EASY-C: DEVELOPMENT AND VERIFICATION OF A
RETROFITTABLE POWER-ASSISTED BASE FOR A SURGICAL C-ARM

by

Andrew Meyer

B.ASc., Queen’s University, 2013

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE
in
THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Biomedical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

December 2019

© Andrew Meyer, 2019
The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, a thesis entitled:

**EASY-C: DEVELOPMENT AND VERIFICATION OF A RETROFITTABLE POWER-ASSISTED BASE FOR A SURGICAL C-ARM**

__________________________________________________________

Andrew Meyer

submitted by ______________________________ in partial fulfillment of the requirements for

the degree of  MASTER OF APPLIED SCIENCE

in  Biomedical Engineering

**Examinining Committee:**

Dr. Antony Hodgson

Supervisor

Dr. Machiel Van Der Loos

Supervisory Committee Member

Dr. Minkyun Noh

Supervisory Committee Member
Abstract

C-Arms, a mobile X-ray machine with emitter and detector on opposite ends of a ‘C’, are used in many surgeries conducted within hospitals, especially in orthopaedic applications such as trauma repairs. The C-Arm, a 350kg unit, must be manually moved between storage, operating rooms, and various positions around the operating table, often requiring considerable physical exertion from the radiology technologists (RTs), putting them at high risk of musculoskeletal injury (MSIs). A powered robotic base was developed to alleviate this high risk of injury as well as the potential to allow for more precise movement and lowering of radiation dosage. This base, named the Easy-C, is retrofittable underneath existing C-Arms so that they retain their certifications while allowing “easy” movement by the RTs.

The Easy-C was designed with a rear set of wheels on a main structure and a nose wheel driven separately. Omni wheels were utilized to give holonomic motion; the Easy-C platform can move freely in the X-Y plane of the floor, unlike a shopping cart or other standard wheeled vehicle. This Easy-C system was verified on a C-Arm through open loop movement along the three major axes available: X, Y, and in-plane rotation ω, all which replicate clinically relevant movements. Low relative error was seen in X and Y movements with both only at 1.7% relative error on the main movement axis, and 7.7% or less in the off axes. In-plane rotation had a larger relative error of 6.2%, with 7.1% or less in the off axes. For open loop control, the Easy-C performed as expected across these movements, allowing for minimal effort from the operator to move the C-Arm and greatly reducing the MSI risk. While several limitations were realized, with future development the Easy-C could provide a new and effective tool to the healthcare industry.
Lay Summary

A mobile X-ray machine is used in many surgeries performed in the hospital, allowing the surgeon to view the internal structures and implants throughout an operation. This X-ray machine, called a C-Arm because of its C-like shape, is very heavy and must be pushed around on its wheel by operators called radiology technologists, or RTs for short. Ergonomically this puts the RTs at high risk of injury because of the postures, forces, and repetition required. To solve this issue, a robotic based, nicknamed the Easy-C, was designed and fabricated that fits under the C-Arm and allows the RT to drive it around with a controller. Through testing the Easy-C was shown to operate correctly in all its movements, potentially allowing any RT to move the C-Arm with minimal effort and drastically lowered chance of injury.
Preface

This thesis is an original piece of work by the author, Andrew Meyer. The original concept of the Easy-C system was developed by Dr. Antony Hodgson and Dr. Carolyn Anglin as part of a larger C-arm technology development program.
Table of Contents

Abstract........................................................................................................................................ iii
Lay Summary................................................................................................................................... iv
Preface............................................................................................................................................. v
Table of Contents ............................................................................................................................ vi
List of Tables ..................................................................................................................................... ix
List of Figures .................................................................................................................................... x
List of Abbreviations ....................................................................................................................... xiv
Acknowledgements ......................................................................................................................... xv

Chapter 1: Introduction and Background .................................................................................. 1

1.1 Image Guided Surgery .................................................................................................................. 1
1.2 Mobile X-Ray Machine (C-Arm) ................................................................................................ 3
1.3 C-Arm Operation and Risk of Strain Injuries ............................................................................ 5
1.4 Currently Available Solutions and Gaps ................................................................................... 8
1.5 Proposed Solution and Thesis Goal: Device Design, Fabrication, and Verification ............. 9
1.6 Thesis Outline ............................................................................................................................. 9

Chapter 2: Design ......................................................................................................................... 10

2.1 Initial Design Considerations ................................................................................................... 10
  2.1.1 Holonomic Wheel Kinematics ......................................................................................... 12
  2.1.2 Design Review ................................................................................................................ 16
  2.1.3 Preliminary System Overview ......................................................................................... 18
2.2 Chassis Design ......................................................................................................................... 19
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.1</td>
<td>Crossbeam Calculations</td>
<td>20</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Wheel Selection</td>
<td>22</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Chain Sizing</td>
<td>23</td>
</tr>
<tr>
<td>2.2.4</td>
<td>Bearing Force Calculation</td>
<td>24</td>
</tr>
<tr>
<td>2.2.5</td>
<td>Shaft Force Calculation</td>
<td>25</td>
</tr>
<tr>
<td>2.2.6</td>
<td>Bolt Force Calculation</td>
<td>25</td>
</tr>
<tr>
<td>2.3</td>
<td>Power System Design</td>
<td>25</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Motor Selection</td>
<td>26</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Gear Ratio</td>
<td>28</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Electronic Motor Controller Selection</td>
<td>28</td>
</tr>
<tr>
<td>2.3.4</td>
<td>Main Computer Selection</td>
<td>29</td>
</tr>
<tr>
<td>2.3.5</td>
<td>System Power Selection and Considerations</td>
<td>30</td>
</tr>
<tr>
<td>2.4</td>
<td>UI Design</td>
<td>33</td>
</tr>
<tr>
<td>2.5</td>
<td>Software Design</td>
<td>34</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Code Structure</td>
<td>35</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Closed Loop Feedback</td>
<td>38</td>
</tr>
<tr>
<td>2.6</td>
<td>Safety Systems</td>
<td>38</td>
</tr>
</tbody>
</table>

**Chapter 3: Development**

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>CAD Design</td>
<td>42</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Stress Analysis</td>
<td>45</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Device Fabrication</td>
<td>48</td>
</tr>
<tr>
<td>3.2</td>
<td>Device Assembly</td>
<td>48</td>
</tr>
</tbody>
</table>

**Chapter 4: Verification Testing**

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
</tr>
</tbody>
</table>

vii
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Verification Methodology</td>
<td>53</td>
</tr>
<tr>
<td>4.2</td>
<td>Verification Results</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td><strong>Chapter 5: Discussion and Future Work</strong></td>
<td>69</td>
</tr>
<tr>
<td>5.1</td>
<td>Research Questions</td>
<td>69</td>
</tr>
<tr>
<td>5.2</td>
<td>Relation to Other Literature</td>
<td>73</td>
</tr>
<tr>
<td>5.3</td>
<td>Limitations</td>
<td>76</td>
</tr>
<tr>
<td>5.4</td>
<td>Recommendations and Future Work</td>
<td>77</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Improvements to Current Platform</td>
<td>77</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Integration with Broader Technology</td>
<td>78</td>
</tr>
<tr>
<td>5.5</td>
<td>Conclusion</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td><strong>Bibliography</strong></td>
<td>80</td>
</tr>
<tr>
<td></td>
<td><strong>Appendices</strong></td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>Appendix A</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>Appendix B</td>
<td>93</td>
</tr>
</tbody>
</table>
List of Tables

Table 1 - REBA and RULA scoring for ergonomic assessment. ................................................................. 7

Table 2 - Risk analysis of the prototype Easy-C system, with relevant mitigation options to ensure safe operation of the Easy-C........................................................................................................ 40

Table 3 - Stress and deflection analysis of primary loading structural components. Max stress for the 304 stainless steel is 215MPa, with a maximum acceptable deflection of 1mm. ................. 46

Table 4 - Primary movement along the X axis to the middle setpoint. .................................................. 61

Table 5 - Primary movement along the X axis to the end setpoint........................................................ 61

Table 6 - Primary movement along the Y axis to the middle setpoint. .................................................. 63

Table 7 - Primary movement along the Y axis to the end setpoint........................................................ 63

Table 8 - Primary movement the in-plate rotation axis ω to the middle setpoint. ................................. 65

Table 9 - Primary movement along the in-plate rotation axis ω to the end setpoint. ............................. 65

Table 10 - Comparison of Easy-C and MEPUC systems. ...................................................................... 75
List of Figures

Figure 1 - Layout of the operating room during use of the C-Arm with supporting guidance technologies. ........................................................................................................................................... 2

Figure 2 - Directions of base movement available to a radiation technologist............................... 4

Figure 3 - Posture used for REBA and RULA scoring. This is a typical posture while pushing the C-Arm. To look past the C-Arm in the direction of travel, the head is usually out of plane from the body........................................................................................................................................... 6

Figure 4 - Proof of concept developed in partnership with Tangent Engineering........................... 11

Figure 5 - Degrees of freedom available when utilizing omniwheels. ............................................. 13

Figure 6 - Omwiwheel contact profiles for different types of roller design (Moreno 2016) .......... 13

Figure 7 - Layout for omniwheel placement......................................................................................... 14

Figure 8 - Holonomic wheel setup for testing driving function......................................................... 16

Figure 9 - Forces applied to C-Arm in static scenario. The CoG can shift forward or backwards, depending on the location of the 'C', which is moveable................................................................. 18

Figure 10 - Layout of components to be used in the Easy-C system................................................. 19

Figure 11 - Sub-components of a wheel assembly that need to be assessed to determine required specifications........................................................................................................................................ 20

Figure 12 - Omwiheel used for the Easy-C wheel assemblies.............................................................. 22

Figure 13 - Approximating the C-Arm as a rod for chain loading calculations. ......................... 24

Figure 14 - Brushed DC motors with gearing, brake, and clutch from the Pride 3 Ultra wheelchair ........................................................................................................................................... 27

Figure 15 - Electronic speed controllers used, the Syren 50A.......................................................... 29
Figure 16 - The Arduino Mega 2560 was used as the main microcontroller of the Easy-C. ....... 30

Figure 17 - Three of these 22.2V 8Ah Lithium Polymer batteries were used in parallel to power the Easy-C. .......................................................................................................................................................... 32

Figure 18 - Layout of the Easy-C control system. Relevant connections to the control handle and higher-level controller are also shown........................................................................................................................................... 35

Figure 19 - Computational flow running on the Arduino during regular operation of the Easy-C. ............................................................................................................................................................................................................... 37

Figure 20 - Final CAD design of the Easy-C system with the chain guards installed.............. 44

Figure 21 - Final CAD design of the Easy-C system without the chain guards installed........... 44

Figure 22 - Several components were designed and analyzed for stress and deflection .......... 46

Figure 23 - FEA stress analysis for the crossbeam. The max stress (115 MPa) is lower than the yield stress (215 MPa). Maximum loading was seen near the boundary conditions (green arrows). Red arrow designates direction of gravity.............................................................. 47

Figure 24 - FEA displacement analysis for the crossbeam. Deflection numbers are within expected ranges, with the maximum deflection occurring at point of force application (pink arrow). Red arrow designates direction of gravity.............................................................. 47

Figure 25 - Easy-C system assembled with wooden chassis for initial testing. ..................... 49

Figure 26 - For mounting the Easy-C system to the C-Arm, ramps were fabricated to lift the C-Arm to the desired height so the Easy-C could be installed. The front wheel was levered up for installation................................................................................................................................................................................................. 51

Figure 27 - Easy-C mounted to the C-Arm............................................................................. 52

Figure 28 - Movement along the operating bed in the X direction, mimicking IM nail implant guidance......................................................................................................................................................................................... 54
Figure 29 - Easy-C at the start of the lateral translation, after a movement to the right, and after its return to the original location. Ideally, the end location would coincide with the starting position. ........................................................................................................................................... 54

Figure 30 - Movement along the operating bed in the Y direction, mimicking the C-Arm moving in and out from the operating space. ........................................................................................................................................... 55

Figure 31 - Easy-C at the start and end of the forward translation movement and following its return. ........................................................................................................................................... 55

Figure 32 – Rotation of the C-Arm about its imaging beam, mimicking achieving an optimal position in the operating environment. ........................................................................................................................................... 56

Figure 33 - Easy-C at the start of a rotation movement, after shifting the base clockwise relative to the front pivot location, and after returning to the original position. ........................................................................................................................................... 57

Figure 34 - Location of triads placed on the C-Arm for positional tracking. The primary triad is the bottom of the two pictured, with the secondary top triad used for redundancy as needed. .... 58

Figure 35 - Single run of X movement. ........................................................................................................................................... 60

Figure 36 - Primary movement along the X axis of the C-Arm (~800 mm displacement). ........ 60

Figure 37 - Single run in the Y direction. ........................................................................................................................................... 62

Figure 38 - Primary movement along the Y axis of the Easy-C (~700 mm displacement). ....... 62

Figure 39 - Single run of in-plane rotation. ........................................................................................................................................... 64

Figure 40 - In-plane rotation of the Easy-C (~62 deg of rotation). ........................................................................................................................................... 64

Figure 41 - Drift of the center of rotation during in-plane rotation for 4 selected movements. Point 1 is the start of the movement, 2 is the middle of the movement, 3 is the end of the movement........................................................................................................................................... 66
Figure 42 - The front wheel does not move correctly when compared to the expected motion of the C-Arm. ........................................................................................................................................................................... 67

Figure 43 - Small oscillations can be seen in the device trajectory as it moves in the X direction. ...................................................................................................................................................................................................................... 68
List of Abbreviations

CHHM – Centre for Hip Health and Mobility

ESC – Electronic Speed Controller

IMU – Inertial Measurement Unit

OR – Operating Room

PWM – Pulse Width Modulation

RC – Radio Control

REBA – Rapid Entire Body Assessment

RT – Radiation Technologist

RULA – Rapid Upper Limb Assessment
Acknowledgements

I would first like to thank Dr. Antony Hodgson and Dr. Carolyn Anglin, both who supervised my thesis project. Without their help, guidance and perseverance, this thesis may never have come to fruition.

Dr. Pierre Guy was tremendously helpful by allowing me to view several orthopedic surgeries and has given me lots of feedback on the system I was working with. This gave me untold insight into the problem that I was working on.

My colleagues and friends through the program were unendingly helpful in providing me with what I need when I needed it. Lots of knowledge and expertise was shared that can never be replaced.

The Centre for Hip Health and Mobility at Vancouver General Hospital for their access to devices and space for development and testing.

Special thanks are owed to my parents, who have supported me throughout my years of education, both morally and financially. My mother especially, who was always the strong-willed guidance that is required when in a rut.

Canadian Institute of Health Research and the Natural Sciences and Engineering Research Council of Canada for funding this research.
Chapter 1: Introduction and Background

X-ray fluoroscopy is widely used in orthopaedic procedures and is increasingly important as surgeons strive to reduce the invasiveness of surgical procedures (Zheng 2015). However, using the main intraoperative X-ray imaging device (called a C-arm, because the X-ray emitter and receiver are mounted at opposite ends of a large C-shaped beam that allows the X-ray beam to be positioned to pass through a patient on an operating table (Fujii 2003)) exposes the surgical team and the patient to radiation and, because X-ray images frequently need to be obtained from different angles and perspectives, requires frequent maneuvering into new positions (Suhm 2004); these positioning maneuvers are often time-consuming and can involve stressful interactions between the surgeon and the radiology technologist (‘radtech’) who operates the machine. There is therefore a need to improve the operational efficiency of C-arms.

1.1 Image Guided Surgery

With the advent of image guided surgery, operating surgeons are now able to make small incisions or cuts on the patient to access the body without having to open the patient in a more invasive manner. This allows for faster recoveries and a lower risk of post-operation complications; implants can be installed, foreign bodies can be removed, broken tissues can be fixed (MSMC 2016). C-Arms allow for live video or still X-ray images to be taken while the procedure is being conducted, and the C-Arm is moved around the area of interest during the procedure. Image-guided surgery is especially prevalent in orthopedic surgery, especially for trauma-related surgeries.
To complement C-arms, additional devices and systems have also been developed to enable accurate quantitative information to be obtained during surgery and used to guide interventions. These include computers, optical and electromagnetic trackers, and even robotic devices. Using such tools is often referred to as Computer Assisted Orthopedic Surgery (or CAOS / navigation) (Zheng 2015). The goal of these systems (see Figure 1) is to improve accuracy and efficacy of the surgeries, while also reducing the harmful side effects of utilizing the C-Arm such as radiation dosage or repetitive strain injuries.

**Figure 1 - Layout of the operating room during use of the C-Arm with supporting guidance technologies.**
The field of image-guided surgery is continuing to grow and improve as the realization of this technology becomes more widely adopted and accepted. More complicated surgeries are now being approached utilizing this technology, and many more will come as the technology is advanced (Mezger 2013), requiring radtechs to interact with C-Arms on an increased basis.

1.2 Mobile X-Ray Machine (C-Arm)

C-Arms have several degrees of freedom built in, allowing it to image the patient from a wide range of orientations. This includes the ability to move the “C” relative to the base, or to move the base relative to the floor. The “C” has the following five degrees of freedom relative to the base: three rotations (orbit (pitch), wig-wag (yaw), tilt (roll)) and two translations (up-down and in-and-out). This allows for the “C” to be positioned in almost any orientation around the patient, with sufficient flexibility to be able to avoid or accommodate obstacles such as the operating room table support column. For the purpose of this thesis, these degrees of freedom are not monitored or considered.

Rather, the primary interest in this thesis is in the movement of the C-arm base relative to the floor. This is a planar movement with 3 degrees of freedom (X, Y, ω as seen in Figure 2). These movements are similar to those that can be made with a shopping cart, though with some caveats and differences.
Figure 2 - Directions of base movement available to a radiation technologist.

The C-Arm itself is a large and cumbersome device to move as they can weigh approximately 800lbs (roughly 350kg). For reference, this would be similar to pushing around a shopping cart with 5 people inside it, and it can be difficult to make precise maneuvers with such a mass involved.
1.3 C-Arm Operation and Risk of Strain Injuries

The C-Arm is operated by Radiation Technologists, or RT’s. In a typical procedure, they move the C-arms from storage, to the operating room, within the operating room, and back to storage if another surgery is not immediately pending. This can be an ongoing process over the course of a shift, upwards of 12 hours or more.

I attended several orthopaedic surgeries and observed the amount of movement of the C-Arm during these surgeries. The surgeries involved were hip or lower limb operations, typically for hip replacements or fracture fixation where the C-Arm was used to allow for minimally invasive placement of hardware into the patient. These surgeries lasted several hours, and usually required between 16 and 21 base movements, with many additional movements of the ‘C’ itself relative to the base.
To evaluate the risk that RT’s are subject to while moving the C-Arm, I conducted a review using the REBA (Rapid Entire Body Assessment) and RULA (Rapid Upper Limb Assessment) tools to evaluate the posture of the RT while performing a base movement into the operating table (this is also the posture used to push the C-Arm between ORs or storage). These evaluations consider variables such as flexion / extension of various joints, loading scenarios, and the repetitiveness of the task. For this posture, several joints are outside of neutral axes while being subject to high loading, for a task that is repeated several times in a surgery. From my evaluation, the REBA score was 8 while the RULA score was 7. As can be seen in Table 1, both
assessment scores put the RT at high risk of musculoskeletal injury and are normally interpreted as a recommendation to investigate and implement change.

Table 1 - REBA and RULA scoring for ergonomic assessment.

<table>
<thead>
<tr>
<th>REBA</th>
<th>RULA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SCORE</strong></td>
<td><strong>Level of MSD Risk</strong></td>
</tr>
<tr>
<td>1</td>
<td>Negligible risk, no action required</td>
</tr>
<tr>
<td>2-3</td>
<td>Low risk, change may be needed</td>
</tr>
<tr>
<td>4-7</td>
<td>Medium risk, further investigation, change soon</td>
</tr>
<tr>
<td>8-10</td>
<td>High risk, investigate and implement change</td>
</tr>
<tr>
<td>11+</td>
<td>Very high risk, implement change</td>
</tr>
</tbody>
</table>

My evaluation is consistent with one reported by Kim (2014), who conducted REBA and RULA evaluations of RTs pushing a heavy radiation apparatus. They found that postures involved in this task were scored as “should be improved” and “improve immediately”, both classifications that indicate a high risk of injury. Kumar (2004) reviewed RTs and found a significant amount of MSI problems with 83% reporting having backache and other pain in the neck or shoulder, encouraging this link between high MSI risk due to poor ergonomics.

Due to these risks, we identified the need for a device that could reduce the physical demands on the RTs who operate these C-arms. There is also the consideration for accurately quickly moving...
the C-Arm within the operating space. Lowering radiation dosage to the operating staff and patient is another consideration that the Easy-C platform could achieve through returning to known locations without requiring scouting shots. Matthews (2007) found that with guided movement, no scout shots were required when returning to a previous position, whereas it previously required 3 to 12 shots.

1.4 Currently Available Solutions and Gaps
A few approaches have been proposed to improve the operation of intraoperative C-arms and reduce the risk of musculoskeletal injury that they pose.

At the high end, one form of C-Arm is a permanently mounted robotic device installed in dedicated ORs. RTs don’t have strain-related issues with installed and motorized C-Arms as they do not have to manually drive them to position. However, they are expensive and are not readily available in most hospitals (GVR 2017).

A powered base retrofittable to an existing C-arm was developed by Suhm (2003) and named the MEPUC (Motorised Exact Positioning Unit for C-arm). This unit was installed under a conventional C-Arm to allow movement of the base via a control unit; two articles were published providing a qualitative description of the MEPUC’s operation (Suhm 2003, Suhm 2004) but no quantitative verification tests or descriptions of further developments were identified.
Because the MEPUC showed promise for improving the ease of operation of conventional C-arms, but does not appear to have been further developed or commercialized, our group decided to develop our own device to address the need for RTs to be able to more easily maneuver C-arms in the operating room.

Powered operation of other machinery should also be considered for any relevant features or functions that could be incorporated. For example, pallet lifts have been developed that employ powered motion as their method for moving heavy loads (Rose 2010, Dixon 2014). Powered wheelchairs are used for motion without requiring physical exertion on the part of the user or external operator (Mascaro 2002, Wada 2007).

1.5 Proposed Solution and Thesis Goal: Device Design, Fabrication, and Verification

The purpose of the work described in this thesis, therefore, was to design, build and evaluate the performance of a retrofittable power-assisted C-Arm drive unit (referred to as the Easy-C) aimed at enabling RTs to maneuver C-arms through a range of relevant intraoperative motions while markedly reducing or eliminating their risk of musculoskeletal injury.

1.6 Thesis Outline

The remainder of this thesis is organized as follows:

- Chapter 2: Design
- Chapter 3: Development
- Chapter 4: Verification Testing
- Chapter 5: Discussion and Future Work
Chapter 2: Design

This chapter outlines the design of the Easy C system, including a functional overview, description of components, and a presentation of the software architecture.

2.1 Initial Design Considerations

Prior to my taking on this project, an engineering company in Calgary (Tangent Design Engineering) had, in partnership with Drs. Hodgson and Anglin, developed an initial proof of concept for a motorized cart that could be mounted beneath a C-arm (see Figure 4). This design featured three ‘omniwheels’ – a special form of wheel comprised of smaller rollers mounted circumferentially around the main wheels that allows for lateral sliding motion and driven fore-aft motions. By appropriately driving each wheel, the cart can be driven in arbitrary directions in the plane – forward, backward, side-to-side, and rotating around arbitrary pivot points.

The TDE design was fitted to a particular decommissioned C-Arm shell and was driven using a force-sensing handle. It served its function of demonstrating the basic concept, but had several limitations that I sought to address.

First, their design required modification of the C-Arm chassis and would void any manufacturer’s warranty and medical certification, possibly disqualifying it for use in the operating room. Further, this design was customized for the particular C-arm they had at hand and was not suitable for use with the machines available at VGH.
Second, the cart was mounted entirely underneath the existing cart and raised the C-arm several centimeters above the ground – this could impair the C-arm’s functionality as it often has to be lowered sufficiently to reach underneath the operating room table.

Third, only qualitative evaluations were performed, and the electronics design was not suitable for direct interfacing with a higher-level controller.

![Image](image_url)

Figure 4 - Proof of concept developed in partnership with Tangent Engineering.
Our goals, therefore, were to address these deficiencies and design an improved version of this motorized cart that could be readily mounted to at least one C-arm machine available at VGH (the Siemens Arcadis Orbic 3D machine available at the Center for Hip Health and Mobility), and ideally easily adapted to various C-Arm types, including several units from GE and Philips that are in clinical use at VGH. These machines were inspected to determine the relevant mechanical interface parameters needed to ensure cross compatibility of our updated design.

2.1.1 Holonomic Wheel Kinematics

The initial prototype incorporated holonomic motion constraints via the use of omniwheels, which enables the cart to move in any of the 3 DOF available independently (forward/back, left/right, rotation, Figure 5) through an appropriate selection of drive commands to the omniwheels. By way of contrast, a shopping cart cannot move freely left or right and is therefore said to be nonholomically constrained. C-Arms typically incorporate caster wheels and operate similarly to shopping carts, with force provided by the user through guidance handles, but these require significant force to initiate movement. With motorized holonomic omniwheels, the Easy C cart would allow for clinically relevant motions to be executed via software commands and/or low-force-input devices.
Consideration for the type of omniwheel should be incorporated into the design of the wheel drive as different wheel setups can generate different contact profiles (as outlined in Figure 6). A wheel with intermittent contact could cause unwanted vertical motion of the final assembly.

Given the dimensions of the C-Arm, effective positions for the omniwheels were identified and parameterized as shown in Figure 7. From this, the equations for each motor drive could be calculated to achieve this motion, shown in Equations 1-8. The development of omnidirectional movement has been developed for various robotic platforms previously, and the general
methodology is followed here (Baede 2006, Moreno 2016). The following section describes how to determine the motor speed required at each wheel for a desired movement of the cart.

The velocity of any point on the Easy-C can be broken down into contributions due to translational and rotational components of motion expressed based on the motion at a reference point on the cart (shown at the top of Figure 7):

\[
\vec{V}_{EasyC} = \vec{V}_{trans} + \vec{V}_{rot} \quad (1)
\]
The translation and rotational vectors for motor ‘i’ can be expressed in their x, y and \(\omega\) components. Utilizing the dot product of the drive directions and their moment arms gives (2).

\[
\vec{V}_{trans,i} = \cos \alpha_i \vec{x} + \sin \alpha_i \vec{y}, \quad \vec{V}_{rot,i} = L_i \sin \theta_i \vec{\omega}
\]  

(2)

This relates the velocity of each motor to the overall movement of the Easy-C.

\[
\vec{V}_1 = \cos \alpha_1 \vec{x} + \sin \alpha_1 \vec{y} + L_1 \sin \theta_1 \vec{\omega}
\]

(3)

\[
\vec{V}_2 = \cos \alpha_2 \vec{x} + \sin \alpha_2 \vec{y} + L_2 \sin \theta_2 \vec{\omega}
\]

(4)

\[
\vec{V}_3 = \cos \alpha_3 \vec{x} + \sin \alpha_3 \vec{y} + L_3 \sin \theta_3 \vec{\omega}
\]

(5)

The following variables are known from the measurements of the C-Arm shown in Figure 7.

\[\begin{align*}
\alpha_1 &= 180^\circ, \alpha_2 = 300^\circ, \alpha_3 = 60^\circ \\
\theta_1 &= 90^\circ, \theta_2 = \theta_3 = 44.6^\circ \\
L_1 &= 0.45m, L_2 = L_3 = 1.24m
\end{align*}\]

These values were substituted into Equations 3-5 to yield Equations 6-8.

\[
\vec{V}_1 = -1\vec{x} + 0\vec{y} + 0.45\vec{\omega}
\]

(6)

\[
\vec{V}_2 = 0.5\vec{x} - 0.866\vec{y} + 0.87\vec{\omega}
\]

(7)

\[
\vec{V}_3 = 0.5\vec{x} + 0.866\vec{y} + 0.87\vec{\omega}
\]

(8)

Combining Equations 6-8 in matrix format yields Equation 9, the motor mapping for driving the omniwheels.

\[
\begin{bmatrix}
\vec{V}_1 \\
\vec{V}_2 \\
\vec{V}_3 
\end{bmatrix} =
\begin{bmatrix}
-1 & 0 & 0.45 \\
0.5 & -0.866 & 0.87 \\
0.5 & 0.866 & 0.87
\end{bmatrix}
\begin{bmatrix}
\vec{x} \\
\vec{y} \\
\vec{\omega}
\end{bmatrix}
\]

(9)

The rotational speed of each wheel is then determined by dividing the corresponding wheel centre velocity by the radius of each wheel.
To test the driving characteristics of this omniwheel robot, a small-scale version was created (Figure 8) with similar ratios to the Easy-C wheel placement. Taking inputs from a wireless radio controller, it was possible to drive this device through the motions illustrated in Figure 5.

![Holonomic wheel setup for testing driving function.](image)

**Figure 8 - Holonomic wheel setup for testing driving function.**

### 2.1.2 Design Review

Several design requirements were identified for the Easy-C from the above work and review, alongside various other constraints that needed to be taken into consideration during the development process.
The primary driver for the Easy-C platform was to retain full functionality of the C-Arm and not negatively impact its use in the operating room. The includes several key factors, including the following:

- Minimizing the additional height added to the C-Arm. The more the C-Arm needs to be lifted, the higher the minimum height that the C-Arm can achieve and the more likely it is to interfere with other OR tools (such as the operating table).
- Access to all C-Arm UI elements, plugs, latches, etc. The setup, operation, and use of the C-Arm should not be deviated from unless directly required by the Easy-C system.
- Installation of the Easy-C should not require any modification to the C-Arm itself but should be retrofittable. Installation / teardown time / effort should be minimized.
- The Easy-C should operate on battery power. This allows for movement to and from the OR without needing to be plugged in, as well as less cords during use in the OR. This battery power should enable a full day’s use before requiring being recharged.
- The Easy-C should allow movement when the unit is out of power, to allow it to be moved or repositioned normally.
- The Easy-C user interface (UI) should be simple and obvious to use. Even if someone is not trained in the use of the Easy-C, they should be able to determine how to move the device simply by inspecting the UI.
- The maximum speed the device can reach should not be faster than the average human walking speed. Acceleration should be limited to reduce any jarring movement, with the value of this limit to be determined through testing.
- Holonomic motion should be incorporated to enable flexible movement around the OR. This gives the RTs access to specialized movements compared to a standard C-Arm.
2.1.3 Preliminary System Overview

With the design requirements and constraints identified above, a preliminary design concept was produced. Based on a plan to use the Siemens Orbi C-Arm available at VGH, as well as an inspection of other possible C-arms in use in the operating rooms there, I determined that the system should be developed in two separate parts. These would consist of two sets of rear holonomic wheels connected under the body of the C-Arm, with a separate front holonomic wheel set installed under the front C-Arm wheels (see Figure 10). The space constraints of the C-Arm required the motor for this front wheel to also be installed on the main wheel assembly, with a drive shaft installed to connect to and drive the front wheel set. This allowed unrestricted movement of the “C” above the cart components.

Figure 9 - Forces applied to C-Arm in static scenario. The CoG can shift forward or backwards, depending on the location of the 'C', which is moveable.
Figure 10 - Layout of components to be used in the Easy-C system.

The electronics and batteries would be installed on the main wheel assembly of the Easy-C, with the input handle being a load cell input determining drive direction. This was to be mounted alongside the UI elements of the C-Arm itself.

2.2 Chassis Design

From these design elements, a preliminary design was created to determine viability of the various components. This design is depicted in the CAD model below. This design required several calculations to determine loading and deflection characteristics of the various structural
components to ensure the system would be stable throughout its operation. Because of the low speed of operation of the device, all loading calculations were done assuming static loading, with a safety factor of at least 3 used to ensure no critical loading limits would be reached during normal and anticipated loadings when using the Easy-C.

![Diagram of sub-components of a wheel assembly](image)

**Figure 11 - Sub-components of a wheel assembly that need to be assessed to determine required specifications.**

### 2.2.1 Crossbeam Calculations

The major design consideration was the crossbeam portion of the Easy-C, which spans under the main body of the C-Arm between the back left and right wheels. The front wheel was determined to be mountable independently of the rear wheels, so no beam structures were required to attach it to the rear set of wheels.
This crossbeam was considered to function as a simply supported beam structure with two point loads applied at equal distances from the pivot points. Measurements were taken from the Orbic C-Arm to approximate the span and force locations.

The beam structure was initially approximated to be a square hollow tube section made of 304 stainless steel. This allowed for an initial stress analysis to determine the size, thickness, and material type for the crossbeam.

A maximum allowed deflection at the center of the beam was set to 3mm. An initial wall thickness of 2mm was chosen for its availability. Material was selected to be 304 SS for its lower cost, ease of machining, and weldability. This allowed the calculation of the tube dimensions under loaded conditions via maximum deflection beam equations. The weight of the C-Arm was obtained from spec sheets and is approximately 350kg. No additional weight (the weight of the Easy-C) was added for this initial calculation, with the goal of having a minimum safety factor of 3 to allow for this weight to not impact performance.

The beam section was treated as a simply supported beam at both ends with two equidistant forces applied to the beam itself. These mimic the wheels at each end of the beam with the load being transferred to the beam through two contact points to the C-Arm base.

This initially yielded a width and height of 20.3mm with the 2mm wall thickness. Upon searching for an adequate tube, it was determined that 1.5” square tube was easiest to obtain. This resulted in a width and height of 38.1mm with a wall thickness of 3mm.
With this initial tube cross-section, the stresses in the tube were calculated to determine how close to failure the beam was under loading and what factor of safety was available. Using the new tube dimensions, it was calculated that the bending stress was lower than the yield stress of the material with a safety factor of 3.9 and the max deflection was lower than the initial requirement by a safety factor of 5.

2.2.2 Wheel Selection

Omniwheels allow for holonomic motion of the Easy-C, and as such were chosen. Several different options are available on the market. There were several restrictions for these omniwheels, primarily related to the large weight being applied and the confined space they would need to be installed in.

Because the C-Arm weighs ~350kg, several wheels would need to be installed in parallel to accommodate the weight of the C-Arm and Easy-C. This would also allow for smoother motion as well, as the rollers are not uniformly distributed around the perimeter of the wheel.

![Figure 12 - Omniwheel used for the Easy-C wheel assemblies.](image)
The wheels chosen were from Rotocaster and are rated to 115kg each. These wheels were identical to the wheels used on the initial prototype and functioned adequately in that prototype. It was determined that a total of 12 wheels would be used, giving a safety factor of 3.9. These wheels would be spread equally through the 3 motor drives, giving a total of 4 wheels per motor output shaft.

Another important design consideration is the diameter of the wheel itself, as this affects the final achievable speed of the device. Since it is desired to minimize any increase in the operating height of the C-Arm, I felt that the Rotocaster wheel diameter of 125mm was sufficiently low to not affect operation to any great degree while being large enough to roll over objects in the wheels path without skipping (such as cables). I therefore used this diameter to estimate the maximum necessary driving speed for the wheel motors. Based on matching the average walking speed of a human (~1.4m/s, measured by walking a set distance), as faster is not desirable or required, I found that the wheel RPM would need to be 22.4 rad/s or 214 RPM.

### 2.2.3 Chain Sizing

The drive chain from the motor output to the wheel shaft needs to be able to handle the force applied to it. For this, the C-Arm was modeled as a rod rotating about its end with a single motor driving the end of this rod. A rotational acceleration of 0.5rad/s was approximated as the maximum that would be applied to the system. This would approximate the torque required to be applied to the wheel from the chain in the highest loading scenario of maximum output from a single motor.
This yielded a maximum force of 30lbs through the chain, calculated from the ratio of wheel diameter to spur gear diameter. From this, #35 chain was chosen as it has a working load rating of 190lbs giving a safety factor of 6.3.

2.2.4 Bearing Force Calculation

The wheel shaft will be supported by two bearings, one on each end of the shaft. This gives a total of 6 bearings across the three-wheel assemblies. Under a worst-case scenario in which the wheel assemblies are deflected and only the 3 inner bearings were taking the majority of the load, it was determined the safety factor was 5.1 given a max static loading of a single bearing of...
1300lbs (~590kg). Theoretically a single bearing could support the weight of the entire C-Arm. Since the total number of wheel rotations anticipated for the Easy-C device are not expected to rise into the tens of millions, I did not make any adjustments for a longer operating life.

2.2.5 Shaft Force Calculation

The main shaft for each wheel set will contain 4 wheels and be supported on each end by a bearing. The max stress in the beam was calculated by modeling the shaft as a double pin supported beam with two point loads approximating the wheel set locations (same as the crossbeam modeling). Because the omniwheels had a set inner shaft diameter of 7/8”, this was chosen as the starting size for the shaft. From this, it was calculated that the max bending stress seen in the shaft would be approximately 64 MPa. A shaft with a max bending strength of 341 MPa was chosen, yielding a safety factor of 5.3.

2.2.6 Bolt Force Calculation

An initial bolt size of 6mm (M6 thread size) was chosen for bolting the assembly together. The shear stress applied to the bolt was modeled as two plates with loads being applied in opposite directions. With this loading, the bolt would experience loading under its maximum allowable, with a safety factor of 4.7.

2.3 Power System Design

The power system consists of the mechanical and electrical subsystems that work together to produce the desired movement of the Easy-C platform: the motor, electronic speed controller, microcontroller, and battery. These subsystems had several options available but were narrowed
down to the chosen system for several reasons, though many options would still be viable based on changing requirements.

### 2.3.1 Motor Selection

The main drive motors, for which one is required for each of the wheel sets giving a total of three, require a large amount of torque to overcome the inertia of the C-Arm. When choosing the motor, several options were considered with the following constrains:

- Standard operating voltages (12V, 24V)
- Large gear ratios, preferably planetary, to obtain desired RPMs while maintaining high torque
- As low cost as possible
- Easily accessible
- Proven reliability

It was determined that brushed DC motors would be the most economical and easily accessible option for the drive motors. Several brushless systems were considered, but the cost and gearing requirements limited their viability.

It was determined that wheelchair motors would be the best option available that would satisfy all of the requirements and constraints set out. A standard wheelchair motor operates at the required voltages while outputting enough torque and desired RPMs. They are also low cost, easy to purchase, and are well proven in the wheelchair industry.
The wheelchair motor that was chosen was from the Pride 3 Ultra wheelchair. Researching this motor showed that it operated at 24V with a stall current of 40A. From the max speed of the wheelchair of 6.44 km/h, it was possible to determine the shaft output speed at 135 RPM, which with minimal gearing would reach the desired speed of the Easy-C. While no torque information could be obtained from the manufacturer, it was hypothesized that they would have adequate torque as the loading and inertia of a wheelchair would approximate that of one third of the Easy-C with a C-Arm loaded onto it.

These motors were easily obtained, and, though in used condition, operated at expected voltages and RPMs. They also allowed for freewheeling of the output shaft via a lever, which meets one of the project requirements for the Easy-C to be moveable without power. Most other motors and gear sets did not have this available. These Pride 3 motors also had an electronic brake for emergency use and were implemented as part of the safety review.
These motors differed from the motors used on the initial proof of concept for several of the aforementioned reasons including ease of availability, pricing, and ease of integration.

2.3.2 Gear Ratio

From the previously calculated desired speed of 214 RPM, it was possible to calculate the required gear ratio with the known motor output shaft of 135 RPM. This gave a desired gear ratio of 1.59.

The previously chosen #35 chain restricted the sprocket sizes, leading to the true gear ratio of 25/15 teeth, or 1.67. This gave a final maximum wheel rotational speed of 225 RPM.

2.3.3 Electronic Motor Controller Selection

With the known motor specifications, an electronic motor controller could be chosen. These requirements were:

- Brushed DC operation
- Minimum 40A operational current
- 24V operation
- Standard input methods (RC, Serial, Analog)
- Low cost

Based on these requirements, the Syren 50A ESC from Dimension Engineering was chosen. It met the requirements listed above and were easily attainable. These ESCs were also initially used
on the previous prototype, so were proven to be able to handle the requirements above. With software limits available on the ESC, the motors would be able to pull a maximum of 50A at 24V, or 1200W each. For 3 drive motors, this would total 150A at 24V, or 3600W. Based on 40A for stall current at 24V, the motors are not expected to reach this upper limit.

![Figure 15 - Electronic speed controllers used, the Syren 50A.](image)

### 2.3.4 Main Computer Selection

The main computer for the Easy-C initially was selected based on the following requirements:

- USB support for load cell user input method
- Compute power to handle camera tracking support
- Several general I/O ports for motor control, sensor integration, etc
- Code support
- Low cost

From this, the Beaglebone Black was chosen for the initial design. The initial small-scale prototypes utilized this computer. However, upon further review, it was deemed not necessary,
as many of the computational tasks could be delegated to a supervisory computer to be developed later to run the overall system user interface.

An Arduino Mega was therefore substituted as the main microcontroller of the Easy-C system. This would handle low level functionality of the Easy-C while communicating with a main computer (which would be handling various other tasks incorporated in the overall project). This opened up much larger code support and simple I/O interfacing. This also gave access to many accessories within the Arduino space that may be required as part of the future considerations for the project.

![Arduino Mega 2560](image)

**Figure 16 - The Arduino Mega 2560 was used as the main microcontroller of the Easy-C.**

### 2.3.5 System Power Selection and Considerations

The standard voltage of 24V was one of the main considerations of the motor selection, and therefore selects many of the subsidiary electronic components. Several step-down regulators and power filters were required to obtain stable and reliable usage of the Easy-C system as DC motors can generate large amounts of electrical noise during operation.
Initially, tethered operation with an AC/DC unit was considered, as the load cell originally imagined as the primary input device required an AC power source. However, as this input source was later changed, batteries were chosen due to lower cost, tether-less operation, and acceptable power delivery ratings.

The batteries have several criteria that would need to be met:

- 24V operation
- Provide a full day use of normal operation
- Capable of 150A delivery
- Easy to recharge and manage
- Low cost

To meet these requirements, lithium polymer batteries were chosen. When operating in a 6S configuration, they would provide 22.2V – 25.2V depending on charge state, which falls within specification of the ESC and motor. Many standard regulators and chargers are also able to work within this voltage range.

As the power curve of the motor was not known, it was difficult to estimate the required battery capacity to provide a full day’s use. The C-Arm does have many movements during a surgery, but the duty cycle of these movements is very low, compared to the amount of the time the C-Arm is stationary as was observed clinically during several orthopedic surgeries.
At a stall current of 40A, which can be approximated as the starting current of the motor, the total draw would be 120A. With an observed duty cycle of ~10%, this would give a constant draw of 12A from the battery pack. Over an 8 hour period, this would require 115Ah with a 20% buffer to avoid undervolting. From this capacity, it can be calculated that a discharge rating of 1.3C for a 150A peak draw would suffice.

![Image of battery](image)

**Figure 17 - Three of these 22.2V 8Ah Lithium Polymer batteries were used in parallel to power the Easy-C.**

Due to cost requirements for such a large battery pack, and the rough estimate of power requirements, a much smaller battery was chosen. A lithium polymer battery run in a 6S3P configuration (6 series, 3 parallel groups) was obtained. This battery had a capacity of 24Ah with a discharge rating of 15C, giving 360A peak draw capabilities. At the approximated constant current draw of 12A, this would provide 2 hours of use if drained fully at a cost of ~$300CAD.
The initial prototype used lead acid batteries for power delivery; however, the current prototype was upgraded to lithium polymer for the following reasons: higher energy density, low maintenance requirement, high current capacity, and cell longevity.

2.4 UI Design

The user interface for the Easy-C is ultimately intended to be used by the operator of the C-arm, but for this stage of the development, a simplified interface is needed focused on supporting the initial verification and validation testing. Development of a user-focused UI is deferred to a later stage of development. In the initial testing, the primary requirement is accepting basic motion commands from the developer.

Initially, a load cell with an input handle was considered as the main control method for the Easy-C. This load cell would allow for “feeling” the intention of the user without requiring movement of the input handle. I adapted a 6-axis load cell (ATI Inc.), which outputs forces and moments in 3 axes, and from this the desired direction of the Easy-C was calculated. The force commands in the horizontal plane were interpreted as a request to move the centre of mass of the C-arm in the desired direction, and the torque around the vertical axis as a request to rotate the C-arm around its centre of mass.

This input method was used on the scale prototype; however, it was deemed not appropriate for use on the Easy-C prototype for two major reasons: it required AC power which wouldn’t be easily accessible, and it was relatively bulky. The high price of the load cell (~$10,000) also meant that it would not be a practical solution for the next stage of development.
Two movement command alternatives were therefore considered, both utilizing standard control stick potentiometers. Initially, a wireless version was integrated into the system so that the Easy-C could be operated from any location nearby the unit, without requiring direct contact with the system itself. However, this was later removed in favor of a wired unit after the unit experienced several unexpected movements which were discovered to be caused problems in the wireless communication link.

The final design used 2 gimbal sticks alongside 2 push buttons and a tethered line. This allows for direct control of the device without worry of wireless interruption, as well as simplifying the coding requirements for the Arduino. This did remove the ability for wireless operation, but it was deemed low priority considering potential future UI integration.

2.5 Software Design

In order to control the C-arm, code was written to process commands from the movement command unit and generate the appropriate motor outputs. In addition, the software was designed to monitor other aspects of the state of the system and to provide appropriate warnings or implement safety functions. For example, the system also monitored the voltage of the batteries, and the state of the electronic motor brakes. As outputs, there were status LEDs on the handle of the controller to give the user feedback, as well as the outputs to the motors themselves to drive the Easy-C. Several internal calculations had to also be developed to allow this transfer of user intent to Easy-C movement.
Because the initial microcontroller for the Easy-C was the Beaglebone Black board, Python was used for initial coding. However, upon switching to the Arduino, much of the code structure had to be transferred to C++. This allowed for a review of the code structure for the Easy-C to be finalized as described in the following section.

2.5.1 Code Structure

The main structure of the Easy-C code was based around relatively low level functionality; since I anticipated that future work (beyond the scope of this thesis) would focus on integrating a broader set of “Smart-C” technologies, we decided that the scope of this project would be limited to providing an Applications Programming Interface (API) for the main functions of the Easy-C, coupled with a skeleton high level controller application that would be sufficient to test the Easy-C functions.

![layout of the Easy-C control system](image)

**Figure 18 - Layout of the Easy-C control system. Relevant connections to the control handle and higher-level controller are also shown.**
After initialization of the system, including checks of system voltages and brake statuses, the Arduino would read the user inputs. This included the arm state of the system based on the push buttons, and if held high would allow movement of the Easy-C system based on the motor mapping calculations. However, if this switch was ever released (configured as a dead man’s switch), the Easy-C would not react to any inputs.

Several sub-functions were provided as part of the high-level controller, such as entering a menu to configure speed and steering (pivoting) location. Three motor encoders were also installed onto the Easy-C to read wheel rotation for future closed loop control; however, they were not used within the main code structure in the current design iteration.
A nudge feature was also implemented on the Easy-C. This would enable the RT to make small movements of a fixed distance without having the guide the entire C-Arm. In open loop
operation, this nudge distance was set to approximately 1cm. The system will ultimately be capable of smaller increments; however, I discovered in pilot testing that, since nudge movements operate at close to the minimum speeds of the motors, actual distance each motor moves can deviate from the commanded value at these lower operating limits due to factors such as minor differences in frictional losses between motor drives and assemblies. As such, the Easy-C does not currently move accurately in the requested directions for small increments. Under closed loop control, this performance and resolution could potentially be improved upon, possibly down to millimetric accuracy, but adding this capability is out of scope for this thesis.

2.5.2 Closed Loop Feedback

Several sensors were integrated into the system to support future closed loop functionality. These included an Inertial Measurement Unit (IMU) with gyro, accelerometer, and compass, as well as three motor encoders installed on the outputs of the motors. By fusing these outputs of these sensors using a Kalman filter or alternative state estimator, it is expected that the Easy-C could operate under closed loop operation. However, as a higher level control system and feedback is expected to be added in a future development, I decided that I would initially evaluate the performance of the Easy-C system in open loop mode and only revisit this if I had significant difficulties verifying the core functionality of the system.

2.6 Safety Systems

As the Easy-C will be installed onto a C-Arm, there are several safety requirements that need to be considered for operation and use. Total operational weight will be close to 900lbs (~410kg) with a power-driven base. While the system can only attain a hard-capped speed of
approximately 1.4m/s, this still represents a significant amount of inertia that could potentially cause serious harm. In addition, any unexpected movement could potentially lead to injury or damage. A Risk Analysis was conducted to review the probability and severity of several failure methods, and various mitigations were incorporated to lower the likelihood of injury or damage occurring.

One failure method is loss of power while traveling at 1.4m/s, potentially causing the unit to tip over or skid. Analyzing the impulse on the system, a sudden stop (over 0.25s, could vary based on drive train inertia and backlash) would impart a force of 1960N. With the C-Arm weighing 3434N (not including the Easy-C weight) and approximating a coefficient of friction of 0.5, the friction would be 1717N. Slipping may occur, though it would be dependent on final total weight and the variability of the floor’s coefficient of friction.

The purpose of applying this Risk Management analysis to this prototype was to ensure the safety of the personnel involved with the device and the area in which testing would occur. At the present stage of development, the device is not intended to be operated by anyone other than those involved in its development.
Table 2 - Risk analysis of the prototype Easy-C system, with relevant mitigation options to ensure safe operation of the Easy-C.

<table>
<thead>
<tr>
<th>Task</th>
<th>Hazard</th>
<th>Probability</th>
<th>Severity</th>
<th>Risk score</th>
<th>Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating the Easy-C</td>
<td>Objects getting stuck and damaged in gears and drive chain</td>
<td>High</td>
<td>High</td>
<td>Immediately dangerous</td>
<td>Aluminum cover plates for ingress protection</td>
</tr>
<tr>
<td></td>
<td>Controller disconnecting during use</td>
<td>Low</td>
<td>High</td>
<td>Medium risk</td>
<td>Dead man switch with software disable</td>
</tr>
<tr>
<td></td>
<td>Easy-C does not respond to controller input</td>
<td>Low</td>
<td>High</td>
<td>Medium risk</td>
<td>Two failsafe pull plugs, one removes all power from the system and other activates motor brakes to stop movement.</td>
</tr>
<tr>
<td></td>
<td>Low battery damage</td>
<td>Medium</td>
<td>Low</td>
<td>Low risk</td>
<td>Low battery cutoff, disengage motors and can push manually</td>
</tr>
<tr>
<td>Overcurrent of motors and/or controllers</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>Low risk</td>
<td>Current limiting of speed controllers, inline fuses per motor</td>
</tr>
<tr>
<td>Deadman’s switch released during motion</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td>Medium risk</td>
<td>Build in acceleration</td>
</tr>
</tbody>
</table>
Emergency movement required, but power dies

<table>
<thead>
<tr>
<th>Risk Level</th>
<th>Low</th>
<th>High</th>
<th>Medium Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disengage motors for manual motion, process takes ~10s</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Device left powered after use, draining batteries

<table>
<thead>
<tr>
<th>Risk Level</th>
<th>Low</th>
<th>Low</th>
<th>Low Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>No current mitigation, a Battery Management System (BMS) would need to be implemented</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Battery charging could lead to fire or other damage

<table>
<thead>
<tr>
<th>Risk Level</th>
<th>Low</th>
<th>High</th>
<th>Medium Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batteries are always observed while charging, and are balance-charged to maintain cell integrity</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Several safety checks were also conducted as per the initial risk analysis in Table 2. All safety and failure methods that could be evaluated were passed.
Chapter 3: Development

Following the system component selection process described above, I carried out a CAD-based design process to specify the mechanical chassis components. During the process of designing the Easy-C chassis, several key factors had to be considered to address the design goals of the Easy-C system, including the loading and deflection characteristics of the fabricated parts. This was conducted with the use of FEA tools. I then fabricated or obtained the various components, assembled the system, and mounted the Easy-C to the C-Arm for verification testing. This chapter describes the design, fabrication and mounting steps.

3.1 CAD Design

Initially, a CAD model of the Siemens Orbic C-Arm was purchased online to allow for design work to be completed. This model contained all the primary parts of the C-arm; however, it was designed in software that made it incompatible with Solidworks (i.e. the objects were tessellated), and I discovered that some key dimensions were incorrectly specified. I therefore decided to build a simplified model of the C-arm from scratch using measurements I obtained directly from the existing C-arm.

The mounting method of the Easy-C needed to be compatible with the physical design of the C-Arm - no physical alterations to the existing device could be permitted. Complying with these restrictions required modeling of the base metal structure and undercarriage of the C-Arm.

The tipping point of the C-Arm had to be considered as well since the wheel contact points were to be changed to accommodate the omniwheels. The new wheel locations were kept as close as
possible to the current wheels, while minimizing the total “lift” that the device was getting off the floor.

Mounting of the Easy-C to the C-Arm was one of the most important design considerations to be made. Since no physical modifications could be made to the C-Arm itself, the Easy-C system was designed to sit underneath the C-Arm. The main crossbeam had the main chassis of the C-Arm resting on it and was held in place by mechanical restrictions, including adjustable spacers and straps between the Easy-C components and the C-Arm components, none of which required any physical modifications to the C-Arm (see Figure 27 physical mounting). The front wheel was held in place by the geometry of the C-Arm front wheel gantry and strapped into place.

Several brackets were required for mounting the electronics to the Easy-C assembly. This included the electronic speed controllers, microcontroller, and input handle, among several other components (see Figures 20/21). The batteries were strapped to the existing superstructure of the C-Arm, and no other electronics were directly mounted to the C-Arm.

The distance from the motors to the wheel assemblies was specified so as to require an integral number of chain links, with minor tensioning available through an adjustable motor offset on the mounting brackets. This was designed based on the size and pitch of the chosen spur gears and chain.
Figure 20 - Final CAD design of the Easy-C system with the chain guards installed.

Figure 21 - Final CAD design of the Easy-C system without the chain guards installed.
3.1.1 Stress Analysis

Each plate was assessed under representative loading cases using Solidworks Stress Analysis software in order to determine maximum stresses in individual parts at locations of potential failure or unacceptable deflection. If locations were identified to have too high stresses or too large deflections, the parts were modified to address these issues. The parts assessed were those that carried the primary loading of the C-Arm, for which there were four main parts that carried this loading. The wheel assemblies contained these parts, as illustrated in Figure 22.

Boundary conditions and force applications were applied based upon local loading on every part, with a maximum deflection value of 1mm and stress lower than the yield stress of 304 stainless steel (215MPa). The force applied was based on the total weight of the C-Arm (~3434N) to allow for worst case loading scenarios. Boundary and loading locations for the crossbeam can be seen in Figures 23 and 24, with the remaining parts in question shown in Appendix A.

Table 3 shows the final max stress and deflection values for the four wheel assembly and chassis components. These parts were all acceptable under the prescribed conditions and were fabricated as such.
Several components were designed and analyzed for stress and deflection.

Table 3 - Stress and deflection analysis of primary loading structural components. Max stress for the 304 stainless steel is 215MPa, with a maximum acceptable deflection of 1mm.

<table>
<thead>
<tr>
<th>Component</th>
<th>Max Stress (MPa)</th>
<th>Max Deflection (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Side Plate (A)</td>
<td>165 MPa</td>
<td>0.1 mm</td>
</tr>
<tr>
<td>Front Plate (B)</td>
<td>29 MPa</td>
<td>0.002 mm</td>
</tr>
<tr>
<td>Rear Plate (C)</td>
<td>9 MPa</td>
<td>0.0006 mm</td>
</tr>
<tr>
<td>Crossbeam (D)</td>
<td>115 MPa</td>
<td>0.6 mm</td>
</tr>
</tbody>
</table>
Figure 23 - FEA stress analysis for the crossbeam. The max stress (115 MPa) is lower than the yield stress (215 MPa). Maximum loading was seen near the boundary conditions (green arrows). Red arrow designates direction of gravity.

Figure 24 - FEA displacement analysis for the crossbeam. Deflection numbers are within expected ranges, with the maximum deflection occurring at point of force application (pink arrow). Red arrow designates direction of gravity.
3.1.2 Device Fabrication

Fabrication was completed by external machinists (operating out of BC Cancer) to specified drawings. Several minor changes were made from the CAD model based upon machining requirements and the experience of the machinists. The required electronic component mounting brackets were 3D printed from PLA material with a layer height of 0.15mm and a 25-50% infill, depending on the part.

3.2 Device Assembly

Once all parts were fabricated or obtained, assembly of the Easy-C was performed. As per the initial CAD model and design, the mechanical components were assembled. A wooden structure was created to bridge the main wheel assemblies to the front wheel assembly for initial testing and can be seen in Figure 25.

The electronics were mounted and soldered, utilizing 8AWG wire for all power train components and 22AWG for signal lines. All signal wiring was kept shielded for long runs, but around the controller the wiring used unshielded plugs for ease of switching, dependent on connection requirements. Wiring the device was done with extra length retained for prototyping purposes, in case changes needed to be made later.

The control handle used 3 potentiometers and 2 push buttons for input, with RGB LEDs for system state feedback. This was assembled with 3D printed parts and a shielded CAT5 wire to the main microcontroller.
Once assembled, the Easy-C system was flashed with the latest code developed on the test vehicle. From there, modifications were made based on the driving characteristics, with the major addition being the implementation of programmed acceleration ramps for the Easy-C. Due to the large amount of inertia in the system, instant changes in requested velocity would result in large torque spikes and jerky movements.
Several safety tests were also completed to ensure no damage or harm would occur if a failure presented itself. This included disconnecting signal lines from the control unit, between the electronic speed controllers and the microcontroller, and removing power from the microcontroller. If these did occur, the system would default to a zero-motion state through software (dead-man’s switch). The default configuration for all electronic speed controllers was neutral.

Once the basic operation of the Easy-C system was verified, the C-Arm could be installed. To accomplish this, a wooden ramping system was fabricated to elevate the C-Arm. This, combined with the use of a jack, allowed the Easy-C to be installed onto the C-Arm. This ramping system can be seen in Figure 26.
Figure 26 - For mounting the Easy-C system to the C-Arm, ramps were fabricated to lift the C-Arm to the desired height so the Easy-C could be installed. The front wheel was levered up for installation.

The Easy-C system was installed onto the C-Arm and the same acceleration testing steps were completed as with the wooden chassis to ensure proper operation. The acceleration was tuned to minimize shaking with still retaining adequate control by changing the strength of the acceleration ramps. The final assembly is shown in Figure 27.
Figure 27 - Easy-C mounted to the C-Arm.
Chapter 4: Verification Testing

With the Easy-C system assembled and operational, verification of its functionality needed to be completed. The clinically relevant motions had to be determined, scripted, and then run while being recorded with a tracking system. This provided data on the motion of the Easy-C platform for final verification.

4.1 Verification Methodology

Several clinically relevant movements were used to verify the operation of the Easy-C. These movements were determined based on observing clinical usage of C-Arms as well as querying surgeons and RTs using the C-Arm in surgery. The verification movements consisted of three main movements: lateral translation, forward/backward translation, and pivoting about the midpoint of the ‘C’.

Since I had not implemented a closed loop position control system, I executed these movements by programming a change in requested velocity for a prespecified amount of time. This input signal was generated over a period of 8 seconds, including 1s of acceleration and deceleration for each movement. Once the Easy-C finished its 8 seconds of movement, it paused in the “middle” location for 4 seconds, before moving in the reverse direction for the final 8 seconds to return (nominally) to its original position.

The first motion was a lateral translation (X direction). This movement replicates typical C-Arm movements used in a femoral IM nail implant procedure. The x-ray shots needed for guidance of
the nail down the femur requires the C-Arm to be moved from approximately the hip to the knee, with shots being taken along the length of the movement.

Figure 28 - Movement along the operating bed in the X direction, mimicking IM nail implant guidance.

Figure 29 - Easy-C at the start of the lateral translation, after a movement to the right, and after its return to the original location. Ideally, the end location would coincide with the starting position.
The second movement was in the Y direction, or in and out from the operating table. This mimics the movement an RT would have to perform several times during a surgery, depending on when the C-Arm is needed in the operating zone, as well as at the start and end of the surgery.

Figure 30 - Movement along the operating bed in the Y direction, mimicking the C-Arm moving in and out from the operating space.

Figure 31 - Easy-C at the start and end of the forward translation movement and following its return.
The third and final motion was a pivot about the midpoint of the ‘C’ of the C-Arm. This movement is not typically performed in surgery because it is not easily performed with a standard C-arm, but it could be quite helpful when the current alignment of the C-Arm is correct for the picture or video required, but another tool or piece of equipment needs to be placed in the space the C-Arm base currently occupies. On a standard C-Arm, this movement is very difficult to do, but with the holonomic wheels, the effective point of rotation can be programmed to lie anywhere relative the C-Arm. This movement therefore demonstrates a unique movement capability enabled by the Easy-C design.

Figure 32 – Rotation of the C-Arm about its imaging beam, mimicking achieving an optimal position in the operating environment.
Figure 33 - Easy-C at the start of a rotation movement, after shifting the base clockwise relative to the front pivot location, and after returning to the original position.

An Optotrak optical position measurement system was used to accurately measure the location of the C-Arm in space throughout each scripted movement, with two sets of active triad markers mounted on the source-collector axis of the C-Arm. This coincides with the pivot point of the Easy-C omniwheels (as initially referenced in Figure 6), aligning the rotating axes of the triads and that of the C-Arm. This location was chosen as it represents the imaging axis of the C-Arm, the most important to track for accurate movement.
Each scripted motion was run 10 times to allow for repeatability testing and accuracy. The origin of motion was zeroed before restarting the scripted run, so all the runs are measured relative to each movement’s origin, though this location on the floor changed for each run due to the residual errors from the previous run.
4.2 Verification Results

After mounting the Easy-C system to the C-Arm and running through the sets of scripted movements, the data from each movement was analyzed to estimate the average and standard deviation at the middle and end points of the movement, as well as the relative standard deviation and relative error. The mean value and standard deviation were plotted for each axis for each movement (see three following figures and tables).

The relative standard deviation was calculated for the primary direction of movement. The relative errors were calculated for the other axes of movement with respect to the primary direction of movement. To facilitate comparing errors in rotation (expressed in degrees) to the linear motions (expressed in millimeters), rotation errors were converted to effective translational errors by multiplying the angular errors by a characteristic length of the Easy-C – namely, its long axis (~1200 mm). Relative errors assessed following the return motion were divided by twice the outward/forward travel distance.
Figure 35 - Single run of X movement.

Figure 36 - Primary movement along the X axis of the C-Arm (~800 mm displacement).
Table 4 - Primary movement along the X axis to the middle setpoint.

<table>
<thead>
<tr>
<th>Setpoint</th>
<th>Mean Value</th>
<th>Standard Deviation</th>
<th>Relative SD / Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>X Movement</td>
<td>772.6 mm</td>
<td>29.9 mm</td>
<td>3.9% (RSD)</td>
</tr>
<tr>
<td>Y Movement</td>
<td>76.0 mm</td>
<td>8.5 mm</td>
<td>9.8% (RE)</td>
</tr>
<tr>
<td>Z Rotation</td>
<td>2.6 deg</td>
<td>1.0 deg</td>
<td>7.0% (RE)</td>
</tr>
</tbody>
</table>

Table 5 - Primary movement along the X axis to the end setpoint.

<table>
<thead>
<tr>
<th>Setpoint</th>
<th>Mean Value</th>
<th>Standard Deviation</th>
<th>Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>X Movement</td>
<td>-26.7 mm</td>
<td>44.3 mm</td>
<td>1.7% (RE)</td>
</tr>
<tr>
<td>Y Movement</td>
<td>-9.5 mm</td>
<td>18.1 mm</td>
<td>0.6% (RE)</td>
</tr>
<tr>
<td>Z Rotation</td>
<td>5.7 deg</td>
<td>1.1 deg</td>
<td>7.7% (RE)</td>
</tr>
</tbody>
</table>

For the X movement of the C-Arm, mimicking movement along the length of the operating bed, the C-Arm moved a total of 772.6mm with a standard deviation of 29.9mm across the 10 scripted movements to the middle set point. The C-Arm drifted 76.0mm in the Y direction and 2.6 deg in Z rotation to achieve this motion, where both were expected to be zero.

Returning to the starting position, the C-Arm overshot slightly to a mean value of -26.7mm with a standard deviation of 44.3mm. The Y direction and Z rotation were -9.5mm and 5.7deg respectively. All these final values were expected to be zero. The Y error reversed itself upon returning to the original location, but the rotational error continued to grow.
Figure 37 - Single run in the Y direction.

Figure 38 - Primary movement along the Y axis of the Easy-C (~700 mm displacement).
The primary direction of motion for the second test was along the Y axis, representing the C-Arm being moved into the operating area and back out again. The total distance moved in this primary direction to the middle setpoint was 712.5mm with a standard deviation of 17.0mm across the 10 scripted movements. The system was expected to have zero motion in the X direction and Z rotation but had -84.7mm and 4.1deg of movement respectively.

Upon return to the original location, the C-Arm ended at 24.6mm with a standard deviation of 16.7mm in the Y direction. For the X and Z axes, these were 10.3mm and -0.5deg respectively. These values were also expected to all be zero. Both off axis errors tracked back towards their starting point when the C-Arm returned to its starting location.

Table 6 - Primary movement along the Y axis to the middle setpoint.

<table>
<thead>
<tr>
<th>Middle Setpoint</th>
<th>Mean Value</th>
<th>Standard Deviation</th>
<th>Relative SD / Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>X Movement</td>
<td>-84.7 mm</td>
<td>25.3 mm</td>
<td>11.9% (RE)</td>
</tr>
<tr>
<td>Y Movement</td>
<td>712.5 mm</td>
<td>17.0 mm</td>
<td>2.4% (RSD)</td>
</tr>
<tr>
<td>Z Rotation</td>
<td>4.1 deg</td>
<td>0.9 deg</td>
<td>12.0% (RE)</td>
</tr>
</tbody>
</table>

Table 7 - Primary movement along the Y axis to the end setpoint.

<table>
<thead>
<tr>
<th>End Setpoint</th>
<th>Mean Value</th>
<th>Standard Deviation</th>
<th>Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>X Movement</td>
<td>10.3 mm</td>
<td>5.9 mm</td>
<td>0.7% (RE)</td>
</tr>
<tr>
<td>Y Movement</td>
<td>24.6 mm</td>
<td>16.7 mm</td>
<td>1.7% (RE)</td>
</tr>
<tr>
<td>Z Rotation</td>
<td>-0.5 deg</td>
<td>1.2 deg</td>
<td>0.7% (RE)</td>
</tr>
</tbody>
</table>
Figure 39 - Single run of in-plane rotation.

Figure 40 - In-plate rotation of the Easy-C (~62 deg of rotation).
Table 8 - Primary movement the in-plate rotation axis $\omega$ to the middle setpoint.

<table>
<thead>
<tr>
<th>Middle Setpoint</th>
<th>Mean Value</th>
<th>Standard Deviation</th>
<th>Relative SD / Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>X Movement</td>
<td>251.8 mm</td>
<td>42.8 mm</td>
<td>19.4% (RE)</td>
</tr>
<tr>
<td>Y Movement</td>
<td>67.3 mm</td>
<td>24.1 mm</td>
<td>5.2% (RE)</td>
</tr>
<tr>
<td>Z Rotation</td>
<td>-62.1 deg</td>
<td>1.8 deg</td>
<td>2.9% (RSD)</td>
</tr>
</tbody>
</table>

Table 9 - Primary movement along the in-plate rotation axis $\omega$ to the end setpoint.

<table>
<thead>
<tr>
<th>End Setpoint</th>
<th>Mean Value</th>
<th>Standard Deviation</th>
<th>Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>X Movement</td>
<td>185.0 mm</td>
<td>119.0 mm</td>
<td>7.1% (RE)</td>
</tr>
<tr>
<td>Y Movement</td>
<td>85.5 mm</td>
<td>21.4 mm</td>
<td>3.3% (RE)</td>
</tr>
<tr>
<td>Z Rotation</td>
<td>-7.7 deg</td>
<td>3.0 deg</td>
<td>6.2% (RE)</td>
</tr>
</tbody>
</table>

The final movement was rotation about the pivot point of the C-Arm, programmed to be located at the imaging beam of the C-Arm, allowing the device to be optimally positioned in the operating environment. The middle setpoint was -62.1deg with a standard deviation of 1.8deg across the 10 scripted movements. There was movement of 251.8mm in the X direction and 67.3mm in the Y direction to achieve this rotation, both expected to be zero.

Upon return to the nominal starting location, the final Z rotation was -7.7deg with a standard deviation of 3.0deg. The X direction finished at 185.0mm and the Y direction at 85.5mm. All these values were expected to be zero. The off-axis errors varied run to run, as can be seen in Figure 40.
Figure 41 - Drift of the center of rotation during in-plane rotation for 4 selected movements. Point 1 is the start of the movement, 2 is the middle of the movement, 3 is the end of the movement. All 3 of these points should remain at the origin (0,0) during this movement.

During the scripted in-plane rotation motion, I observed significant drift of the nominal center of rotation (located at the source-detector axis of the C-Arm). This error was not consistent across runs. This error was plotted in the X-Y plane for 4 selected movements, which depict the general trend of the 10 total movements.
Figure 42 - The front wheel does not move correctly when compared to the expected motion of the C-Arm.

The red arc shows the expected motion for the front wheel, while the blue ‘X’ is where the image center should remain.

The location of the front wheel was also plotted in the X-Y plane over this same movement, showing a general trend of the front wheel not moving correctly during in-plane rotation movements.
Figure 43 - Small oscillations can be seen in the device trajectory as it moves in the X direction.

For the individual movements, a small oscillation was seen through the movement. Part of this movement from Figure 35 was zoomed in on to show this oscillation in Figure 43. This could be explained by each set of omniwheels having a moving contact center as they rotate and different rollers touching the ground, as shown in Figure 6.
Chapter 5: Discussion and Future Work

The purpose of this study was to develop and evaluate a powered retrofittable base that could markedly reduce the musculoskeletal injury risk for RTs. By motorizing all movements of the C-arm base, we have provided the ability to maneuver the C-arm using only small hand movements applied to a joystick-like control box and without relying on the RT applying large loads to the C-arm itself.

5.1 Research Questions

Using the controller inputs, the force requirement to move the C-Arm was essentially eliminated. The controller is very easy to operate with one’s fingers as a handheld controller. Since the power system is doing all the force application, the user must do very little to induce motion. In addition, as the controller is on a tether, the user can be in various locations around the C-Arm while moving the device, allowing for potential areas of collision to be avoided and to observe the exact amount of movement occurring. Between this reduction of force and more natural posture, the related REBA and RULA scores fall to the lowest risk score bucket seen in Table 1, to negligible risk, no action required. I therefore conclude that the Easy C design would substantially eliminate the load-related risks in operating conventional C-Arms.

The verification process conducted was used to show that the functionality of the C-Arm is retained with the addition of the Easy-C. This was completed through three clinically relevant scripted movements of the C-Arm with the Easy-C system installed.
To mimic C-Arm movement along the length of the operating table, scripted runs along the Easy-C X-axis were conducted. With this primary motion, there was a small amount of drift in both secondary axes (9.8% in the Y direction, 7.0% in the \( \omega \) direction) for the motion to one end of the table, with the return journey seeing the Y drift reduced (0.6%) with the \( \omega \) error increasing slightly (7.7%). The primary drive direction error was low at 1.7% upon returning to its starting location. The motion was repeatable across the ten scripted runs with low standard deviations across all axes.

The X-direction motion was successful in mimicking movement along the operating table as all ten scripted runs had reasonably low error across all three axes (up to an average of close to 80 mm in the orthogonal direction). Given that these were scripted runs in open loop operation, once a closed loop system is implemented, I anticipate that these errors could be reduced substantially. In practice, even without a closed loop controller, an operator driving the Easy-C would likely be able to control its movements using some form of joystick-like input device and would be able to compensate during a movement for any drift that might occur, so I do not believe that the errors I observed will be significant in practice.

During the Y-axis movement, which replicates pulling in and out from the surgical zone, the off axes errors were larger than for the X-axis movement at 11.9% in X and 12.9% in \( \omega \) for the initial movement into the table. These errors were minimized once the Easy-C returned to its starting location with 0.7% in X and 0.7% in \( \omega \). In Y over this entire motion, the error remained relatively low at 1.7%.
As with the X movement, the Y movement is repeatable with relatively low errors in all axes. There is moderate deviation in the off axes, but, as with the X movement, this would be further minimized with a closed loop system or corrected for if the operator were using a joystick-like control. Under its current capabilities, the motion in the Y direction is acceptable.

When the Easy-C system is driven purely in the Y direction in the forward direction, the front wheel is not driving but simply passively translating on the rollers of the omniwheels. When driving forward over obstacles (such as wires or door sills), the bump-over radius is that of just a single roller, instead of the radius of the wheel when going over these obstacles in the drive direction. It is recommended that any movement over a known obstacle such as a sill or gap be taken at an angle or in reverse to minimize the force being applied to the rollers.

The final motion to be tested was \( \omega \) movement, a motion required to move the ‘C’ out of the way of instrumentation or tools in the surgical space while retaining the current image generated by the C-Arm. On the initial rotation, there was significant drift of 19.4% in X and a smaller amount of 5.2% in Y. The error in the X direction did not improve when the C-Arm returned to its starting position, finishing at 7.1%. The standard deviation was also large in the X direction as the motion was not repeatable. The Y axis error finished at 3.3% and 6.2% in \( \omega \); both standard deviations for these axes showed repeatability with small errors.

As Figure 41 shows, the center of rotation tends to wander during the in-plane rotation \( \omega \) movements. This wander is not consistent as the final position for the center of rotation ends in different locations depending on the run even though the scripted code is identical. The front
motor for these movements has a small input request, and therefore a low requested velocity as it is close to the center of rotation. Because of this low input, the motor likely cannot repeatably make the small movements necessary to drive at the requested speed, causing the Easy-C to wander more during this movement than the others I evaluated. This is evident in Figure 42, when the front wheel follows an erratic path.

These deviations also were evident during all three of the verification movements where off-axis movement was present, and when the Easy-C did not return to its exact original position. This most likely occurs because of various and differing losses within each drive chain and are likely primarily composed of frictional losses within the bearings and other moving components. Differing states of wear on the motor brushes, which causes varying power outputs, may also explain the wandering of the Easy-C. These losses may also not be symmetrical between forward and reverse.

Overcoming frictional losses, in addition to the large inertia of the system, does come with inherent control difficulties. To move a system a small amount while requiring a large force to overcome the initial static friction means that the energy supplied to the system as the device starts moving will create a burst of momentum that then needs to be quickly dissipated. This may pose some challenges in a future closed-loop design, at least for smaller movements.

Small oscillations in the movement of the C-Arm were present during the motion as seen in Figure 43, and, while they may not have a large direct impact upon accuracy with open loop operation, they may negatively affect the operation of a closed loop system. A possible cause of
this is due to the ordering of the wheels in a wheel set. In their current setup, the positions of the rollers alternate from one wheel to the next, which, during a rotation, shifts the effective contact point based on which rollers are touching the ground. A possible solution to this is to install them in a mirrored order, causing the effective contact point to remain in the same location.

Nonetheless, overall, I feel that the errors in movement I observed are within clinically acceptable limits, as the Easy-C does generally track in the appropriate directions from the given input. In future development, combining this with a closed loop feedback system would remove many of these errors and compensate for the varying losses or discrepancies between the different wheel subsystems within the drive system.

A secondary consideration of the Easy-C platform is unpowered operation, which could occur if the batteries become depleted or there is a fault in the electrical system that stops the Easy-C from being powered. To allow free movement, each motor has a disengagement knob. When manually turned off, the Easy-C can be moved manually in the same method as a standard C-Arm, though with the ability to push the C-Arm in any direction (vs the standard “shopping cart”). If the front wheel remains locked, the C-Arm can be maneuvered in a reverse “shopping cart” motion which may allow for easier tracking around corners as the front wheel could then provide some of the cornering force.

5.2 Relation to Other Literature

The MEPUC system from Suhm (2003, 2004) is the only other similar system with which I am familiar that has been documented in the literature. The MEPUC and Easy-C systems are both
retrofittable powered bases that attach to a C-Arm, though their structure differs significantly. In particular, while the MEPUC uses caster wheels alongside pivoting drive wheels for movement, the Easy-C utilizes 3 sets of holonomic wheels to allow for full range of motion. This makes the Easy-C system more compact and lighter. Also, the MEPUC system requires more space at the front of the C-arm to accommodate its motors and steering units, which most likely results in interference issues with the bed or instrumentation in the surgical zone, though this is not discussed in their journal articles. A rough estimate puts the Easy-C at 20% of the footprint of the MEPUC.

In both devices, the controller is separate from the main unit, allowing for remote movement of the C-Arm. This allows the RT or even the surgeon to move the C-Arm from various locations, not being restricted to movement by pushing the C-Arm itself.

The MEPUC system has apparently implemented a closed loop control system, along with a tracking system utilizing external cameras that allows the C-Arm to know where it resides in space relative to its previous positions. This allows for the MEPUC system to move to stored positions but is restricted by requiring extra external equipment and potential line of sight issues. I anticipate that the Easy-C platform can be augmented with similar onboard technologies in future. This onboard technology allows for integrated flow tracking, and would minimize footprint increase while giving full base tracking, no external equipment would be required.

For unpowered motion, the Easy-C simply requires disengaging the 3 drive motors in about 10 seconds, and then the C-Arm can be manually pushed. In emergency situations when the battery
is dead, the C-Arm can be quickly moved. From my understanding of the 3 pivot / 3 drive system of the MEPUC, this unpowered motion would be difficult to achieve as the motors could no longer pivot for required direction changes; however not enough is known about the system to make an accurate comparison.

Unfortunately, there does not appear to be any available verification or validation data published describing the performance of the MEPUC. The primary journal articles describing MEPUCC (Suhm 2003, 2004) discuss the potential capabilities and functionality of the system but do not present supporting data. Therefore, all these comparisons can only be completed on a qualitative level.

Table 10 - Comparison of Easy-C and MEPUC systems.

<table>
<thead>
<tr>
<th>Verification</th>
<th>External Tracking</th>
<th>Omni Unit size</th>
<th># of Actuators</th>
<th>Unpowered Movement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy-C</td>
<td>Yes</td>
<td>Onboard, to be implemented</td>
<td>Yes</td>
<td>Small footprint</td>
</tr>
<tr>
<td>MEPUC</td>
<td>Not Available</td>
<td>Optical tracking</td>
<td>No</td>
<td>Large footprint</td>
</tr>
</tbody>
</table>

In summary, I feel that the Easy-C platform has demonstrated a level of open-loop performance that would likely be clinically acceptable. Its overall size is smaller than that of the MEPUC device, and likely lighter and less expensive. The closed loop capabilities of the MEPUC are likely readily added to the Easy-C.
5.3 Limitations

The major limitation to the current Easy-C platform is the open loop functionality. This resulted in moderate levels of unintended drift in non-primary axes during the standard movements I evaluated. When the input signal is not large enough to overcome the losses within the drive, this causes the Easy-C to move with some errors. This open loop operation also means no relative position tracking is available to the system and it therefore cannot return accurately to pre-defined locations. However, another student in our lab has developed an on-board optical tracking system (OPTIX) to address this problem (Haliburton 2017) and I expect this technology will be integrated in the next iteration of the Easy-C.

There are also some minor issues with the current hardware design. In particular, the Arduino microprocessor I am using has limited processing speed. During testing, I sometimes encountered issues related to asking the system to handle interrupt service requests while performing pulse-width modulation signal generation; these occasionally caused the system to behave erratically and unexpectedly. These issues were substantially resolved by shifting from wireless to wired control, but I nonetheless recommend that this microcontroller be upgraded for future revisions of the Easy-C.

A final issue relates to the current mounting process for the Easy-C system. While the system was designed to fit several different models of C-Arms without requiring physical modifications to them (thereby preserving their clinical certifications), it currently requires some manual ‘fiddling’ to make everything fit properly. While acceptable for prototyping, this would not be
acceptable if testing the device in a live clinical environment was conducted, so another round of interface design would be required prior to such testing.

5.4 Recommendations and Future Work
Throughout the development and testing of the Easy-C, several areas of improvement were noted for consideration for future development of the Easy-C system, as well as integrations with other available technologies that would improve the performance of the system.

5.4.1 Improvements to Current Platform
During the design of the Easy-C power system, a chain and sprocket drive were chosen for their ease of acquiring and integration. Specifically, the chains could be customized in length to easily accommodate the distance between input and output shafts. However, there are several inherent issues with chains revolving around backlash and maintenance requirements. While these issues were not observed to have a direct impact on the current system, upgrading this drive to a timing belt and tensioner system would remove these issues.

During motion of the Easy-C platform, uneven surfaces could cause loading to the individual wheels of each set. Under worst case loading scenarios, these loadings could exceed the rated loading of the wheel, potentially causing deformation or failure. To compensate, the wheel assembly could include a pivoting wheel set or individually floating wheels allowing uneven loading to be distributed more evenly. A change to higher rated omniwheels could also be a consideration, depending on compatibility.
To reduce the oscillations seen in Figure 41, changing the omniwheels from an alternating pattern to a mirrored pattern may reduce this error due to changing effective drive lengths. No major modifications would be required for this, just a disassembly and reassembly of the wheel set.

5.4.2 Integration with Broader Technology

We feel that implementing some form of closed loop control is an important next step for the Easy-C platform. In its simplest form, we would simply integrate the existing wheel encoders into the signal processing system and rely on the sensed wheel positions rather than the commanded ones. However, while such a system would detect an absence of wheel rotation, it would not be able to detect slow accumulations of error during rolling; for that, we would need to implement a more accurate position system such as the aforementioned OPTIX system.

With closed loop capability, we could explore and implement additional capabilities, such as remembering previous positions, moving accurately around the OR table, and sharing control with the surgeon.

In our current work, we did not provide a fully functional user interface. An important next step, therefore, is to design and implement such a system in order to facilitate, first, lab-based evaluations in a mock OR and, subsequently, initial clinical trials. Moving to the clinic would also require a usability and safety evaluation by the hospital biomedical engineering staff prior to deployment.
5.5 Conclusion

As image guided minimally invasive surgeries continue to grow in use, the utilization of machines such as the C-Arm will also grow. With this comes the increased risk of injury to radiation technologists moving this heavy piece of machinery throughout the day.

With the Easy-C, a retrofittable power assisted base for the C-Arm, this strenuous activity could be offloaded from the RT so that they can concentrate on the task of accurately and efficiently positioning the C-Arm in the operating room. The Easy-C platform has shown its ability to function as expected with the C-Arm installed, which justifies further development along the lines outlined above: namely, adding closed loop control and a practical user interface.
Bibliography


Javier Moreno, Eduard Clotet, Ruben Lupiañez, Marcel Tresanchez, Dani Martínez, Tomàs Pallejà, Jordi Casanovas, Jordi Palacín (2016). Design, Implementation and Validation of the
Three-Wheel Holonomic Motion System of the Assistant Personal Robot. MDPI Sensors 2016, 16, 1658.


Appendices

The following Appendices were compiled to document the calculations and code that were required to design and operate the Easy-C. Appendix A primarily focuses on the hardware design and calculations, while Appendix B documents the Matlab code used to evaluate the verification data and the Arduino code that runs on the Easy-C itself.

Appendix A

Most of the calculations used to design the Easy-C components are documented in Appendix A.

Beam stress calculations

\[ D_{\text{max}} = \frac{F \times x}{(24 \times E \times I)(3L^2 - 4a^2)} \]

\[ I = \frac{M \times g \times x}{(24 \times E \times D_{\text{max}})(3L^2 - 4a^2)} \]

Where:

- \( M = \frac{350 \text{ kg}}{2} = 175 \text{ kg} \) (loading half the C-Arm to each F location)
- \( x = 0.1 \text{ m} \)
- \( E = 205 \text{ GPa} \)
- \( D_{\text{max}} = 0.003\text{m} \)
- \( L = 0.5\text{m} \)

\[ I = 175 \times 9.81 \times 0.1 / (24 \times 205 \times 0.003) \times (3 \times 0.5^2 - 4 \times 0.1^2) \]

\[ I = 172 \times 0.71 / 14.76 \times 10^9 \]

\[ I = 8.26 \times 10^{-9} \text{m}^4 = 8.255 \text{mm}^4 \]

\[ I = \frac{(a^4 - b^4)}{12} ; b = a - 4 \text{mm} \) (2mm wall thickness)

\[ 0 = -12 \times 8255 + a^4 - (a^4 - 4a^3 + 6a^2 - 4a + 256) \]
\( 0 = -99,316 + 256a - 96a^2 + 16a^3 \)

**a = 20.3mm**

Chose:

\( a = 1.5'' = 38.1 \text{mm} \)

\( b = 1.5'' - (2*0.12'') = 1.26'' = 32 \text{mm (\sim 3mm wall thickness)} \)

\[
\text{Sigma}_\text{max} = M*c/I = M*g*x*c/I
\]

\[
\text{Sigma}_\text{max} = 175 \text{kg} \times 9.81 \times 0.1 \text{m} \times 0.019 \text{m}/88.22 \times 10^{-9}
\]

**Sigma_max = 37 MPa**

**Max bending strength = Yield Strength} * 0.66 = 32000 psi * 0.66 = 21,120 psi = 146 MPa**

\[
\text{SF} = 146 / 37 = 3.9
\]

\[
\text{Dmax} = F*x/(24*E*I)(3L^2-4a^2) = 0.6 \text{mm}
\]

\[
\text{SF} = 3 / 0.6 = 5
\]

**Wheel size and RPM**

For walking speed (1.4m/s), desired wheel RPM

\[
\text{Theta} = V/r = 1.4/0.0625 = 22.4 \text{ rad/s} = 214 \text{ RPM}
\]
Chain Sizing Calculation

\[ T = I \cdot a \]

\[ T = \frac{1}{3} M L^2 \cdot a \] (treat C-Arm as thin rod rotating about its end)

\[ T = \frac{1}{3} \times 330 \text{kg} \times 1 \text{m}^2 \times 0.5 \text{rad/s} \]

\[ T = 55 \text{N.m} \]

\[ F = \frac{T}{L} = \frac{55 \text{N.m}}{1 \text{m}} = 55 \text{N} \]

\[ T_w = F \cdot r = 55 \text{N} \times 0.0625 \text{m} = 3.4 \text{N.m} \]

\[ F_c = \frac{T_w}{r_c} = \frac{3.4 \text{N.m}}{0.0254 \text{m}} = 135 \text{N} \]

\[ W = \frac{F_c}{a} = 135 \div 9.81 \text{m/s}^2 = 13.8 \text{kg} = 30 \text{ lbs} \]

Roller Chain Working Load = 190 lbs

\[ SF = \frac{190}{30} = 6.3 \]
Chain Length Calculations

![Chain Diagram]

Bearing Size Calculation

\[ F = M \times a \] (weight of C-Arm)

\[ F = 350\text{kg} \times 9.81\text{m/s}^2 \]

\[ F = 3434\text{N} = 772\text{ lbf} \]

Mass = 350kg = 770lbs

Single bearing radial static loading max = 1300lbs

6 bearings, but theoretical worst-case scenario spread across 3 bearings = 3900 lbs

\[ SF = \frac{3900\text{lbs}}{770\text{lbs}} = 5.1 \]
Shaft Size Calculation

Force calculation for the shaft under static loading, 7/8”.

MI for Solid Round Beams = \( \frac{\pi \times (OD)^4}{64} \)

\[
MI = \pi \times (\frac{7}{8})^4/64 = 0.0288 \text{ in}^4 = 1.2 \times 10^{-8} \text{ m}^4
\]

\[
\Sigma_{\text{max}} = \frac{M \times c}{I} = \frac{M \times g \times x \times c}{I}
\]

\[
\Sigma_{\text{max}} = 175 \times 9.81 \times 0.038 \times 0.011 / 1.2 \times 10^{-8}
\]

\[
\Sigma_{\text{max}} = 1.2 \times 10^8 \text{ Pa} = 64 \text{ MPa}
\]

Max bending strength = \( \text{Yield Strength} \times 0.66 = 75000 \text{ psi} \times 0.66 = 49500 \text{ psi} = 341 \text{ MPa} \)

\[
\text{SF} = 341/64 = 5.3
\]
Bolt size calculation

Calculation of forces on bolts

Shear Stress = $4*F/(\pi*d^2) = 4*3434N/(\pi*(0.0055)^2)$

Shear Stress = 144.5MPa = 20,958psi
Bolt shear stress = bolt tensile stress * 0.58 = 170,000psi * 0.58 = 98,600psi

For single bolt under this load:

SF = 98,600 / 20,958 = 4.7

**Ultra Pride 3 Speed Calculation**

Stall current ~40A @ 24V

Max Speed = 6.44 km/h = 1.79 m/s

Wheel Diameter = 0.254 m ; Radius = 0.127m

Output Shaft Rotational Speed = V/r = 1.79/0.127 = 14.1 rad/s = 135 RPM
Stress Analysis of Plates
Appendix B

Included in Appendix B is the method for mounting the Easy-C to the C-Arm. The code operating the Easy-C and the Matlab code used to analyze the Easy-C verification movements is also documented in Appendix B.

Easy-C Mounting

Initially, the C-Arm is rolled onto the wooden ramps to elevate the rear section. From there, the Easy-C rear section is slid underneath the C-Arm base. A jack is then installed at the furthest rear point of the C-Arm and the ramps removed, leaving the C-Arm resting on the jack. This jack is slowly let out until the C-Arm rests on the Easy-C main strut. Depending on the C-Arm, different spacers may have to be utilized. Remove the jack once the full weight of the C-Arm is on the Easy-C.

The front wheel requires using a lever (2x4s, one as lever arm and one as fulcrum) to tilt the C-Arm back. The front wheel assembly is slid into place, including the drive shaft, and may require different spacers depending on the C-Arm. The lever is released slowly until the front wheel is in place. Reverse this process for removal of the Easy-C.
Matlab Code

The code utilized to analyze the Easy-C verification data read in the information from each movement file, split it into various variables (X, Y, and Z rotation, alongside error). This data was then zeroed, analyzed, and plotted.

clear
clc
close all

for i = 0:9
    %Read data from files
    filename = "X Movements_2019_08_22_144656_01"+i+"_6d.xls"
    M(:,:,i+1) = dlmread(filename, '\t', 5, 0);
    %Split data into separate variables
    Frame(:,i+1) = M(:,1,i+1);
    X_1(:,i+1) = M(:,5,i+1);
    Y_1(:,i+1) = M(:,6,i+1);
    Rz_1(:,i+1) = M(:,2,i+1);
    Err_1(:,i+1) = M(:,8,i+1);
    X_2(:,i+1) = M(:,12,i+1);
    Y_2(:,i+1) = M(:,13,i+1);
    Rz_2(:,i+1) = M(:,9,i+1);
    Err_2(:,i+1) = M(:,15,i+1);
    %Zero the data sets to the first value
    X_1(:,i+1) = X_1(:,i+1) - X_1(1,i+1);
    Y_1(:,i+1) = Y_1(:,i+1) - Y_1(1,i+1);
    Rz_1(:,i+1) = Rz_1(:,i+1) - Rz_1(2,i+1);
    X_2(:,i+1) = X_2(:,i+1) - X_2(1,i+1);
    Y_2(:,i+1) = Y_2(:,i+1) - Y_2(1,i+1);
    Rz_2(:,i+1) = Rz_2(:,i+1) - Rz_2(1,i+1);

for j = 1:2400
    framezerodiff = Y_1(j,i+1) - Y_1(1,i+1);
    if framezerodiff >= 2
framezero(i+1) = j;
break
end
end

X_1(:,i+1) = circshift(X_1(:,i+1), -1*framezero(i+1),1);
Y_1(:,i+1) = circshift(Y_1(:,i+1), -1*framezero(i+1),1);
Rz_1(:,i+1) = circshift(Rz_1(:,i+1), -1*framezero(i+1),1);
X_2(:,i+1) = circshift(X_2(:,i+1), -1*framezero(i+1),1);
Y_2(:,i+1) = circshift(Y_2(:,i+1), -1*framezero(i+1),1);
Rz_2(:,i+1) = circshift(Rz_2(:,i+1), -1*framezero(i+1),1);
end

for j = 1:2400
    X_1_meanstd(j,1) = mean(X_1(j,:));
    X_1_meanstd(j,2) = std(X_1(j,:));
    Y_1_meanstd(j,1) = mean(Y_1(j,:));
    Y_1_meanstd(j,2) = std(Y_1(j,:));
    Rz_1_meanstd(j,1) = mean(Rz_1(j,:));
    Rz_1_meanstd(j,2) = std(Rz_1(j,:));
end
set(0, 'DefaultFigureColor', 'white');
grid on
hold on
x0=100;
y0=100;
width=1400;
height=900;
set(gcf, 'position', [x0, y0, width, height])
set(gca, 'FontSize', 20)
set(gca, 'LineWidth', 2);
title('Mean C-Arm Data')
xlabel('Time (centiseconds)')
ylabel('C-Arm Displacement (mm)')
xlim([0 2050])
ylim([-1000 1000])
p11 = plot(Frame(:,1),X_1_meanstd(:,1), '-b', 'LineWidth', 2);
plot(Frame(:,1),X_1_meanstd(:,1)-X_1_meanstd(:,2), '-b', 'LineWidth', 2);
plot(Frame(:,1),X_1_meanstd(:,1)+X_1_meanstd(:,2), '-b', 'LineWidth', 2);
p22 = plot(Frame(:,1),Y_1_meanstd(:,1), '-r', 'LineWidth', 2);
plot(Frame(:,1),Y_1_meanstd(:,1)-Y_1_meanstd(:,2), '-r', 'LineWidth', 2);
plot(Frame(:,1),Y_1_meanstd(:,1)+Y_1_meanstd(:,2), '-r', 'LineWidth', 2);
ppp = plot(Frame(:,1),Rz_1_meanstd(:,1), '-g', 'LineWidth', 2);
plot(Frame(:,1),Rz_1_meanstd(:,1)-Rz_1_meanstd(:,2), '-g', 'LineWidth', 2);
plot(Frame(:,1),Rz_1_meanstd(:,1)+Rz_1_meanstd(:,2), '-g', 'LineWidth', 2);
hold off
legend([p11 p22 p33], 'X Movement', 'Y Movement', 'Z Rotation')

Arduino Code

The Easy-C operates through an Arduino installed onboard. The block diagram for this
functionality can be seen in Figure 17 of the main body of the thesis.

#include <Servo.h>
#include "FastLED.h"
#define NUM_LEDS 16
CRGB leds[NUM_LEDS];
int valZ = 0;
int valX = 0;
int valY = 0;
int offsetZ = 0;
int offsetX = 0;
int offsetY = 0;
int potPin1 = 3; // yaw aka Z
int potPin2 = 2; // roll aka X
int potPin3 = 4; // pitch aka Y
int inputX = 1500;
int inputY = 1500;
int inputZ = 1500;
int ndgX = 1500;
int ndgY = 1500;
int ndgZ = 1500;
int sensorPin = A0;
int batteryvoltage = 1400;
int avgamt = 10;
int motortable1[] = {-1000,0,-450};
int motortable2[] = {-500,866,870};
int motortable3[] = {500,866,-870};
long motorval1 = 1500;
long motorval1old = 1500;
long motorval2 = 1500;
long motorval2old = 1500;
long motorval3 = 1500;
long motorval3old = 1500;
long motorvallimit = 1500000;
long PIDad1 = 0;
long PIDad2 = 0;
long PIDad3 = 0;
unsigned long currentTime = 0;
unsigned long startTIme = 0;
long driveTime = 250;
int driveStart = 0;
int encoder1 = 0;
int encoder2 = 0;
int encoder3 = 0;
bool motorarm = false;
bool lowbatt = false;
const int CSn1 = 49; // Chip select
const int CSn2 = 43;
const int CSn3 = 33;
const int CLK1 = 51; // Clock signal
const int CLK2 = 45; // Clock signal
const int CLK3 = 35; // Clock signal
const int DO = 48; // Digital Output from the encoder which delivers me a 0 or 1, depending on the bar angle..
const int SWT1 = 10; //aux
const int SWT2 = 11; //arm
int ArmTogg1 = 0;
int DirTogg2 = 0;
Servo myservo1; // create servo object to control a servo
Servo myservo2;
Servo myservo3;

void setup() {
    FastLED.addLeds<NEOPIXEL, 12>(leds, NUM_LEDS);
    Serial.begin(9600);
    pinMode(CSn1, OUTPUT);
    pinMode(CSn2, OUTPUT);
    pinMode(CSn3, OUTPUT);
    pinMode(CLK1, OUTPUT);
    pinMode(CLK2, OUTPUT);
    pinMode(CLK3, OUTPUT);
    pinMode(DO, INPUT);
    pinMode(SWT1, INPUT_PULLUP);
    pinMode(SWT2, INPUT_PULLUP);

digitalWrite(CLK1, HIGH);
digitalWrite(CLK2, HIGH);
digitalWrite(CLK3, HIGH);
digitalWrite(CSn1, HIGH);
digitalWrite(CSn2, HIGH);
digitalWrite(CSn3, HIGH);
offsetZ = 512 - analogRead(potPin1);
offsetX = 512 - analogRead(potPin2);
offsetY = 512 - analogRead(potPin3);
    for (int i= 0; i<8; i++) {
        leds[i].setRGB( (150), (0), (150));
    }
    for (int i= 8; i<16; i++) {
        leds[i].setRGB( (0), (150), (0));
    }
    FastLED.show();

delay(1000);
void loop() {
    currentTime = millis();
    valZ = analogRead(potPin1)+offsetZ;
    valX = analogRead(potPin2)+offsetX;
    valY = analogRead(potPin3)+offsetY;
    ArmTogg1 = digitalRead(SWT1);
    DirTogg2 = digitalRead(SWT2);
    inputZ = map(valZ,0,1024,1000,2000);
    inputX = map(valX,0,1024,1100,1900);
    inputY = map(valY,0,1024,2000,1000); //inverted!
    //inputY = map(val2,0,1023,1000,2000);
    //inputZ = map(val3,0,1023,1000,2000);
    if ( ArmTogg1 == 0){ //(inputX>1515) || (inputX<1485) || (inputY>1515) || (inputY<1485) ||
        (inputZ>1515) || (inputZ<1485) )){// && lowbatt == false) {
        if (motorarm == false) {
            motorarm = true; // attaches the servo on pin X to the servo object
            myservo1.attach(6);
            myservo2.attach(7);
            myservo3.attach(8);
        }
    }
    if (DirTogg2 == 0){
        if (driveStart == 0){
            if ((abs(inputX-1500)>100) || (abs(inputY-1500)>100) || (abs(inputZ-1500)>100) ){
                driveStart = 1;
                startTime = currentTime;
            }
            if ((inputX-1500)>100){
                ndgX = 1700;
                ndgY = 1500;
                ndgZ = 1500;
            }
            else if ((inputX-1500)<-100){
                ndgX = 1300;
                ndgY = 1500;
                ndgZ = 1500;
            }
            else if (inputY-1500>100){
                ndgX = 1500;
                ndgY = 1700;
                ndgZ = 1500;
            }
            if (driveStart == 1){
                if (currentTime - startTime > 10000){
                    driveStart = 0;
                }
            }
            if (driveStart == 1){
                if (motorarm == false){
                    motorarm = true;
                    myservo1.attach(6);
                    myservo2.attach(7);
                    myservo3.attach(8);
                }
            }
        }
    }
}
else if (inputY - 1500 < 100) {
    ndgX = 1500;
    ndgY = 1300;
    ndgZ = 1500;
}
else if (inputZ - 1500 > 100) {
    ndgX = 1500;
    ndgY = 1500;
    ndgZ = 1700;
}
else if (inputZ - 1500 < -100) {
    ndgX = 1500;
    ndgY = 1500;
    ndgZ = 1300;
}
else {
    ndgX = 1500;
    ndgY = 1500;
    ndgZ = 1500;
}
}
}
if (driveStart == 1) {
    if ($(currentTime - startTime <= driveTime)) {
        inputX = ndgX;
        inputY = ndgY;
        inputZ = ndgZ;
    }
    else if ($(currentTime - startTime > driveTime) && $(currentTime - startTime <= 2*driveTime)) {
        inputX = 1500;
        inputY = 1500;
        inputZ = 1500;
    }
    else {
        inputX = 1500;
        inputY = 1500;
        inputZ = 1500;
        driveStart = 2;
    }
}
}
if (driveStart == 2) {
    driveStart = 0;
    inputX = 1500;
    inputY = 1500;
inputZ = 1500;
}

motorval1 = (1L*inputX-1500+PIDad1)*motortable1[0] + (1L*inputY-1500+PIDad2)*motortable1[1] + (1L*inputZ-1500+PIDad3)*motortable1[2];
motorval2 = (1L*inputX-1500+PIDad1)*motortable2[0] + (1L*inputY-1500+PIDad2)*motortable2[1] + (1L*inputZ-1500+PIDad3)*motortable2[2];
motorval3 = (1L*inputX-1500+PIDad1)*motortable3[0] + (1L*inputY-1500+PIDad2)*motortable3[1] + (1L*inputZ-1500+PIDad3)*motortable3[2];

motorval1 = constrain(motorval1, motorvallimit, motorvallimit);
motorval2 = constrain(motorval2, motorvallimit, motorvallimit);
motorval3 = constrain(motorval3, motorvallimit, motorvallimit);
motorval1 = map(motorval1, -motorvallimit, motorvallimit, 1000, 2000);
motorval2 = map(motorval2, -motorvallimit, motorvallimit, 1000, 2000);
motorval3 = map(motorval3, -motorvallimit, motorvallimit, 1000, 2000);

// Averaging
motorval1 = (long)round(((avgamt-1.0)*motorval1old+motorval1)/avgamt);
motorval2 = (long)round(((avgamt-1.0)*motorval2old+motorval2)/avgamt);
motorval3 = (long)round(((avgamt-1.0)*motorval3old+motorval3)/avgamt);
motorval1old = motorval1;
motorval2old = motorval2;
motorval3old = motorval3;
motorval1 = constrain(motorval1, 1000, 2000);
motorval2 = constrain(motorval2, 1000, 2000);
motorval3 = constrain(motorval3, 1000, 2000);

myservo1.writeMicroseconds(motorval1);
myservo2.writeMicroseconds(motorval2);
myservo3.writeMicroseconds(motorval3);
}
else if (motorarm == true) {
motorarm = false;
myservo1.detach();
myservo2.detach();
myservo3.detach();
}
else {
}

/*
if (DirTogg2 == 0 & ArmTogg1 == 1){
if ((abs(inputX-1500) > abs(inputY-1500)) & (abs(inputX-1500) > abs(inputZ-1500))){
// X is being selected
*/
for (int i= 1; i<3;  i++) {
    leds[i].setRGB( (150), (0), (0));
}
for (int i= 13; i<15;  i++) {
    leds[i].setRGB( (150), (0), (0));
}
FastLED.show();

else if ((abs(inputY-1500) > abs(inputZ-1500))){
    //Y is being selected
    for (int i= 1; i<3;  i++) {
        leds[i].setRGB( (150), (0), (0));
    }
    for (int i= 13; i<15;  i++) {
        leds[i].setRGB( (150), (0), (0));
    }
    FastLED.show();
}
else {
    //Z is being selected
    for (int i= 1; i<3;  i++) {
        leds[i].setRGB( (150), (0), (0));
    }
    for (int i= 13; i<15;  i++) {
        leds[i].setRGB( (150), (0), (0));
    }
    FastLED.show();
}
*/

unsigned int readSensor1(){
    unsigned int dataOut1 = 0;
digitalWrite(CSn1, LOW);
delayMicroseconds(1); //Waiting for Tclkfe
//Passing 12 times, from 0 to 11
for(int x=0; x<12; x++){
    digitalWrite(CLK1, LOW);
delayMicroseconds(1); //Tclk/2
digitalWrite(CLK1, HIGH);
delayMicroseconds(1); //Tdo valid, like Tclk/2
dataOut1 = (dataOut1 << 1) | digitalRead(DO); //shift all the entering data to the left and past
the pin state to it. 1e bit is MSB
}
digitalWrite(CSn1, HIGH); //deselects the encoder from reading
return dataOut1;
}
unsigned int readSensor2(){
unsigned int dataOut2 = 0;
digitalWrite(CSn2, LOW);
delayMicroseconds(1); //Waiting for Tclkfe
//Passing 12 times, from 0 to 11
for(int x=0; x<12; x++){
    digitalWrite(CLK2, LOW);
delayMicroseconds(1); //Tclk/2
digitalWrite(CLK2, HIGH);
delayMicroseconds(1); //Tdo valid, like Tclk/2
    dataOut2 = (dataOut2 << 1) | digitalRead(DO); //shift all the entering data to the left and past
the pin state to it. 1e bit is MSB
}
digitalWrite(CSn2, HIGH); //deselects the encoder from reading
return dataOut2;
}
unsigned int readSensor3(){
unsigned int dataOut3 = 0;
digitalWrite(CSn3, LOW);
delayMicroseconds(1); //Waiting for Tclkfe

//Passing 12 times, from 0 to 11
for(int x=0; x<12; x++){
    digitalWrite(CLK3, LOW);
delayMicroseconds(1); //Tclk/2
digitalWrite(CLK3, HIGH);
delayMicroseconds(1); //Tdo valid, like Tclk/2
    dataOut3 = (dataOut3 << 1) | digitalRead(DO); //shift all the entering data to the left and past
the pin state to it. 1e bit is MSB
}
digitalWrite(CSn3, HIGH); //deselects the encoder from reading
return dataOut3;
}