) c=2;
if (d==0) d=2;

enable();

j = inp(JPORT);

b[0] = !(j & 0x10);

b[1] = !(j & 0x20);

b[2] = !(j & 0x40);

b[3] = !(j & 0x80);

```

```

}

/* Compute axis coordinate */

short compute_axis (int axis, int min, int max) {
    if (max == min) return (0);
    else
        return (((long) 1000 * (long) (axis - min)) / (long) (max-min));
}

/*
 * Test program. Read and print the four axes (X1, Y1, X2, Y2)
 * and the four buttons over and over, until a key is pressed.
 */

main(int argc, char **argv)

{
    clrscr ();

    /* start with zeros, indicating nothing there */

    memset(v, 0, sizeof(v));

    joystick(v, b);

    if ((a || bb) ==1) {
        printf ("\nMove stick A to upper left corner, and press a button\n");
        while ((b[0] || b[1] || b[2] || b[3]) ==0) joystick(v, b);
        while ((b[0] || b[1] || b[2] || b[3]) !=0) joystick(v, b);
        stick_a_min_x=v[0];
        stick_a_min_y=v[1];

        printf ("Move stick A to lower right corner, and press a button\n");
        while ((b[0] || b[1] || b[2] || b[3]) ==0) joystick(v, b);
        while ((b[0] || b[1] || b[2] || b[3]) !=0) joystick(v, b);
        stick_a_max_x=v[0];
        stick_a_max_y=v[1];

        printf ("Move stick A to center, and press a button\n");
        while ((b[0] || b[1] || b[2] || b[3]) ==0) joystick(v, b);
        while ((b[0] || b[1] || b[2] || b[3]) !=0) joystick(v, b);
        stick_a_cen_x=v[0];
        stick_a_cen_y=v[1];
    }
}

```

```

    }

    if ((c || d)==1) {
        printf ("Move stick B to upper left corner, and press a button\n");
        while ((b[0] || b[1] || b[2] || b[3]) ==0) joystick(v, b);
        while ((b[0] || b[1] || b[2] || b[3]) !=0) joystick(v, b);
        stick_b_min_x=v[2];
        stick_b_min_y= v [3];

        printf ("Move stick B to lower right corner, and press a button\n");
        while ((b[0] || b[1] || b[2] || b[3]) ==0) joystick(v, b);
        while ((b[0] || b[1] || b[2] || b[3]) !=0) joystick(v, b);
        stick_b_max_x=v[2];
        stick_b_max_y= v [3];

        printf ("Move stick B to center, and press a button\n");
        while ((b[0] || b[1] || b[2] || b[3]) ==0) joystick(v, b);
        while ((b[0] || b[1] || b[2] || b[3]) !=0) joystick(v, b);
        stick_b_cen_x=v[2];
        stick_b_cen_y=v[3];
    }

    if ((a == bb == c == d) ==0) {
        printf ("\nNo Joystick Detected...\n");
        printf ("Make sure Joystick is connected to Game Port and try
again.");
        goto exit;
    }

    printf("\nStick A - X-Axis: MIN=%5d MAX=%5d
CENTER=%5d\n",stick_a_min_x,stick_a_max_x,stick_a_cen_x);
    printf("Stick A - Y-Axis: MIN=%5d MAX=%5d
CENTER=%5d\n",stick_a_min_y,stick_a_max_y,stick_a_cen_y);
    printf("Stick B - X-Axis: MIN=%5d MAX=%5d
CENTER=%5d\n",stick_b_min_x,stick_b_max_x,stick_b_cen_x);
    printf("Stick B - Y-Axis: MIN=%5d MAX=%5d
CENTER=%5d\n",stick_b_min_y,stick_b_max_y,stick_b_cen_y);

    printf("\n\n      Stick A          Stick B\n");
    printf("      -----          ----- \n");
    printf(" X-Axis  Y-Axis Btn 1 Btn 2  X-Axis  Y-Axis Btn 1 Btn 2\n");
    printf(" -----          ----- \n");

```

```

while (!kbit ()) {
joystick(v, b);

if (a==1) {
    if (v[0] < stick_a_min_x)
        v[0] = stick_a_min_x;

    if (v[0] > stick_a_max_x)
        v[0] = stick_a_max_x;
    printf ("%8d", compute_axis (v[0], stick_a_min_x,
stick_a_max_x));
}
else
if (v[0] == 0) printf ("    0");

if (bb==1) {
    if (v[1] < stick_a_min_y)
        v[1] = stick_a_min_y;

    if (v[1] > stick_a_max_y)
        v[1] = stick_a_max_y;
    printf ("%10d", compute_axis (v[1], stick_a_min_y,
stick_a_max_y));
}
else
if (v[1] == 0) printf ("    0");

if (b[0]==0) printf ("  UP");
else printf("  DOWN");
if (b[1]==0) printf ("  UP");
else printf("  DOWN");

if (c==1) {
    if (v[2] < stick_b_min_x)
        v[2] = stick_b_min_x;

    if (v[2] > stick_b_max_x)
        v[2] = stick_b_max_x;
    printf ("%10d", compute_axis (v[2], stick_b_min_x,
stick_b_max_x));
}
else
if (v[2] == 0) printf ("    0");

```

```

        if (d==1)    {
                    if (v[3] < stick_b_min_y)
                        v[3] = stick_b_min_y;

                    if (v[3] > stick_b_max_y)
                        v[3] = stick_b_max_y;
                    printf ("%10d", compute_axis (v[3], stick_b_min_y,
stick_b_max_y));
                    }
        else
        if (v[3] == 0) printf ("      0");

        if (b[2]==0) printf ("    UP");
        else printf("    DOWN");
        if (b[3]==0) printf ("    UP");
        else printf("    DOWN");
        printf("\r");
        }

exit:

        Pause ();
        return 0;

}

```

D. C++ Code for Motors

```

motor_angles(double dx, double dy, double dz, double dp, double dr, double theta_1,
double theta_2, double theta_3, double theta_4, double theta_5)

{

double a, b, c, d, e;

double s1, s2, s23, s234, c1, c2, c23, c234;

int k1, k2, k3, k4, k5, r1, r2, r3, r4, r5;

int L2, L3, L4;

double dtheta_1, dtheta_2, dtheta_3, dtheta_4, dtheta_5;

double dphi_1, dphi_2, dphi_3, dphi_4, dphi_5;

```

```

s1 = sin(theta_1);
s2 = sin(theta_2);
s23 = sin(theta_2 + theta_3);
s34 = sin(theta_3 + theta_4);
s234 = sin(theta_2 + theta_3 + theta_4);

```

```

c1 = cos(theta_1);
c2 = cos(theta_2);
c23 = cos(theta_2 + theta_3);
c234 = cos(theta_2 + theta_3 + theta_4);

```

```

a = L2*c2 + L3*c23 + L4*c234;
b = L2*s3;
c = L2*c2 + L3*c23;
d = L2*s2 + L3*s23;
e = L3*s4 + L2*s34;
f = L3*s3 + L4*s34;
g = L2*L3*s3;
h = L3*s3;

```

```

if (a > -0.001 & a < 0.001)

```

```

    dtheta_1 = (-s1/a)*dx + (c1/a)*dy;

```

```

else

```

```

    dtheta_1 = 0;

```

```

if (b > -0.001 & b < 0.001)

```

```

{
    dtheta_2 = ((-c1*c23)/b)*dx + ((-s1*c23)/b)*dy + (s23/b)*dz + ((L4*s4)/b)*dp;
    dtheta_3 = ((c1*c)/g)*dx + ((s1*c)/g)*dy + (-d/g)*dz + ((-L4*e)/g)*dp;
}

```

```

else

```

```

{
    dtheta_2 = 0;
    dtheta_3 = 0;
}

```

```

if (h > -0.001 & h < 0.001)

```

```

    dtheta_4 = ((-c1*c2)/h)*dx + ((-s1*c2)/h)*dy + (s2/h)*dz + (f/h)*dp;

```

else

$$d\theta_4 = 0;$$

$$d\theta_5 = dr;$$

$$d\phi_1 = k_1 r_1 d\theta_1;$$

$$d\phi_2 = k_2 r_2 d\theta_2;$$

$$d\phi_3 = k_3 r_3 d\theta_2 + k_3 r_3 d\theta_3;$$

$$d\phi_4 = k_4 r_4 d\theta_2 + k_4 r_4 d\theta_3 + k_4 r_4 d\theta_4 + k_4 r_4 d\theta_5;$$

$$d\phi_5 = k_5 r_5 d\theta_2 + k_5 r_5 d\theta_3 + k_5 r_5 d\theta_4 - k_5 r_5 d\theta_5;$$

}

E. Arm Transmission Bill of Materials

Part Description	Qty.	Supplier	Part #	Matl.	Mfg.	Cost	Total
Wrist and Gripper							
differential	1	Allied	DV7	Steel	none	\$252.95	\$252.95
bike cable	1	Shimano	***	Steel	none	\$8.99	\$8.99
1.75 dia grooved pulley	1	OSU	***	Steel	complete	***	***
.25" shaft collars	2	Bearing Dist.		Steel	none	\$1.50	\$3.00
Arm Transmission							
.1475" plate sprocket for differential	2	Allied	DEM36T44	Aluminum	open bore	\$10.45	\$20.90
.1475" plate sprocket for wrist transmission	8	Allied	DEM36T44	Aluminum	open bore, bolt circle	\$10.45	\$83.60
.25" hubbed sprocket for shoulder and elbow torque	2	Berg	25-EM-B-26	Steel	key, bolt circle	\$12.30	\$24.60
.25" plate sprocket for elbow transmission	2	Bornell Supply	25A26	Steel	open bore, bolt circle	\$10.86	\$21.72
elbow spacer (.313 thick)	1	OSU	***	Aluminum	bore	***	\$0.00
wrist spacer (.1875 thick)	4	OSU	***	Aluminum	bore	***	\$0.00
wrist shaft	1	OSU	***	Steel	turn, tap	***	\$0.00
elbow shaft	1	OSU	***	Steel	turn, tap	***	\$0.00
Miscellaneous Arm Transmission							
#4-40 socket head	75	Kimball	130 none	Steel	none	\$0.08	\$6.00
.1475" chain	25	Allied	DEL336	Plastic	none	\$7.52	\$188.00