ABSTRACT

Title of Document: DEVELOPMENT AND EVALUATION OF A

                                      FLEXIBLE DISTRIBUTED ROBOT

                                      CONTROL ARCHITECTURE

                                      Andrew Ellsberry, Masters of Science 2010

Directed By: Associate Professor Dr. David Akin
                                      Department of Aerospace Engineering

The communications and electronic systems that comprise a distributed control architecture for a robotic manipulator tie the high level control and motion planning to the electromechanical components. Custom solutions to this problem can be expensive in terms of time, cost, and maintenance. The integration of commercial off the shelf (COTS) motion controllers, combined with a robust communication standard, offers the potential to reduce the costs and development times for new robots. This thesis demonstrates an implementation of this architecture using commercial controllers and the CANopen communications bus on two existing dexterous robots. Testing is conducted to quantify the single joint performance of these modules. Additionally, the implementation of the system on a second robot arm was conducted in order to test the flexibility of the system for use with different actuators and feedback.

DEVELOPMENT AND EVALUATION OF A FLEXIBLE DISTRIBUTED
ROBOT CONTROL ARCHITECTURE

By

Andrew John Ellsberry

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2010

Advisory Committee:
Associate Professor David Akin, Chair
Professor Norman Wereley
Associate Professor Raymond Sedwick

Acknowledgements

This space is insufficient to acknowledge everyone who made this thesis possible, but here is an attempt.

I would like to thank my advisor Dr. David Akin for his support, guidance, and assistance over the years I have spent at the Space Systems Laboratory. I would also like to thank the other members of my committee, Dr. Norman Wereley and Dr. Raymond Sedwick, for their perspective and assistance at the end of this process.

Throughout this research, I have leaned (rather hard at times) on Nicholas D'Amore and his complementary research on the design and implementation of the high level controller that ran the arm at the multi joint level and interfaces with the operator with his amazing GUI. I cannot thank him enough not only for working with me, but for still talking to me at the end of the process.

I would also like to thank Dr. Mary Bowden for all her help over the last 6+ years. I can honestly say that I would not be where I am today without some of the help and guidance that she has patiently imparted.

I want to thank the rest of the SSL students and staff who either helped with this research or laid the foundation for it. Thanks also to Kate, Nick, Connie, Nick, Sharon, Kate, and Heather for helping with the prescreening of my thesis and presentation for my committee.

Table of Contents

List of Tables

List of Figures

List of Illustrations

If needed.

Chapter 1: Introduction

This chapter discusses the foundation of motor control systems and introduces the RANGER robotic system.

Motivation

A critical component to any robot is the electronic controllers that control the motion of the system. These controllers sit between the high level task and path planning and the actuators that comprise the physical robot. When functioning properly, they rarely define the system to the same level as the high level software, which determines the tasks the robot can complete, or the actuators that define the physical performance, but without highly capable controllers, the full potential of both these systems will never be realized. This research aims to develop a reusable architecture for controlling dexterous robotic systems that can reduce the cost and time to a fully functional robotic arm. By reducing the required resources, both human and monetary, that must be dedicated to the electronics for a robot, the focus can shift more towards the application of the robot, which is the primary focus for research laboratories and grants.

The Space Systems Laboratory developed and operates a family of dexterous robotic manipulators that are designed for both research and operational uses in extreme environments. The first generation of these robots is called RANGER NBV-I or Neutral Buoyancy Vehicle 1. This robot features arms for both dexterous tasks (for

interacting with the environment) and a camera arm originally designed to position a set of stereo cameras. The second generation was designed for a demonstration flight that was to fly on the Space Shuttle and is a more advanced version of the manipulators from NBV-I. The space rated version of this arm is called RTSX for Ranger Telerobotic Shuttle eXperiment. A nearly identical engineering unit for the RTSX program, NBV-II, has proven an indispensible workhorse for earth gravity and neutral buoyancy testing for various projects. A new version of the manipulators, SAMURAI, is developed for deep sea exploration and depth rated for 5 kilometers depth.[1]

Of the robots mentioned above, only NBV-II was operational at the inception of this research. The primary reason for this was a lack of functional motion control electronics. The NBV-I manipulators never reached their full potential due to controllers that were limited in comparison to the ones designed for NBV-II, resulting in the system being retired when NBV-II became operational. SAMURAI featured ambitious targets for electronics performance and miniaturization that resulted in the program being stalled for years. This thesis is based around the design and testing of a robot motion controller system that could provide comparable performance to the NBV-II electronics in order to resurrect the NBV-I manipulators and finish the SAMURAI project.

---

[1] More information on each of these manipulators can be found in Chapter 2

Requirements

Requirements Definition

The design goal of the project was to implement a distributed control architecture

using automation technology to operate in a similar manner as the local processing

units used in RANGER NBV-II. This breaks down into a number of definable

requirements that derive from either the RANGER system or the specifics of the

NBV-I camera arm. The solution should meet the following requirements:

1. Drive a brushless trapezoidal servo motor in the following modes

    a. Current

    b. Velocity

    c. Position

2. Operate off of a 28 – 35 volt power supply

3. Output a continuous current of 5-10 amps

4. Fit in the Ranger NBV-I manipulator and interface with the existing

    connections

5. Communicate with the higher level controller on a data bus capable of 1 Mbps

    or greater with a throughput of ~500kbps or greater[2]

6. Operate using Hall Effect sensors and incremental encoders

7. Be implementable at minimum cost and development time

8. System should be flexible to operate with other motor technologies

9. System should support rapid integration into different robotic systems in use

    or planned by the Space Systems Laboratory

---

[2] Based on the Space Systems Laboratory's past experience with data buses, see Chapter 3 for more details

10. System should be support manual reconfiguration of the robot such as adding

or moving joints

These requirements will be met by a combination of selecting the proper

components, testing and demonstration.

Chapter 2: Background and Literature Review

Motion Control and Serial Link Manipulators

Motion Control

Motion control refers to the process and equipment utilized to move a mechanism or actuator in a controlled fashion. Typically, this takes the form of a system with the following components (1):

1. Controller

2. Driver

3. Actuator

4. Sensor/Feedback

5. Gearing/Mechanisms

The motion controller consists of the software and hardware that take in a command for a desired state and any feedback and then send the appropriate signals to the motor driver or amplifier. For a single degree of freedom, this typically consists of running a control loop for velocity and/or position and the appropriate interfaces for the motor being used. When controlling multiple degrees of freedom together, an additional layer that converts input commands, usually in a Cartesian frame, is used to generate the appropriate joint space commands. This conversion can either be a

simple translation to create an open loop upper layer or take in different types of feedback to operate in a closed loop mode.

The motor driver or amplifier is the circuit that converts the low voltage command signal from the motion controller to the high voltage, high current power signal that actually flows into the motor. In a brushless DC motor, this takes the form of the drive transistors for each phase and the commutation logic, but a brushed motor can use a simple H-bridge.

The actuator refers to the component responsible for the motion of the mechanism. In the case of the robots being used at the Space Systems Lab, this is a rotary brushed or brushless DC motor.

Feedback is any system used to determine the current state of the actuator and usually is limited to information used in one of the motion controllers. Using this definition, temperature would not be considered a feedback signal unless it was used as a parameter in the motion controller (including a torque loop) for an actuator that varied performance with temperature. In the RANGER and SAMAURAI robots, feedback for the brushless DC motors includes digital Hall Effect sensors and incremental encoders, with the RTSX arms also using absolute encoders (placed after the harmonic drive).

The gearing and mechanisms translate the motion of the actuator into the desired work output. The gearing and robot links fall into this category. On all of the SSL robots, this takes the form of harmonic drives. A harmonic drive is composed of a circular spline, a flexspline, and a wave generator(2) as shown in Figure 1. The wave generator is the input of the gearing and the output is connected to the flexspline. While the gearing varies, common gear ratios used range from 60:1 to 200:1.
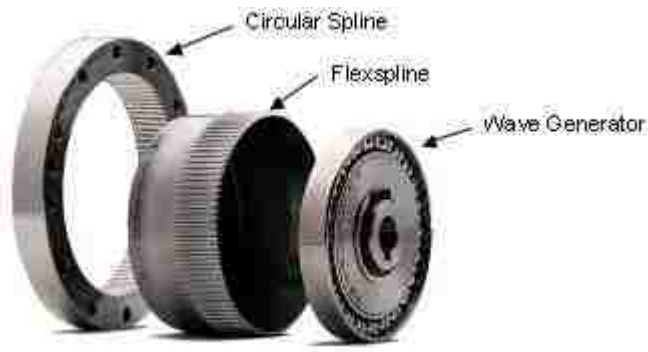
Figure 1 - Harmonic Drive Components (2)

Serial Link Manipulators

A serial link manipulator is composed of individual joints, either revolute or prismatic, that are connected in series by rigid links. The most familiar example of this is the human arm that consists of rotational joints separated by rigid bones or links. In addition to their large workspace for a minimal footprint, it is this similarity to the human physiology that makes serial manipulators ideal for many robotics uses as they are suitable for tasks originally intended to be completed by humans.

Motion Control Topologies

The interactions of the controller(s), driver, actuator and the feedback of a motion control system and how they are connected are based on the physical requirements of a robot and the available technology. Every robot has unique requirements and there is no strict definition of centralized and distributed control, resulting in a large number of unique solutions for different robotic systems.

Centralized Control

Centralized control refers to a configuration where a single monolithic processor controls the motion of a robotic system. This controller receives the feedback signals from encoders, feeds them into the motion controller, and sends the output to the motor drivers.

The main advantage of such a system is simplicity as it uses the absolute minimum number of components. This may be slightly tempered by the requirement that all feedback sources and actuators must be wired directly to the main controller. This is often acceptable in systems that inherently lean towards a star configuration, where nodes have a radial distribution from a central controller rather than a chain. Serial link manipulators unfortunately may build up a very significant trunk by the time the wiring for 6 or more degrees of freedom are combined. In the case of RANGER, each motor has 3 heavy gauge wires for the motor phase, 5 wires for the Hall Effect

sensors, and 5 wires for the encoders. This results in a total of 18 power wires and 60 signal wires comprising the trunk as it enters the robot. Moreover, the direct point to point connections preclude most reconfigurable designs.

Distributed Control

Distributed control is characterized by splitting the controller into discrete high level control and multiple joint level controllers.

The high level controller is responsible for all multidimensional or multi-DOF motion. In the robots described here, this is usually in Cartesian space.

The control of individual joints moves from the main controller to additional processors that may be responsible for one or more joints. The SSL has named these distributed computer processors Local Processing Units or LPUs. From this point, LPU will refer to the hardware to differentiate it from the control loops that run on it.

While splitting the controller has implications for the controls and software, it is done to simplify the hardware of the system. By moving control of a joint to a collocated LPU, the wiring with motion commands to the drivers and feedback to the sensors are significantly shorter and do not have to be routed back to the main controller outside the robot. This is only an advantage however, when it is significantly easier to route the network signal than it is the driver and feedback signals.

From a practical perspective, it was not worthwhile to move to a distributed control configuration until microcontrollers (with sufficient processing power) became small enough to fit into the structure of a given manipulator. The Rapidly Deployable Manipulator System, or RDMS, was developed by Carnegie Mellon in the mid-1990s and is a prime example of the early distributed motion control systems. This design featured 1 degree of freedom modules that were each controlled by a processor located inside the joint. A prototype of the joint is shown in Figure 2.

Figure 2 - Rapidly Deployable Manipulator System (RDMS) Joint

The design of the RDMS is shown in Figure 3. The physical wiring is for power and a differential serial bus, using the RS-485 standard, a half duplex, multimode bus. The microcontroller communicates with a master controller over this bus and then uses a parallel bus to communicate with the analog motor and sensor interfaces that were external devices and not integrated in the microcontroller. One of the common features of designs from this era is the use of an analog servo amplifier, either an

embeddable module or more commonly, a repackaged standalone driver. While this

was the only practical design option at the time, the analog connection can hold back

the performance of digital drivers.
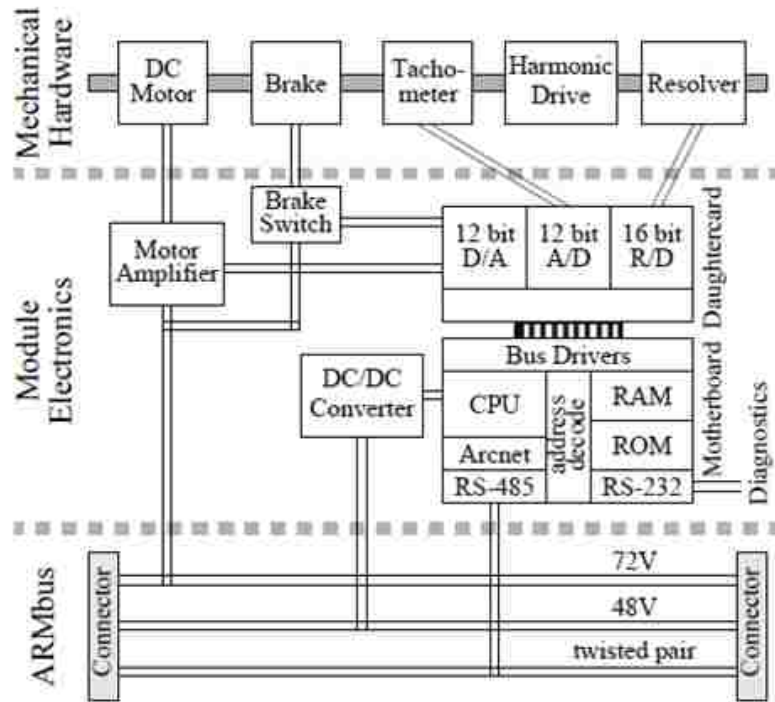
Figure 3 - RDMS Design and Interfaces

As processors became powerful enough to control multiple motors and more degrees

of freedom were built into a single joint, an additional consideration becomes the

number of distributed nodes and if it is more appropriate to have one per joint or one

per assembly. The primary issues are those related to size, and cost. At one point,

embedded computers of sufficient processing power were too large and costly for use

in each joint. Thanks to miniaturization and the advent of Systems on a Chip or SoC,

effectively a computer on a single die, the processing components are similar in size

to the large transistors used to drive the actuators.

11

Robonaut 2, a humanoid robotic torso shown in Figure 4, designed to operate on the

International Space Station, is an example of modern distributed motion control as

can be achieved today. It features what its designers call "supperdrivers." These

controllers are an evolution of the basic design from the RDMS and similar robots but

features more modern technology including the use of Field Programmable Gate

Arrays or FPGAs which allow the designer to customize the processor interfaces for

the application. Each controller drives 3 motors with absolute and incremental

position feedback and communicates with a central controller using a custom

Multipoint Low Voltage Differential Signaling Bus (MLVDS). (3)

Figure 4 - Robonaut 2 (3)

The RANGER Architecture

RANGER is a dexterous robotics system that has been in continuous development at

the University of Maryland Space Systems Lab since the early 1990s. The family

encompasses several generations of the system that aimed at neutral buoyancy simulation of space robotics, flight qualified robots, and derivative designs for use in applications such as deep sea exploration.

NBV-I

The RANGER program started with the design of RANGER Neutral Buoyancy Vehicle or NBV (later NBV-I), designed to operate in a neutral buoyancy environment and serve as a testbed for spacecraft servicing. The system consisted of a free flying vehicle with two 7-DOF dexterous arms, one grapple arm to attach to the worksite, and a 6-DOF manipulator to position a set of stereo cameras. The vehicle (sans camera arm) is shown in Figure 5.
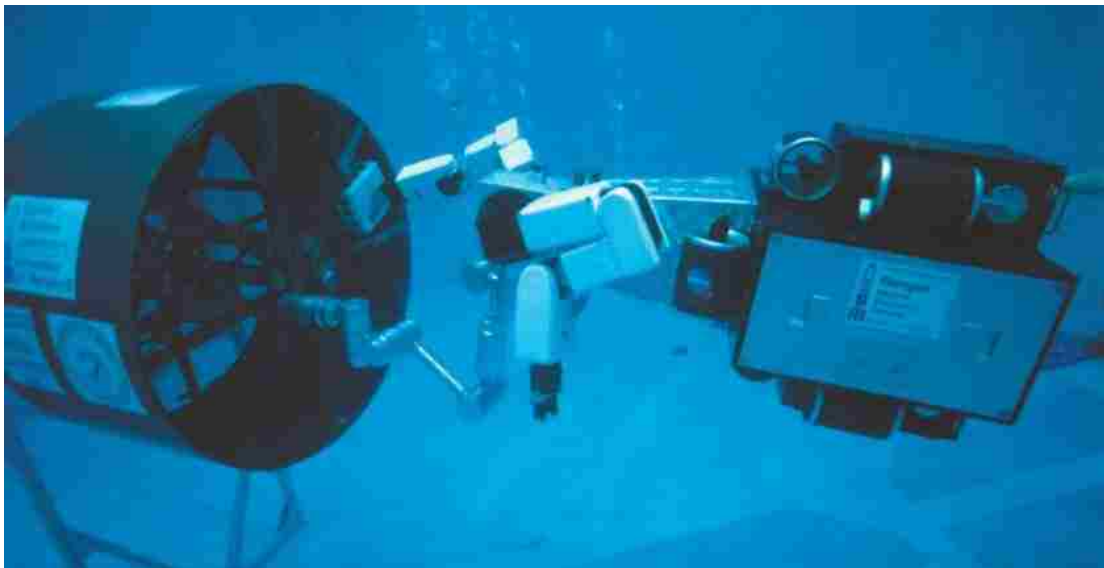
Figure 5 – RANGER NBV-I in Testing with Dexterous and Grapple Arms

RTSX/NBV-II

The RANGER Telerobotic Shuttle eXperiment (RTSX) was a follow-on to the NBV-I generation to design, build, and fly a space rated version of RANGER to test

advanced robotic servicing capabilities on a Space Shuttle flight. In addition to the flight articles, a nearly identical prototype called NBV-II was built to test the systems capabilities in earth gravity and neutral buoyancy. The system moved from a free flying vehicle to a 6 DOF positioning leg that would be mounted onto a SpaceLab pallet in the shuttle payload bay. Additionally, the dexterous arms gained a significant amount of functionality due to an advanced wrist with three degrees of freedom and two tool drives and the ability to use interchangeable end effectors. While the system never reached orbit due to the effects of the Columbia accident, the NBV-II manipulators have accumulated thousands of hours of operation proving not only their function, but also enabling research into reconfigurable robots, spacecraft servicing, and autonomous operation.

SAMURAI

SAMURAI is a derivative of the RANGER system developed for deep sea use. It is a modular 6 degree of freedom manipulator that is designed to operate 5-6 kilometers below the ocean surface without human intervention. It keeps the same basic modular design based on brushless motors and harmonic drives, but features a simpler, stronger structure to take the immense pressures at depth, an oil compensation system, and external wiring. It is also the first SSL robot developed to operate autonomously.

Figure 6 - SAMURAI Manipulator with End Effecter

Previous RANGER Electronics

NBV-I

The electronics for NBV-I consisted of a controller board for each joint assembly (1-4 degrees of freedom) and a separate motor driver board for each actuator. The sections of the arm between joints consisted of a skeletonized tube that had a guide to mount the controller board down the center. A serial bus provided the communications through the manipulator.

The controller card consisted of a motherboard that hosted a smaller commercial 68HC11 microcontroller board. The 68HC11 is an 8 MHz, 8 bit processor with basic support for serial and parallel buses. The larger motherboard provided the connections for the feedback and the motor drivers by using a parallel bus connected

to commercial integrated circuits for digital to analog conversion to generate current commands, analog to digital for analog conversionfeedback, and buffered inputs for the encoders, both incremental and absolute. Each motherboard could support up to 4 actuators, but all varieties used in the NBV-I camera arm only had the components for 2 drivers populated.

The main communication bus from the high level controller to the nodes was originally a multidrop serial bus. A custom communications protocol called MIMICS ran on top of this and handled the addressing and commanding. This setup proved problematic as the serial bus was relatively slow and could not handle the commands and the overhead for three controllers. In a later research project, the limited capacity was rectified by replacing the single serial bus with a connection for each of the three controllers. This resulted in serious modifications to the wiring, including additional serial connections both inside and outside the arm.

RTSX

The RTSX/NBV-II electronics are an evolved version of those used in the NBV-I vehicle, but with a vastly higher level of refinement congruent with being designed as part of a flight program. The RTSX manipulator was not used for this thesis directly. However, at the inception of this research, it was the only dexterous robot with a functional robot controller. In addition to being the metric by which the new electronics are measured, it also provided a starting point for the design and the minimum requirements.

The main difference apart from evolution of all the systems was the switch to a
backplane configuration. A backplane of the appropriate size hosts a processor, a
power converter, a communications board, and from one to five motor driver boards
as required. Figure 7 shows this setup with the power, processor, and
communications boards being populated. The backplane mounts into an aluminum
housing either built into the arm or in the case of the shoulder, a protruding "tumor."
Each of the cards is protected against vibration and other loads through the use of
wedgelock connectors that lock the cards in place within the aluminum chassis.
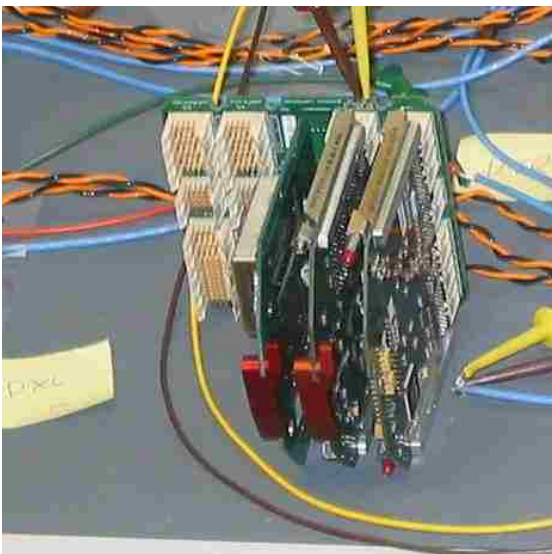
Figure 7 – RANGER RTSX/NBV-II Backplane (Prior to Installation in Arm)

The communications was greatly improved by the use of a MIL-STD-1553 serial bus.
This bus is fully redundant and operates at 1 Mbps in half duplex mode. The
additional bus throughput combined with the replacement of the MIMICS
communications with a more efficient one, solved many of the issues that had been

experienced in the earlier NBV-I design. For more information on MIL-STD-1553, please see the communications section in Chapter 3.

The commercial 68HC11 board was effectively replaced by a new custom processor board hosting a radiation tolerant 30 MHz embedded Intel 386SX processor. The new processor interfaces with the rest of the boards through the backplane. In order to save overhead, the processor has no operating system, but rather the embedded joint space control program is run on its own. Also, owing to some of the flight requirements, the software is identical for all LPUs, but the configuration values are stored on an externally programmed EEPROM.

The motor driver modules, shown in Figure 8, utilize the same motor drivers used in the NBV-I arms, the PWR-82520. Not shown, but critical are low profile heatsinks that are attached on top of the motor driver package. In order to get the drivers thin enough for the backplane, a significant number of taltum and ceramic capacitors replace the larger electrolytic ones used on NBV-I.

Figure 8 - RTSX/NBV-II Motor Driver Boards

FireWire Experiments

At the inception of the SAMURAI project, an effort was started to develop a completely new set of electronics for the deep sea arm with significant margin on processing power along with communications speed and throughput to enable future advanced robotics work. The design team carried over the same design philosophy as used in the RTSX electronics, but with completely new hardware. This new design needed to be much smaller in order to fit in a small pressurized housing protruding from each joint. Additionally, the move away from the space environment meant that the designers were free to use more sophisticated commercial technology such as ARM processors and FireWire communications that were not accessible for a space rated version.

The PW-82520 motor drivers were replaced by slightly smaller, but otherwise similar M.S. Kennedy modules. Both of these units contain hardwired torque control logic driven by an analog input. These modules provide a simple way to implement a robust, radiation tolerant motor driver, but do lack the abilities of more modern systems where the commutation can be controlled directly by the microcontroller and thus have more control and support for operations such as microstepping. There is also the consideration that the commanding of these motor drivers is analog, increasing the complexity of the circuit and presenting potential noise and performance challenges.

The highlight of this new design was the ability to communicate using a daisy chained IEEE 1394 FireWire bus. FireWire has the advantage of being affordable (compared to MIL-STD-1553), fast (at 400 Mbps), and real time. While all the components exist, this system needs to be built from individual components and integrated circuits. One major consideration with FireWire is that the high speeds require tight tolerances for circuit impedance and cabling.

Unfortunately, while the complexity of modern processors, FPGAs, and the FireWire communications provide the potential for a powerful motor driver, the complexity of the system means that the cost may be too high. This research was initiated after the first version of the 1394 electronics became defunct, and the second attempt unable to meet schedules. The question was not if the FireWire electronics could provide

sufficient performance but if there was a different and more affordable solution that

could do so sooner.

Chapter 3: Architecture Development

Communications

Interestingly, while the design of a distributed robotic control system is dependent on the processor, motor driver and communications, it is the communication that dominates the design. This does not mean that the processor and motor driver are less critical, but they simply need to be executed properly. The processor requirements are minimal as NBV-II has shown that a 30 MHz 386 is sufficient to operate five drivers. Similarly, the motor driver side is almost generic and the design process is limited to choosing a driver that is compatible with the proper motors and feedback (Hall Effect sensors for commutation). From this perspective, it is the communications architecture, including the bus and the data protocol that defines the entire motion control system for the robot.

This section takes a look at the major options available for the communications bus. While the list is not exhaustive, it includes those standards that could reasonably be used for communication with the LPUs based on the following requirements:

1. Bus must be capable of ~1 Mbps, comparable to the throughput on RANGER NBV-II

2. Drivers and support hardware should be commercially available

3. Must be a multidrop, either a single master with multiple nodes or a multipeer configuration

Fully custom buses and those that would require a Field Programmable Gate Array (FPGA) were avoided in order to limit the development time required.

MIL-STD-1553

The current NBV-II robots utilize MIL-STD-1553 as their primary data bus, in large part because of the space rating that was required for the electronics to be used as part of the RTSX shuttle flight. The standard dates back to 1973 when it was developed as a serial bus for avionics in the F-16 fighter jet. Since that time, it has become a standard data bus in dozens of aircraft, weapons systems, and satellites. The bus has a maximum speed of 1 Mbps and operates with a master unit and a number of slave devices or nodes, allowing the bus to operate in a completely deterministic manner.

There is no set protocol that defines what data is passed between the controller and nodes except for a 5 bit address (supporting 32 nodes) and a basic transmission protocol. Any commands or messaging past the addressing is left to the user to develop. The current DMU and LPUs used in NBV-II already have a set of commands that could be used to make any new system compatible.

The downsides to 1553 include the cost, wiring, and the lack of commercial drivers. From the start, the cost of such a system precluded it from being a realistic contender. A PCI interface card for the DMU would cost between $5,000 and $15,000 with the node costs being similarly high. While this was an acceptable price tag for a multimillion dollar flight program, it does not fit into the low cost manipulators that are currently needed for research. The other challenge with MIL-STD-1553 is the

standard wiring, known as "twinax" or twinaxial cable, shown in Figure 9, which resembles coaxial cable but with a twisted pair at its center. This cabling adds to the cost and complexity and is incompatible with the wiring in the NBV-I arm as well as the underwater cables used on SAMURAI.
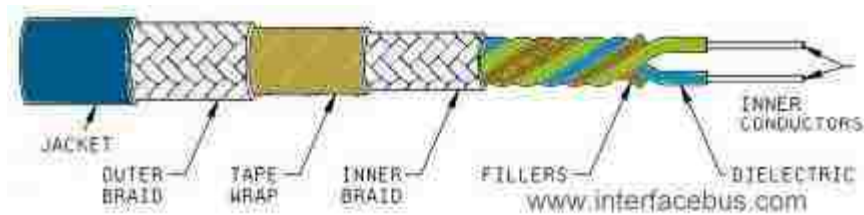
Figure 9 – Twinaxial Cable

RS-485 and Custom Serial Solutions

RS-485 is a high speed half or full duplex multidrop serial bus. It features differential signaling and can operate at moderately high speeds (up to 35 Mbps) over standard Unshielded Twisted Pair (UTP) wiring. Either two or four wires are used for half-duplex and full-duplex communications respectively. The number of nodes depends on the line drivers used, but is at least several dozen. The biggest advantage of the system is the very low cost. A simple commodity microcontroller with a UART and a line driver is all that is needed for a node, with a price tag of a couple dollars.

The main disadvantage of using RS-485 (or similar protocols such as RS-232 or RS-422) is that they are even less defined than MIL-STD-1553. It would require a completely custom communications protocol. The NBV-I controllers actually used a similar communications system called MIMICS. In the end, the custom communications were holding back the entire robot, with the only solution being to replace MIMICS with direct serial port dumping raw data to each controller.

Some commercial motor controllers using RS-485 exist but they are generally at the

hobbyist level and do not follow any major standard. Additionally, most of the

manufacturers are small outfits that only offer one type of controller. Between the

limited selection, quality, and standardization, none of these products would meet the

needs of this research.

Ethernet

Ethernet is a networking standard that forms the backbone of most computer

networks. It was originally developed by Xerox in 1980 for networking desktops and

servers, but has moved into small microcontrollers in the last decade and is thus

worth considering here. Ethernet originally shared a common medium, operating all

the nodes on a single set of wires, but has moved to a tree topology with switches and

hubs since the late 80s. The standard is designed for peer to peer communications

and thus is far less controlled than many of the other buses being considered. This

combined with its physical layer, means that the bus is susceptible to collisions where

multiple nodes attempt to communicate at the same time. The solution is that each

node can detect a collision and then wait for its turn to retransmit the packet, making

the network non-deterministic.

The biggest advantage of Ethernet is the inclusion of it in nearly all computers and

the vast support in every modern operating system, especially when using the TCP/IP

and UDP protocols. TCP or Transmission Control Protocol creates a reliable

communications path where any packet that is not copied correctly is resent until it is

received. IP, or Internet Protocol, refers to the addressing system that provides a unique address to every node on a network and manages the routing of packets. UDP or User Datagram Protocol is a simpler non-reliable transmission method that is faster than TCP. UDP provides the best option for use in robotics as it can, with proper coding and control of the network, provide something close to deterministic operation, making up for the lack of true real time operation with its much faster speeds.

The main issue preventing the use of Ethernet in a serial link manipulator is that modern Ethernet only operates in a star topology. This means that every node, or at least every compartment containing nodes, in the robot must also include an Ethernet hub or switch. Hubs, and the more advanced switches are both complex and do not lend themselves to extreme miniaturization. No commercial hubs or switches that could fit with a controller in a SAMURAI pressure housing were ever identified. Building a custom switch was considered, but the complexity of a miniature hub would add a significant amount of development and debugging time.

In addition to traditional Ethernet, a number of standards attempting to adapt Ethernet into an industrial field bus were gaining popularity at the start of this research. The main ones were EtherCAT, Ethernet/IP, and SERCOS III. At the time, these standards were not considered, as the complexity (often requiring custom FPGAs or ASICs) was too high for a custom solution and the commercial options were too large for use inside SAMURAI if not RANGER. At the time of publication, commercial

modules are becoming available that meet the requirements of these types of robots

and one option is further explored in the future work chapter.

FireWire

FireWire, or more properly, IEEE 1394, is a high speed serial bus standardized in

1995 with the intention of replacing the parallel SCSI bus used for disk drives as well

as provide an audio/video interface. The original standard specifies a half duplex

serial bus that operates at 100, 200 and 400 Mbps although newer revisions to the

specification have raised this to 800 Mbps and higher.

While FireWire first appeared as a consumer product, it has begun to replace serial

buses such as MIL-STD-1553 and CAN in places like aircraft avionics and

automobiles and spacecraft. The bus operates with a single master and is

deterministic. Typical applications include those that require more bandwidth than

the earlier buses can provide for high throughput applications such as radar and video.

The overhead, complexity, and cost prevent it from replacing the legacy buses

wholesale.

In addition to the speed, which provides room for growth and more room for

debugging, FireWire has a couple of advantages that stem from its daisy chain

topography. By operating with the nodes in a daisy chain, the order is easily

identified, which opens up options for reconfigurable robots that identify their

configuration automatically. Another possibility is adding FireWire cameras directly

into the main network and avoiding the complication of analog wiring.

One non-trivial issue with FireWire is the tight standards for the system wiring. The standard operates over 110Ω twisted pair, but the high speed, 400 Mbps standard, means that it is far less tolerant than many other buses including Ethernet, but especially CAN and RS-485. One issue with the SAMURAI manipulator is that the SSL was never able to procure depth rated cables that were capable of supporting a 1394 bus although 10 and 100 Mbps Ethernet functioned and CAN2.0B had no issues at all. Even when the proper cabling is available, it is challenging to use in robots as all the boards it operates on need to be impedance matched during design and manufacture, a process both costly and time consuming. The other issue that occurs is that cables change in impedance when they are deformed. While this is mitigated by modern bonded construction in high quality cables, it is something that should be tested to ensure that the cables will not fail with continuous bending as seen in the SSL serial link manipulators. The latest FireWire LPU mitigates a number of the wiring issues with 1394 by adding transformers and a dedicated equalizer IC to allow the use of lower quality CAT-5e cables, but this setup adds its own level of complexity.

The last major consideration with FireWire is that most products utilizing it are mass produced (such as hard drives or cameras) and the rest, including military components, are very high value. Because of this, most of the chipsets are not aimed at small developers. When products can be acquired in reasonable quantities, they are

often protected by nondisclosure agreements that can cause issues in research labs

that have a transient student workforce.

CAN

The Controller Area Network or CAN bus was originally designed as a vehicular bus

by Bosch in 1986 and became popular in both vehicles and industrial automation in

the 1990s. It is an industrial fieldbus that operates at speeds from 10 kbps to 1 Mbps

over a balanced twisted pair network.

The physical wiring of the bus consists of a simple multidrop network where each

node is connected to the same physical line.   This has the advantage over a daisy

chain topology in that a failed node does not affect the rest of the network. The

wiring consists of two wires, CAN-High and CAN-Low, that form a differential pair

and a ground connection. On the far extremities of the bus, the high and low signals

are tied together through a 120Ω resistor. The bus speed and length are inversely

proportional but the maximum speed 1 Mbps data bus rate is theoretically good for

lengths up to 30 meters, more than enough for any of the SSL robotic applications.

The CAN bus is not completely deterministic as any node is allowed to initiate

communications and the bus supports a hardware arbitration Carrier Sense Multiple

Access/Collision Avoidance or CSMA/CA(3). CSMA/CA implementation is through

the use of dominant (logic 0) and recessive (logic 1) bits. When two nodes attempt to

transmit simultaneously they both start their transmissions but the first one that

attempts to transmit a recessive bit when the other transmits a dominant bit will stop

when the bus does not respond as expected and retransmit its entire message after the

other node finishes. While this does prevent the bus from being deterministic, it has a

minimal impact on the system as high priority transmissions are specially formatted

in order to ensure they win out in any arbitration, ensuring the critical safety and fault

messages make it through on the first attempt.

While the CAN bus is an attractive option for communications on its own, the

CANopen standard which is implemented on top of the standard bus makes it ideally

suited for the motion control application. CANopen adds a specific addressing

system, a format for transmission and what are called Process Data Objects or PDOs.

The PDOs are defined by the CAN in Automation (CiA) standards for various device

profiles that range from motion controllers to GPIO and stand alone encoders. These

standards define the commands for all the basic commands that a given device should

respond to. Effectively, this allows the design team to move away from the custom

communications interface and commands used by the existing RANGER robots to an

open and supported standard.

Bus Selection

A summary of the communications options is shown in Table 1. Looking at the

complexity to implement, MIL-STD-1553 and FireWire were not chosen. Cost was

also a non-starter for the MIL-STD-1553 option. Ethernet was disqualified as it could

not be implemented inside the volume constraints of the SAMURAI manipulator.

The best options for a simple, low cost, but capable communications buses were CAN

and RS-485, with the former being preferred for use with COTS hardware and the latter being a valid option for an all custom solution.

Table 1 - Communication Bus Options

	Speed	Established Protocol	Wiring	Deterministic/ Timing
CAN2.0B	0.01-1.0 Mbps	Node	2 Conductor 120Ω UTP	Timing
MIL-STD-1553	1.0 Mbps	Node	2 Conductor 78Ω Twinaxe	Yes
RS-485	0.1-35Mbps	None/Custom	2/4 Conductor 120Ω UTP	Not Standard
Ethernet	10-1000 Mbps	Address	2-8 Conductor UTP	No
IEEE 1394	100-400 Mbps	Address	6/8 Conductor Multiple Shields	Yes

Custom vs. COTS

Custom LPU

The obvious solution to the motion controller problem was an upgraded, but functionally similar design to the existing RANGER Local Processing Units. Similar performance could be realized by keeping the same motor drivers, the DDC PW-82520 or MSK 4364, and upgrading the LPU CPU to a modern microprocessor.

An initial design for this option was implemented, based around a ColdFire 68k microcontroller that could interface with a servo amplifier or the PW-82520 motor drivers used in NBV-I. The microprocessor ran the uC/OS real time operating system (RTOS) and utilized the Queued Serial Peripheral Interface (QSPI) of the ColdFire, a 16 Mbps peripheral bus to communicate with ASICs for input and output. The inputs

included analog readings for current and temperature and a digital quadrature counter, along with an analog output that could control the servo amplifier or motor driver module. The hardware was tested using a system called TSUNAMI, a brushed motor force feedback test platform.

Challenges with Custom Hardware

The main issue with a custom LPU is that it does not exist in a vacuum. The DMU code, communications bus and protocol, the LPU electronics, and the LPU embedded code are all highly interdependent. Moving to a new LPU would require either porting the original code, itself a difficult task when the original used a custom toolchain and operated without an operating system, or rewriting the code from scratch. Additionally, the communications software, both on the LPU and DMU would require revisions to deal with the new communications; even if using the same commands, the differences in the communications buses would likely require significant rewrites.

This stage coincided with the loss of the Space Systems Lab's last staff member from the original RTSX team. This highlighted two significant realities that determined the direction of this research, namely loss of experience both developing and operating the legacy RTSX system and that it is infeasible to run future programs in the same form as RTSX without the latter's 17 million dollar budget. With the loss of the professional staff at the laboratory, the cost, especially in time and risk to modify the existing software became significantly higher and effectively stalled the entire

project. It became obvious that a new direction was needed at least for this research, if not for the lab's robotic architecture.

Move to COTS

In the RANGER program, the design of the DMU, the communications, the LPUs, and the LPU software was almost entirely custom, based on the stringent requirements of the flight program. That custom design translates into custom support and maintenance that must be provided by a lab now primarily composed of graduate students. An alternative option is a full or partial transition to a Commercial Off The Shelf (COTS) solution. COTS itself is possibly too narrow a classifier. Many of the options for DMU code are actually Open Source rather than commercial, and the communications is maintained as an international standard rather than a corporate project.

Module Selection

Types of Modules

There are hundreds of CANopen controllers manufactured by a large number of companies. In general, these controllers fall into one of three categories: integrated with a motor, packaged units, or PCB mountable miniature units.

A number of manufacturers currently sell motor and CANopen driver combinations. These units allow for the simplest and quickest integrations as they are already wired, packaged, and tuned for the motor they ship with. These drives also often include encoders or other feedback built right in. As all current SSL manipulators are based

on frameless servomotors, this type of drive is wholly inappropriate. It should be

noted that these drives may have potential for future projects such as rovers or a

positioning leg where volume is less of an issue and could be useful to meet very

aggressive timetables.

The second type of CANopen driver is a packaged module. These are by far the most

numerous variety as they fill the market that was once dominated by similarly

packaged analog servo amplifiers.   They consist of a moderately sized driver board

that is packaged into a metal case, usually using one of the sides as a heat sink. They

have mounting holes that make them easy to attach to large machines or fit into rack

units. Usually the electrical connections are a d-subminiature or 0.1" header that can

easily be crimped and installed in an industrial environment or even more accessible

screw terminals.

It should be noted that the original NBV-I dexterous manipulators originally used

servo amplifiers in this form factor. Some of the units were simply mounted in the

robot where space was available, while others were removed from their packaging to

save space. This would have been an option for this project if embeddable modules

were not available and is still one of the few options for some of the more

sophisticated Ethernet based solutions.

The third and final category of CANopen drivers is those that are marketed as

embeddable modules. These units are designed to either plug into a socket or be

directly soldered into the end user's printed circuit board. The embedded modules are generally the smallest end of a manufacturer's product line if they are offered, but are also the least common CANopen drivers on the market.

Each module requires a motherboard that it can be plugged into. This motherboard can be complicated and include all the non-motion parts of the design or can be a simple breakout to the user's connection. Depending on the module, services such as power conversion and physical layer data transceivers are often left to the OEM to provide. This limits what must be included in the package to reduce size as all functions will not be required for all applications, but also leaves more work to the integrator. Another common feature of embedded modules is a more limited breakout for different position sensors as the number of pins is limited.

Supplier Selection

A review of the available modules was done at the beginning of the project in order to select the most appropriate one for use in the RANGER NBV-I arm. The selection of embeddable modules is increasing as it becomes more feasible to miniaturize drives to this level. It should be noted that the selection was made using the best available material at the time, but the selection will likely change over time. As discussed in the protocol section, CANopen provided the flexibility to easily change manufacturers or even operate a hybrid network. Future drivers should be selected based on the individual requirements with experience using the present units being only one of many factors.

Due to CAN in Automation standards defining the basis of a CANopen motion controller in DSP402, the units available on the market are very similar as far as basic functionality. The major differences are form factor and interfaces, both human and machine. Table 2 shows the three major options considered from Elmo Motion Control, Advanced Motion Controls, and Copley Controls. Each of these manufacturers has a significant market share and provides quality products.

Table 2 – CANopen Motor Driver Selection

Motor Driver	Continuous Current (Max)	Dimensions (mm)	Automatic Tuning	Notes
Elmo Whistle	20 A	55 x 46.5 x 15	Current, Velocity, Position	2mm Pinouts Custom Scripting
AMC DZCANTE-020L080	12 A	63.5 x 50.8 x 22.9	None Advertised	0.1" Pinouts Requires External CAN Transceiver
Copley ACK-055-10	5 A	64 x 41 x 21	Current	0.1" Pinouts

The maximum continuous current shown is from the highest performing units in each manufacturer's product line. All the units also have a peak output current that is roughly double the continuous current and can be driven for a maximum of 1-10 seconds depending on the unit. The units all benefit from modern high efficiency switching drivers that only produce a fraction of the waste heat compared to previous generations and can therefore source more current from a compact package. The Copley Acellnet solution does not include a heat sink at all. The Advanced Motion Controls unit is similar in design, but can drive higher currents thanks to a built in heat sink and the Elmo driver benefits similarly as it is completely enclosed with its

own heatsink. All units are designed to be cooled solely through convection, but the

AMC and Elmo heatsinks can be attached to the robot chassis for better heat flow.

All the units are compact, but the Elmo drive actually fares the best despite being

fully enclosed. All of the units are small enough to be used in the RANGER NBV-I

arm, but some consideration was given to future applications including the

MORPHBOTS and SAMURAI arms that are more restricted in electronics volume.

Possibly more important than a couple millimeters in dimensions is the

interchangeability of the modules from the same manufacturer. The AMC drives all

have different pinouts, so even a small step between drivers would mandate a whole

new board layout. The Copley and Elmo drivers are pin-compatible over a range of

voltage and current ratings. Elmo even offers a version of the Whistle without the

enclosure or heatsink in its Tweeter line that shares the same pinout but has been

derated to 3 Amps continuous.

The configuration software for the drivers often contains some support for tuning the

gains on the drivers. This can be either a manual user interface or an automated

process, but is usually included as industrial drives are often installed and tuned on

the application machinery. Tuning the current loop is the simplest to automate and all

manufacturers usually do this. Copley advertises this capability and it is likely that

although not stated in the datasheets, AMC supports it as well. Tuning velocity and

position loops is a more sophisticated process, making Elmo stand out as its software

supports both manual and automated tuning.

The other differences that played into the decision making process include that the Advanced Motion Controls unit required external transceivers for both RS-232 serial and CAN. This is not a major issue, but increases the complexity and layout of the host motherboard. The Copley and AMC offering supported standard 0.1" pins vs. the 2mm pitch pins on the Elmo driver. While this was not originally given much weight, it is much easier to procure the appropriate imperial headers than the less common metric ones.

The final advantage of the Elmo drives is that they support onboard scripting and programming. While this is not currently being used, it offers the capability to add features to the drive that are not included, the idea being that the more unique behaviors of the RANGER LPUs could be duplicated.

Interestingly, cost was not a major factor in the decision. The pricing amongst the units was very similar with much greater variations being due to the high power transistors being used as drivers and thus increasing the price of the higher rated products. There is limited distribution across the board and although the manufacturers have good sales support, lead times are going to be higher than consumer equipment or raw components and are often in the neighborhood of 8 weeks.

The final selection was made in favor of the Elmo Whistle drivers. They provided the best combination of performance and form factor while also having the onboard programming and the automatic tuning features. The drive that was eventually settled on was their standard unit rated to 10 Amps continuous at 60 Volts although a couple different models have been used interchangeably.

Elmo SimplIQ Drivers

Future Expandability

As stated previously, by settling on an open standard such as CANOpen, there is always the option to mix modules from different manufacturers for future applications. That said, by sticking with the same product family, there is a much shallower learning curve associated with implementing new hardware.

Full Implementation of DS402

The potential flexibility of a CANopen network can be utilized through the higher level software that implements the relevant portions of DS402: *CANopen Profile for Drivers and Motion Control.* The DS 402 standard defines the communications and operation of all CANopen motion controllers. While manufacturers are still allowed to add proprietary extensions, all motor drivers from any manufacturer can be operated using the same commands and responses as seen by the DMU.

One of the downsides to using commercial drivers is that one manufacturer may not offer a product that fits the exact needs of a given project. By using a completely

open standard such as CANopen, specifically DS402, it is possible to select from many manufacturers and still have an easy-to-integrate solution. This is especially useful when moving either to a smaller form factor (Proteus/MORPHBOTS) or to a much larger one. It also allows for a mixed solution where drives must meet vastly different requirements in the same project or robot such as in the RAVEN rover, which has a requirement for ~5A Brushless DC drivers in a robotic arm and also 20-40A brushed motor drivers for the drive train. In this case, it is feasible to have the two CANopen drivers from different companies operating on the same bus and even using the same commands.

Currently, the software that is running on the DMU is using a mix of commands including a simplified interpreted mode that is unique to the Elmo motor drivers and the standardized CANopen commands. A completely DS402 implementation could be added as a second profile or command mode that defines how commands are sent to the drivers. The current Elmo drives could be operated under either mode. It is highly recommended that as this architecture is implemented in an operational manner that a deliberate decision is made to transition from the proprietary Elmo interpreted commands to the DS402 standard for future use with other manufacturers and standards compliance.

Chapter 4: System Flexibility

Design for Flexibility

Communications and Wiring

One of the major challenges with adapting a robot to new controllers is the existing

wiring in the manipulators. If at all possible, existing wiring and connectors should

be reused in order to reduce the amount of work and redesign required. The three key

items are the wiring impedance, the connectors and the number of conductors.

Every high speed data bus has a defined range of acceptable characteristic impedance.

This is determined by coaxial cable construction or the twisting of twisted pairs in a

balanced configuration. Typically, impedance variances become more of an issue

with higher speeds and longer bus lengths. The CAN bus operates with a

characteristic impedance of 120Ω. This means that it is compatible with most twisted

pair industrial wiring and that used for differential serial buses such as RS-485. The

other advantage of the CAN bus is that it supports a wide range of standard

transmission speeds from 10 kbps to 1 Mbps. Lower speeds can be used if the wiring

is of the wrong impedance or otherwise less than optimal. In fact, the bus can operate

over a single wire without the other differential pair at speeds up to 125 kbps (4).

Standard Connections

The intention with the NBV-I manipulator was to keep as much of the existing wiring but to remove the controller cards and replace the existing DDC PW-82520 motor drivers and their supporting circuitry with the Whistle modules. This provides both a mechanical attachment point and a conductive thermal path for waste heat from the driver.

Board Design and Implementation

On its own, the Whistle modules are not easy to integrate with the actuators and sensors in an arm due to their raw 2mm pitch headers. The two options for integrating the Whistle modules into the NBV-I vehicle were to use a version of the Whistle presoldered to a breakout board known as the "Whistle Solo" or design a custom carrier board.

The Whistle Solo, shown in Figure 10, is the simplest way to integrate a Whistle into a robot. It breaks out all of the connections to easy to use screw terminals and crimp headers. The advantage to this option is that there is no need for custom circuit board design and soldering. The major downsides to this option are limited flexibility as the design is fixed and that one must use the Elmo supplied connections and pin outs which are incompatible with those in use by the SSL. The mechanics of working inside the skeletonized body of the robot would have made the use of screw terminals for power and motor coils challenging if not impossible when compared to the use of prewired connectors. While the issue had not been identified at the time of

purchasing modules, the encoder issue described later would have required a separate supporting board for each joint to modify the signal.

Figure 10 - Elmo Whistle Solo (Elmo MC)

A custom PCB allows for the existing connections (and positions) to be maintained, limiting the number of changes to the existing wiring. The cost of a custom PCB is primarily development time and manufacturing.

The printed circuit board was designed using a PCB Computer Aided Design (CAD) package called Eagle developed by CadSoft. Using this software, footprints and symbols were created for each component based on the manufacturer's documentation. These footprints were then combined into a complete schematic for the board. The final step was to convert this schematic into a board layout. In addition to placing the components, the conductor traces need to be routed. There are auto-routing features that can connect traces, but the performance is sub-optimal and do not handle the high current traces for the power and motor phases well. Because of this, the traces needed to be routed by hand. One unique feature to the board is that there is no common ground plane. The Whistle includes a built in ground star so

adding a ground plane on the PCB would only create a situation conducive to ground loops.

Manufacturing costs for printed circuit boards can vary widely based on complexity. The goal for the NBV-I manipulator was to design a board that minimized complexity and thus cost, which allowed for numerous design revisions while navigating the learning curve with the modules. In order to do this, designs were limited to 2 layers and square shapes so they could be cut in house using a press break. Additionally, the board layouts were done with wide tolerances in order to decrease the chances of defects and thus make it unnecessary to pay for electrical testing of the final boards. In this manner, a full set of boards for 1 or 2 arms can be manufactured for $100-200 making it a minor cost in terms of the complete system.

The power and ground connectors used on the boards are the same 3 pin Commercial Mate-N-Lok connectors used in the original wiring, one each for power and the motor coils. A 3 pin Molex KK connector, also carried over from the original design, provides the separate control power. Hall Effect sensors are connected using the 1x5 Harwin MTE connector and the similar 2x5 connector is used for the encoder connection. The two 8P8C "RJ-45" connectors on the far side are used for the CAN bus. A small 5cm stub connects to the controller, maximizing the bus performance, and the second connector acts as a pass through to the next node. The only installed wiring that was modified was that used for the networking. The original wiring was quite out of order after being modified in a prior experiment. Between the nodes,

short Cat5e Ethernet patch cables were used and the installed 4 wire industrial UTP

cable that passed through the base and the elbow was reterminated with a 8P8C

modular connector.

Encoders

Figure 11 demonstrates that the two channels quadruple the positioning accuracy

State	Ch A	Ch B
S_1	High	Low
S_2	High	High
S_3	Low	High
S_4	Low	Low
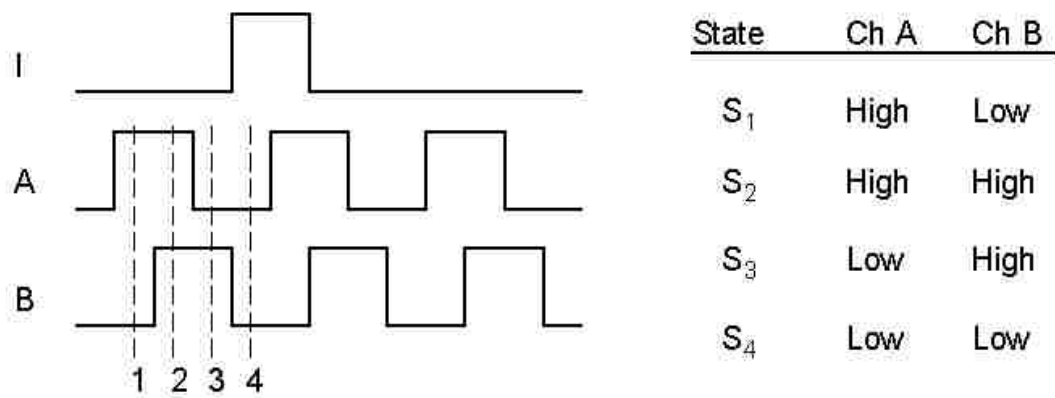
Figure 11 - Quadrature Encoder States (5)

Differential Signaling

In industrial automation, encoders are often separated from the controller by several

meters as compared to the ~10 cm between encoders and the motor drivers in the SSL

robotic arms. This requires a long cable run through an noisy electromagnetic

environment that can cause significant electrical noise. Because of this, encoder

signals are almost exclusively transmitted as a differential or balanced pair.

Differential signaling, as shown in Figure 12, involves sending an inverted duplicate

of the original signal. The two channels are sent as a twisted pair so that any noise

affects both signals and then converted back to the original signal at the receiver. In

the example shown, "A" is the original signal and "A-" is its differential pair.
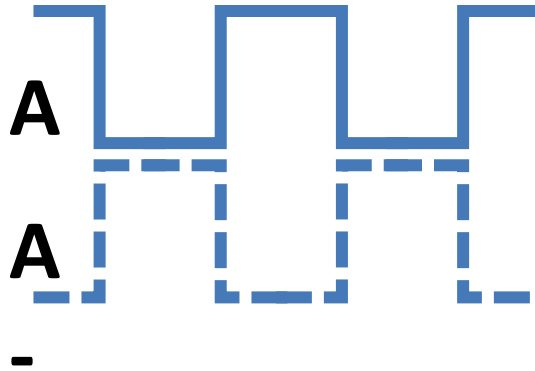

Figure 12 - Differential Signaling

The design philosophy in use at the Space Systems Lab has been to use only single ended encoder signals for very different, but reasonable motives. Firstly, there is little need for differential signaling as the wire runs are usually short, in the neighborhood of 10 cm. Since there is no need for it, differential signaling has been eschewed in order to simplify the electronic and mechanical design requirements. From an electronics perspective, differential signaling requires a subtractor circuit to convert the differential signal back to a single ended one, making it one more component that is needed for the electronics. Possibly even more of a challenge, is that differential signaling requires a twisted pair instead of a single conductor for each signal. These extra wires pose challenges with routing through the moving portions of each joint and, in the case of SAMURAI, double the number of penetrators required.

The two different design methodologies result in the single ended encoders being used by the Space Systems Lab being incompatible with the Whistle drives that have a differential receiver for use with balanced encoder signals. Since the encoders and the wiring could not easily be replaced, the single ended input needed to be converted into a differential one at the LPU board. This could be implemented with logic, but it is not trivial, especially since the timing of the non-inverted and inverted signals must be synchronized. There is actually a readily available solution designed to convert single ended to differential signals, but it has nothing to do with encoders or sensors. A number of serial buses, including RS-422 and RS-485, operate using differential signaling and both drivers and receivers are readily available to interface with

microcontroller UARTs. The AM26C31 from TI, shown in Figure 13, is just one of
many differential driver ICs designed for RS-422. Aside from the timing, a
quadrature pulse is no different than serial data and the AM26C31 can operate at up
to 10 Mbps or in this case 10 million pulses per second (pps), more than enough for
the application. One AM26C31 is used for each encoder and is integrated onto the
LPU motherboard as shown in Figure 14, with one driver being used for the A, B and
index channels.
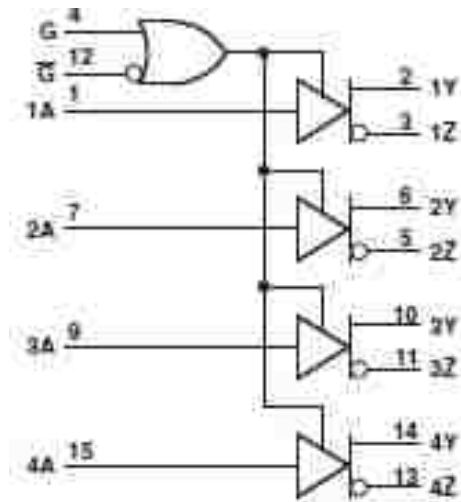
Figure 13 - AM26C31 Logic Diagram(6)
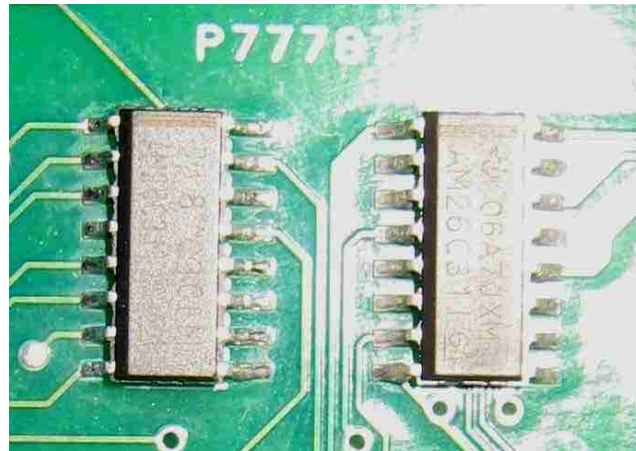
Figure 14 - Differential Drivers on SAMURAI PCB

SAMURAI Implementation

SAMURAI is designed as a deep submergence underwater manipulator. Depending

on the configuration, the arm itself is designed to operate at depths of up to 6000

meters. Much of the basic design is derived from the MORPHBOTS PROTEUS

system, itself an evolution of the mechanical design of RANGER NBV-II and RTSX.

Much like the RANGER NBV-I camera arm, each actuator consists of a brushless DC

servomotor combined with a harmonic drive. Unlike any other actuators currently in

use by the Space Systems Lab, the mechanical portions of the manipulator are oil

compensated to withstand the pressure. The only portion of the arm that is filled with

air is the electronics bubble that contains the motor drivers.

Utilizing the Existing Wiring and Connectors

All of the electrical connections travel through pressure rated penetrator plugs and connect to the bottom board of the electronics stack.

The original design called for communications over IEEE-1394 FireWire. After some initial difficulties, the design switched to a FireWire over twisted pair Ethernet line. The cabling used is a depth rated Cat-5e Ethernet cable, of which two twisted pairs are available for communications while the other lines carry the control power and ground.

Encoders

SAMURAI, as it operated during testing, could use either an optical or magnetic incremental encoder. Both of these encoders do not operate as traditional optical quadrature encoders, but simulate their outputs instead.

The optical encoder used is a Numerik Jena Kit R. This is a high precision encoder, with more counts than the ones used in the NBV-I arm. On top of this, the encoder channels are filtered through a digital signal processor, effectively multiplying the output to the driver. The encoder has 1800 real counts per revolution, but the signal processor has a 5x multiplier. This simulates a 9000 count per revolution encoder, or roughly 9 times the resolution of those in the RANGER NBV-I manipulator.

The SAMURAI arm is oil compensated for deep sea work, being filled with oil instead of air, and thus it cannot use a typical optical encoder to any level of reliability. The main issues are the transparency and viscosity of the oil and

50

especially the particles that will be flowing in it. Instead, the arm will be using a digital magnetic encoder. The unit chosen was a Renishaw RMB20IC. Instead of an optical disk, a small cylindrical neodymium magnet is mounted to the motor shaft and rotates approximately 2mm from a sensor IC. This IC uses an internal array of microelectromechanical or MEMS Hall effect sensors to determine angular position at up to 13bit resolution or approximately 2.6 arc minutes between measurements. This position is then translated into a simulated quadrature position that can be read using existing hardware that is expecting to receive quadrature from an optical encoder.

While the magnetic encoders offer a high degree of precision, this should not be confused with accuracy. The specified accuracy is only +/- 0.5°, or similar performance to a 180 pulse optical encoder. An additional error comes from the alignment of the magnets used for the encoder. A 0.5mm misalignment will result in a 1.25° error. The final area of concern is the time delay from processing the Hall Effect data, resulting in ~1.25° of error per 1000 RPMs of actuator velocity. This last error is significantly mitigated as the system was tuned as a whole and thus the gains already account for the sensor delay.

Board Design

The pressure housing for the electronics was purpose designed for the original FireWire design, featuring the precise volume and penetrator pin placement that was optimal for the original electronics layout. The volumetric requirements were eased

slightly as the second generation of FireWire boards was slightly larger and an

aluminum spacer ring was designed to raise the height of the pressurized area. The

final Whistle design features a horizontal penetrator board designed to fit the original

high pressure penetrator connectors and a vertical riser board to hold the two motor

drivers as shown in Figure 15. Futurebus connectors were used to connect these two

boards as they feature the ability to combine high current (6.5 Amps per contact) and

high density 2mm pitch connectors for signal connections and also provide the

mechanical connection to mount a riser card.

Figure 15 - SAMURAI PCB/Motor Driver Assembly

The board design started with the final operational schematic from the NBV-I

implementation. This schematic was duplicated to create a board with both

controllers with their own connections to motors and sensors, but shared power and communications. The board layout was redesigned to fit the riser card form factor with all the connections being routed through the Futurebus connectors. A minor but significant upgrade first implemented on SAMURAI is high current machine pin connectors for the power and coil connections. The high current pins can be easily differentiated from the standard 2mm headers used for signals on the right and left sides of the board in Figure 16 respectively.

Figure 16 - SAMURAI Vertical Riser Card

The penetrator board was designed by taking the schematic and board files from the second FireWire LPU attempt, which had previously been fit tested with the penetrator assembly, and removing all components besides the penetrators and the board routing. The vertical Futurebus connectors were added onto this design,

positioned down the middle of the board as can be seen in Figure 17, and the connections were routed to the proper penetrator pins shown in Figure 18.
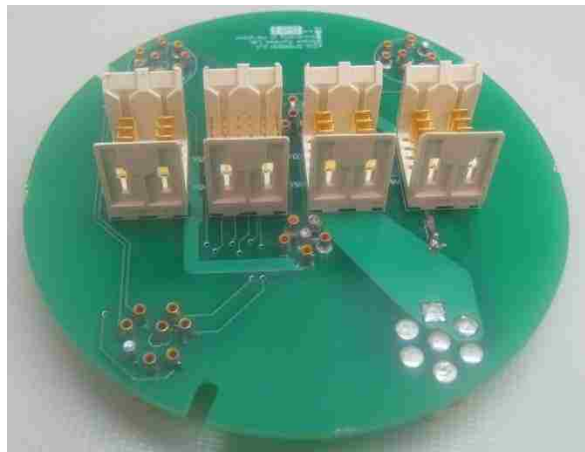
Figure 17 - SAMURAI Penetrator Board with Futurebus Connectors (Top)

Figure 18 - SAMURAI Penetrator Board with Hypertronics Penetrator Pins

Other Robotic Systems

In addition to the two main implementations of the system, the CANopen motion controllers were also used in a more limited manner for two other robotic systems in the lab. In both cases, it took less than two days to achieve single joint motion. While these setups did not include a full installation with the proper electrical and mechanical connections, they lay the groundwork for future expansion of the system and demonstrate that there are no complications from the different hardware.

54

MorphBOTS 2-DOF Actuator

MorphBOTS is a miniaturized version of the RANGER system designed as a follow-on to RTSX to enable low mass (and thus low cost) spacecraft servicing. A two degree of freedom manipulator was designed and built under a DARPA grant, and was originally powered by a set of servo amplifiers. Despite not yet progressing past a single joint, the hardware has been used for a number of research projects.

As part of a demonstration, the MorphBOTS actuator was driven using two of the motor driver boards developed for RANGER NBV-I. The encoders utilized are the same Numerik Jena Kit R units used in the optical encoder variant of SAMURAI with 1800 pulses and a 5x multiplier for an equivalent of 9000 pulses or 36000 counts. At the joint level, full function was achieved although the overall functionality was limited as there is no higher level control available for manipulators with less than 6 degrees of freedom.

The MorphBOTS joint has the potential to be an ideal test bed for any future research working with controls and the motor drivers presented here as it features a nearly pristine actuator and the high resolution (and precision) encoders from Numerik Jenna. While the Whistle module was used for the proof of concept testing, it is not the appropriate device for long term use with the MorphBOTS test rig. Instead, a packaged device such as the Harmonica from Elmo would be much more rugged and resistant to damage. If there was an effort to install electronics in the joint, either the

Whistle or the Tweeter, a pin compatible model that saves space by omitting the heatsink, would be appropriate based on the packaging requirements.

RANGER NBV-II Shoulder

In September of 2010, the DXL manipulator was experiencing some issues related to the shoulder pitch joint. This issue was narrowed down to two possibilities: either the motor was damaged or the electronics (including the backplane) were experiencing issues. It was decided to temporarily install a Whistle module to rule out any issues with the motor itself.

Due to their similar design, the shoulder for the NBV-II arm operates in a similar manner to the joints on the NBV-I camera arm. Instead of a Kollmorgen RBE-01812A motor, the DXL shoulder utilizes a RBE-02112A motor that produces nearly double the torque. The changes required for this motor are limited to changing the allowable speed and currents as it still has 6 pair poles of magnets and thus the same commutation. The most significant delta between the joints in the DXL and camera arms is the use of encoders. For reasons relating to space rating, the NBV-II arms use custom optical incremental encoders. The number of counts in each revolution, 763, was reverse engineered from the existing ranger software.

The wiring of the joints, specifically the lack of connectors and all the motor and sensor wires being soldered directly as shown in Figure 19, meant that all connections needed to tap into the existing backplane. Adapters were created for the motor windings, Hall Effect sensors, and incremental encoders. Also, since the original

design used a power card to power all the sensors, a separate power injector was

created as the single Whistle can only source 200 mA at 5v, plenty for a single

encoder and Hall set, but insufficient for the 6 encoders and 3 sets of Hall sensors

hardwired to the shoulder backplane.
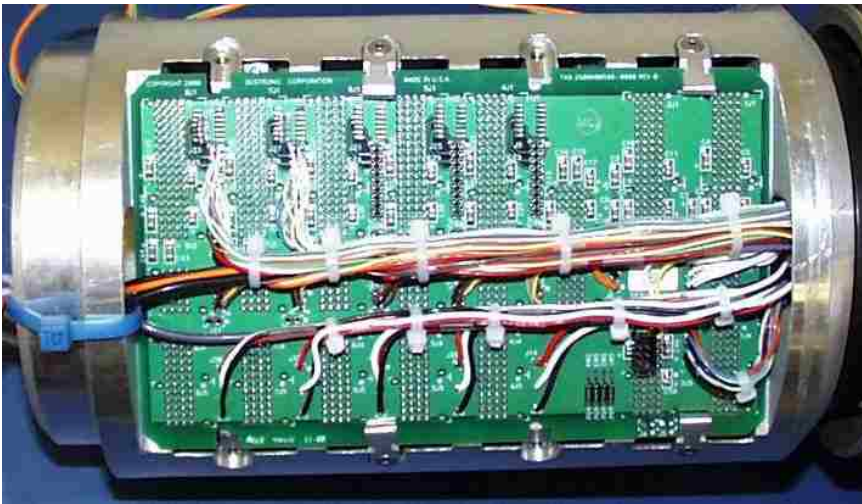
Figure 19 - RANGER NBV-II/RTSX Backplane Connections

Once all the connections were made and the proper encoder counts set in

configuration, the motor was recognized by the driver's current tuning procedure,

establishing that the motor was functioning. Further testing also confirmed that the

hall effect sensors and encoders were functional and demonstrated that the Whistle

modules are suitable for use in the RTSX/NBV-II manipulators.

Chapter 5: Testing

The COTS motion controllers are an inherently complicated product that includes communications, electronics, control loops and motor commutation and is being implemented in a similarly complex robotic manipulator with different actuators and gearing. There is no way to compare these modules to the existing custom electronics except through implementing the system in one of the lab's manipulators and evaluating the performance either in absolute terms or against the custom hardware. The tests in this chapter evaluate the current, velocity, and position performance of the modules in addition to an analysis of the self-tuning performance of the configuration software. All testing was completed with the Whistle drivers controlling single or multiple joints of existing SSL manipulators in order to gauge the performance for this specific application.

Test Setup

Faro Arm

Both the static positioning and the repeatability measurements require that the end effector position of the manipulator be measured accurately at each point. A Faro portable Coordinate Measuring Machine (CMM) Platinum series 6-DOF measuring arm was used for all measurements. Unlike traditional gantry type CMMs, the Faro arm calculates a three dimensional position using the precise rotary encoders in an unpowered arm. The CMM used was a Faro model P0802, serial number

P08020503418, with a 2.4 meter reach and a 2 sigma single point accuracy of +/-

0.025 mm. This is the same unit used for the original testing on the DXL

manipulator.

Recording Measurements inside the driver

All distributed motor drivers collect and use data regarding the motion of the motor

and the status of the drive. Additionally, motion controllers must calculate values

such as velocity for use in their control loops.

One of the advantages of the Elmo SimplIQ drivers is that they have a

debugging feature that can store many of the measured and calculated values over a

given time frame and then output them to the configuration software. Up to 8

different values can be stored at a user-defined rate and the internal memory has

space for 1024 measurements of each value. Most of the testing of individual joint

performance is taken with the internal sensors and output in this manner, as it is

impractical to measure them externally.

Using the internal measurement functions

Configuring what measurements are taken and their timing is done through

the "Motion Monitor" built into the Elmo Composer software. After establishing a

CAN connection to the device in Composer, this tool can be accessed by selecting

"Motion Monitor" from the "Tools" menu. At this point, the Motion Monitor will

appear on the bottom half of the screen. The data collection tools are located in the

recorder section of the window as shown in Figure 20Figure 20.

Figure 20 – Elmo Composer Motion Monitor Control Panel

There are 8 fields that can be selected by using the drop down menus under the

signals tab. The most commonly used values are available by default, but nearly

every internal variable is available by using the mapping tab. The resolution and

timing controls are located in the lower left of the window. Resolution defines the

time between data points as would be expected. The maximum record time is largely

a function of the number of signals and resolution as there is a limit to the onboard

ram that can be used for storage during a single run. For most of the testing done in

the following chapters, the resolution and timing parameters were selected to give the

highest resolution while gathering the full length of the task.

The trigger section, located on the right side of Figure 20 controls the start of data

acquisition. The trigger most commonly used for the data acquisition in this thesis is

the start of a motion command, but any of the measured values selected earlier can be

used when they reach a specified threshold value. The delay field specifies a

percentage of the "Max Record Time" that the controller will wait after the trigger before taking data.

Positioning and Repeatability

ANSI/RIA R15.05-1.1990 Standard

One of the continual challenges on the mechanical side of dexterous robotics is determining a means to measure and compare the relative performance of different systems due to both their complexity and flexibility. Most of the work developing standards has centered on industrial robots in order to compare competing models from multiple manufacturers for use on assembly lines. Taking this into account, the existing industrial standards can provide the basic framework for testing after a number of modifications are implemented to make them applicable. In the testing presented in this section, any major changes to the standards will be noted.

The American National Standards Institute (ANSI) and its member organization, the Robotic Industries Association (RIA), is the dominant US standards organization. The standards ANSI/RIA R15.05-1 and ANSI/RIA R15.05.2 are the major standards for static positioning and repeatability, and dynamic performance respectively. The testing performed for the manipulators in this thesis focus on the static and repeatability testing as the dynamic performance is significantly defined by the higher level computer control software. At the time of writing, a real time, closed loop, manipulator control program is still under development.

In 2005, the Space Systems Lab tested the performance of the DeXterous Left, or DXL, manipulator of the NBV-II RANGER system according to a modified version of the ANSI/RIA R15.05-1 standard. The tests included static position accuracy, repeatability, and static compliance. The testing in this section attempts to repeat that original test as much as possible on the NBV-I Camera Arm and the SAMURAI manipulators. Testing was conducted for static positioning and repeatability but not for compliance as the highly geared nature of all three manipulators means that most of the compliance measurement results from the deflection of structural members and the harmonic drives; this will vary significantly based on the different ages, harmonic drive gearing, and materials.

Test Setup

Determining the Denavit-Hartenberg Parameters

Denavit-Hartenberg or D-H parameters specify a minimal system for representing the position and orientation of a robot joint with two rotations and two translations. The parameters can be either constant, such as fixed offsets, or variable in cases like joint rotations. All of the D-H parameters represented here utilize the modified D-H tables(7).
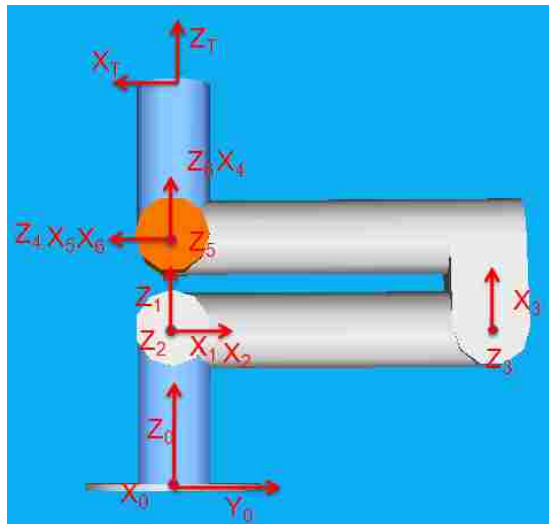
All of the D-H parameters were measured using the Faro CMM. The two major challenges are that the link frames are located inside the robot along the internal axes of rotation and the Faro arm (or rather its operator) is far more accurate when measuring a target hole or a large scale feature. Thankfully, the RANGER NBV-I arm has a large number of finely machined symmetrical features. The measurements

shown in Table 3 were not measured directly, but are rather a geometric composite that is based on the intersection of features such as link segment axes. In his related research on higher level control, D'Amore validated the accuracy of these measurements through the use of an optimization scheme (8).

Table 3 - Denavit-Hartenberg parameters for NBV-I

i	α_{i-1} (deg)	a_{i-1} (m)	d_i (m)	θ_i (deg)
1	0	0	0.2491	θ_1
2	90	0	0	θ_2
3	0	0.5589	0	θ_3
4	-90	0.1514	0.5388	θ_4
5	90	0	0	θ_5
6	90	0	0	θ_6
T	0	0	0.2666	0

Figure 21 - RANGER NBV-I Coordinate Frames

The static positioning test defined by the standard takes the form of measuring and analyzing points located in the test a standard test plane. The test plane, shown in Figure 22 is positioned and sized with respect to the robot and the workspace, but is always an inclined plane with measurement points along the upper and lower edges.
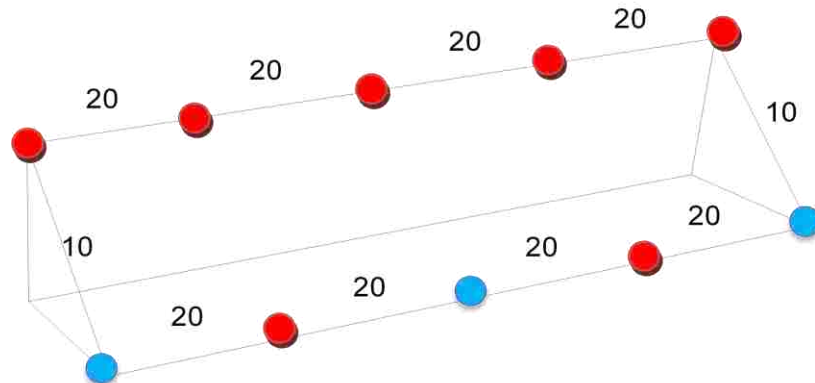
Figure 22 - Standard Test Plane – dimensions in cm

While the Faro arm is capable of measuring points on the robot, it is both impractical

and inaccurate to measure this manually. Instead, for the original testing done on

NBV-II in 2005, a mounting fixture, shown in Figure 23, was designed to mount on

the NBV-II tool adapter and move the Faro arm in sync with the RANGER arm. The

original tool mounting pattern was duplicated so that the mounting fixture could be

attached to the NBV-I manipulator. The full test setup with the active RANGER arm

moving the Faro arm is shown in Figure 24.

Figure 23 - Faro-RANGER Mount
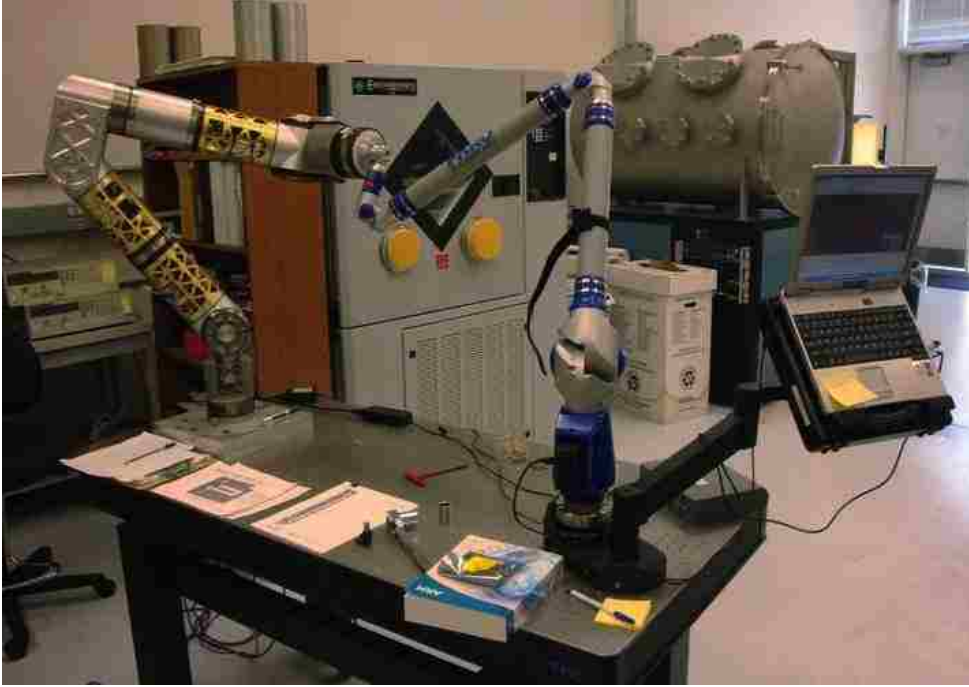
Figure 24 - Test Setup for Static Positioning and Repeatability

The testing was composed of acquiring 50 data points in a random fashion from the 10 test points identified in Figure 22. After rotating and translating the positions from the Faro frame into the RANGER base frame, the error for each point was calculated using Equation 1 by calculating the distance between the commanded coordinate and the measured position. Positional accuracy and standard deviation were then calculated using Equation 2 and Equation 3 for the entire data set.

$$d_i = \sqrt{(X_{ai} - X_{ci})^2 + (Y_{ai} - Y_{ci})^2 + (Z_{ai} - Z_{ci})^2}$$

Equation 1 - Distance Formula(9)

$$\bar{d}_{PA} = \frac{1}{N}\sum_{i=1}^{N} d_i$$

Equation 2 - Positional Accuracy(9)

$$S_{PA} = \sqrt{\frac{\sum_{i=1}^{N}(d_i - \bar{d}_{PA})^2}{N-1}}$$

Equation 3 - Standard Deviation(9)

65

The measurment of repeatability is very similar except that the measurments are in relation to the centroid of the mesurments for the same point. Also, only three points are used for the analysis, specifically the center, far left, and far right points on the bottom of the standard test plane. It should be noted that the repeatability testing here is based on the 2005 RANGER testing and not the full standard. The most notable difference is that the standard defines a warm up time for the robot arm, but the Faro CMM does not have a timestamp function making this impractical. While this reduces the ability to compare the arm to generic robots, it is not an issue for the primary comparison to the NBV-II DXL. The centeroid of the point clouds was derived from the static positioning data, allowing for the calculation of the mean radius for each point using Equation 4. The mean repeatability and standard deviation are calculated using Equation 5 and Equation 6 respectivly, but N is now the number of times all of the test points were measured.

$$r_{ai} = \sqrt{(X_{ai} - \bar{X}_a)^2 + (Y_{ai} - \bar{Y}_a)^2 + (Z_{ai} - \bar{Z}_a)^2}$$

Equation 4 - Mean Radius for Point A Measurements(9)

$$\bar{r}_{REP} = \frac{\sum_{i=1}^{N} r_{ai} + \sum_{i=1}^{N} r_{bi} + \sum_{i=1}^{N} r_{ci}}{3N}$$

Equation 5 - Mean Repeatability(9)

$$S_{REP} = \sqrt{\frac{\sum_{i=1}^{N}(r_{ai} - \bar{r}_{REP})^2 + \sum_{i=1}^{N}(r_{bi} - \bar{r}_{REP})^2 + \sum_{i=1}^{N}(r_{ci} - \bar{r}_{REP})^2}{3N - 1}}$$

Equation 6 - Standard Deviation for Repeatability(9)

Test Results

As can be seen in Table 4, the positioning results from the NBV-I camera arm show a reduction in error by a factor of 2-4 over the corresponding performance of the DXL

manipulator. Part of the improvement in both readings is going to be based on the

higher precision encoders (1024 counts vs. 763 pulses per revolution), but that alone

cannot account for the entire performance gain. The data suggests that the

commercial drivers perform better in static positioning than the NBV-II electronics,

likely due to direct commutation control and microstepping enhancements. This

coincides with qualitative responses from teleoperation operators on the smooth low

speed control compared to the NBV-II arms.

Table 4 - Static Positioning and Repeatability Results

	NBV-II DXL	*NBV-I Camera*
Mean Position Accuracy	22.8 mm	5.11 mm
Standard Deviation	4.3 mm	2.0 mm
Mean Repeatability	0.495 mm	0.190 mm
Standard Deviation	0.408 mm	0.113 mm

Single Joint Current Testing

Overview

The basis of any motion in a servo motor is the torque loop.

The torque generated by the given current is given as:

$$T = K_t \cdot I$$

Equation 7(10)

Where Kt is the torque sensitivity as specified by the manufacturer and T and I are

torque and input current respectively. It should be noted that while Kt is specified as

a constant, the relationship is not completely linear. Tsl is specified as the "max

torque for linear Kt" and specifies the point where Tsl = 0.9 * Kt * Isl.

Test Setup

Torque can be measured directly or by the measurement of an exerted force at a known radius. The latter method was chosen due the availability of measuring equipment. The first tests were conducted using the an A&D electronic balance, serial number 3901260, with the ability to measure mass in 0.5 gram increments. This scale was preloaded with 38 kg of lead weights and the first 2 joints of the NBV-I camera arm were used to lift a portion of that mass. This test setup is shown in Figure 25.

Figure 25 - Electronic Balance Test Setup for Current/Torque Tests

The second joint of the NBV-I manipulator is the one being used for this test with a Kollmorgen RBE-1812-A motor with a Kt of 0.257(10) and a harmonic drive with a 200:1 gear ratio. An extra skeletonized segment was used to create a 0.485 meter

lever arm positioned perpendicular to the load. A ratchet strap was used to attach the platter on the scale in order to provide a softer stop to prevent damage to the arm.

One of the challenges with current testing as shown here is that in order to overcome static friction, a suitably high current must be commanded, but once static friction has been overcome, the drive has a tendency to accelerate until coming to an abrupt halt when it makes contact with the sensor or in this case tensions the strap.

For the second round of testing, the experiment was repeated with a Chatillon Centurion DWT-5000 crane scale. This scale has a maximum capacity of 5000 lbs and measures in 2 lb increments. Because of the limited precision of this unit, it was used to supplement the earlier testing with smaller loads.

Figure 26 – Crane Scale Test Setup for Current/Torque Test

The testing procedure was the same for both test rigs. The motor was set to operate in torque mode and given a fixed torque command of 0.5 Amps in order to tension the strap and prevent a high speed acceleration followed by an abrupt deceleration as described above. At this point, the full current was commanded. Measurements were conducted at approximately 3 seconds which was the time it took for either scale to settle.

Results

The data collected is shown in Figure 27 and Figure 28 for the balance and the crane scale testing respectively. The torque plotted is the torque output of the motor applied to the load. The measured torque is plotted against the theoretical output from the motor based on Equation 7.

One interesting anomaly is that some of the lower current tests suggest higher than expected efficiencies. It is hypothesized that this is a result of the inertia of the arm as it is stopped by the strap causing a higher load than just the current and then the static friction helping to hold the load.
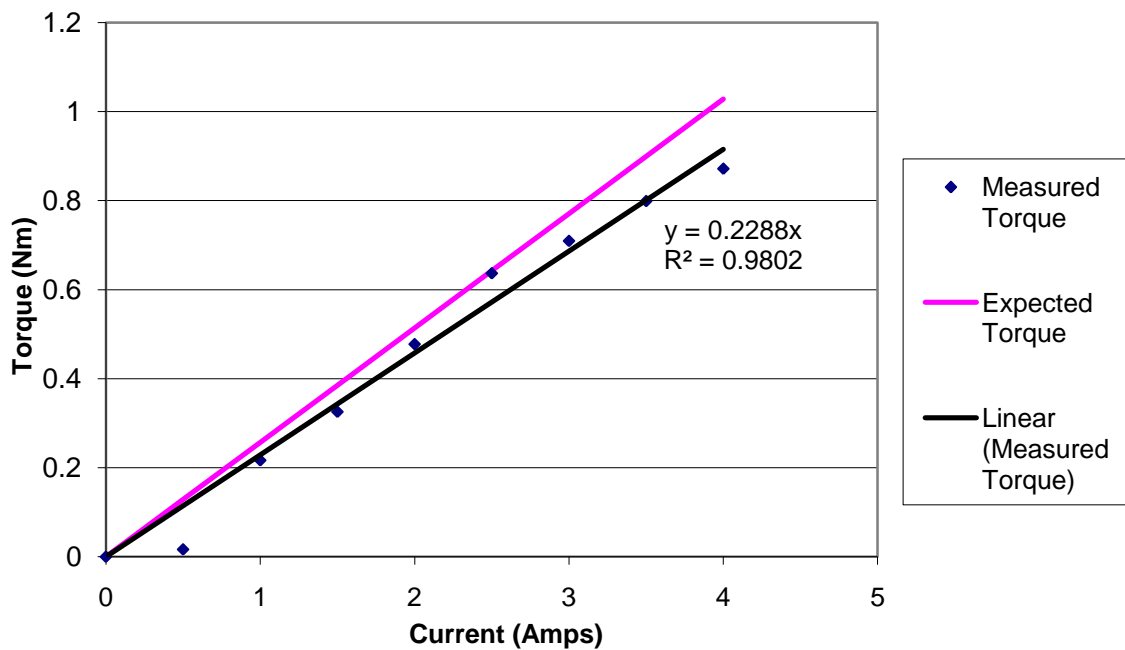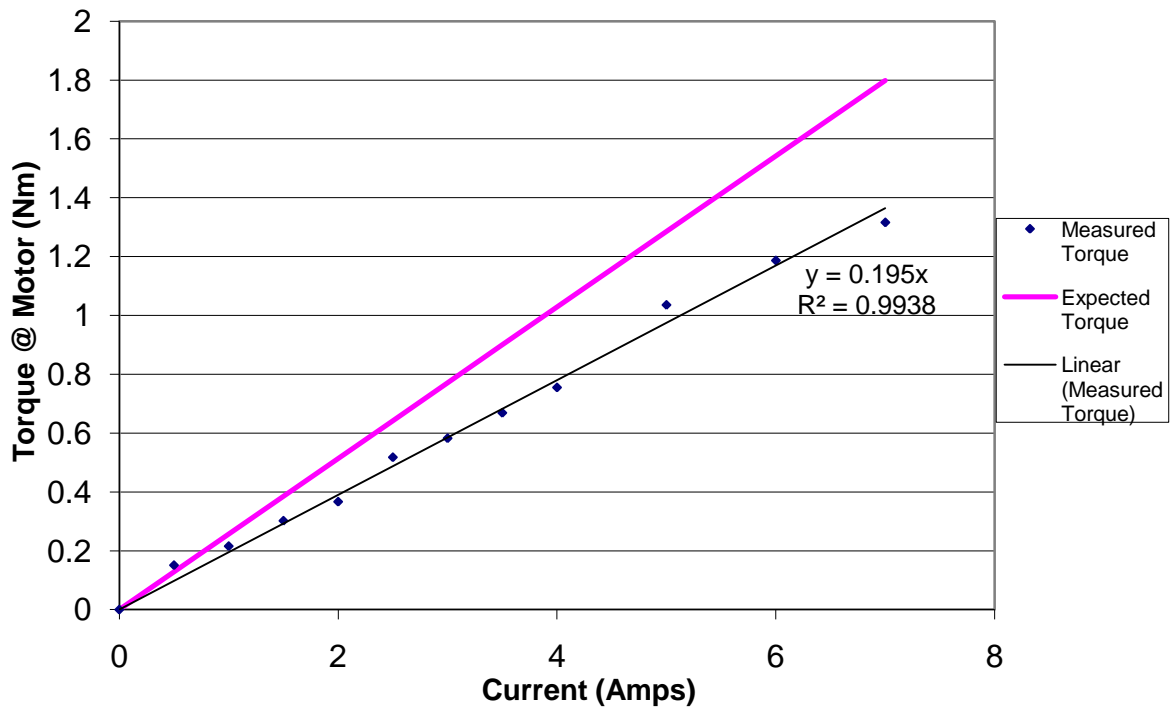

Figure 27 – Measured Output Torque vs. Motor Current – Light Loads

Figure 28 - Measured Output Torque vs. Motor Current – Heavy Loads

The slope of the linear curve fits can be converted into an efficiency value for the

drive. This efficiency includes the losses from the entire drive train, but it is

dominated by the harmonic drive. A harmonic drive has a high variability in

efficiency based on gear ratio, tooth type, temperature, load, speed, and

lubrication(11). The RANGER design team assumed that the similar 1:160 harmonic

drives had a worst case efficiency of 62% as used in RTSX(12). Based on this, the

measured efficiency from the low and high torque tests of 89.0% and 75.9%

respectively are reasonable, although the former is close to the theoretical limit of

90% and may be overly high as discussed earlier.

Offset Load Single Joint Velocity Testing

Overview

In a dexterous manipulator, operating a joint at a specific velocity is atypical as an end unto itself, but tracking a desired velocity is critical for proper movement along a trajectory. When operating in a position-based mode, motion commands are typically handled as a trapezoidal velocity trajectory where the motor accelerates at a given rate to a set velocity and maintains the velocity until it decelerates. A sample of this motion, a translation of 1000 encoder counts on the shoulder joint, is shown in Figure 29. The trapezoid is the commanded velocity and the damped curve shows the actual velocity and shows the need for accurate velocity control.
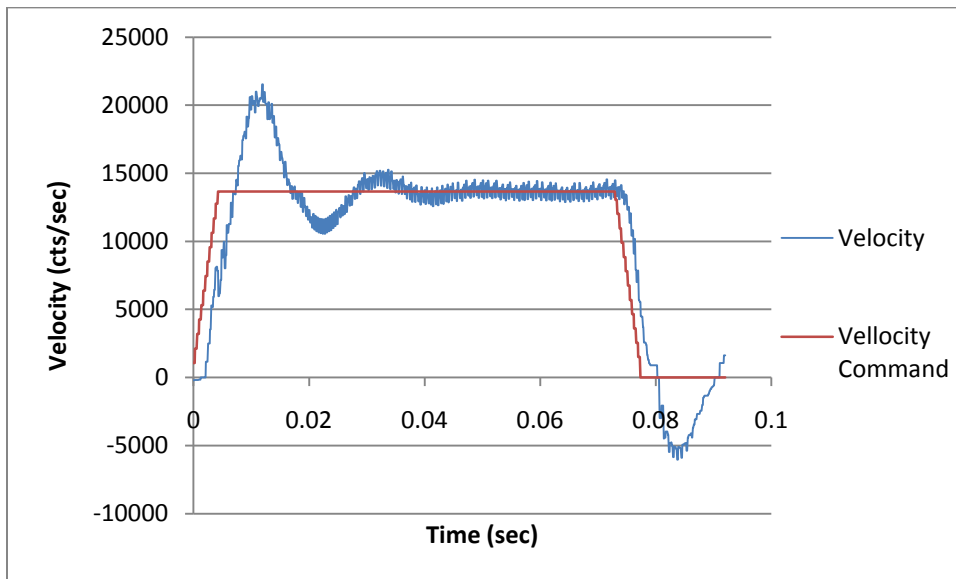
Figure 29 - Trapezoidal Velocity Profile

A system that is properly tuned should be capable of maintaining a set velocity. One of the issues with manipulators such as the RANGER arms is that the loads on a joint can vary significantly. Variations in joint loads are commonly from two sources, namely external loads and the arm pose. External loads are commonly payloads that the arm is moving or the environment itself for cases like scooping soil. Especially in earth gravity, the pose of the arm itself can be a significant variable load on a joint. In the case of the NBV-I Camera Arm, the load on the shoulder joints is the rest of the arm, with a mass of 27kg, with a highly variable center of gravity. When in microgravity, there is no gravity, but often times larger objects are being manipulated that place large inertial loads on the manipulator during motion.

Test Setup

The continuous rotation velocity testing was conducted on the NBV-I Camera Arm. The final rotational joint, joint 6, was chosen for the testing as it has a tool mount and can rotate infinitely without running into any hard stops or placing any strain on the internal wiring.

Joint 6 consists of a Kollmorgen RBE-01812A motor and a 100:1 harmonic drive. The joint is intended to rotate freely and nominally would only handle the load applied by a set of cameras attached to it. The motor itself is rated for 4.91 amps of continuous current and up to 21.3 amps for 10 seconds.

A mount, consisting of 1/8" aluminum was constructed to hold an offset mass 25cm from the center of the tool drive. The masses used for the testing were lead blocks

with a ¼" bolt hole drilled through them. Up to 4 blocks were mounted by securely

bolting them to the aluminum lever arm as shown in Figure 30.
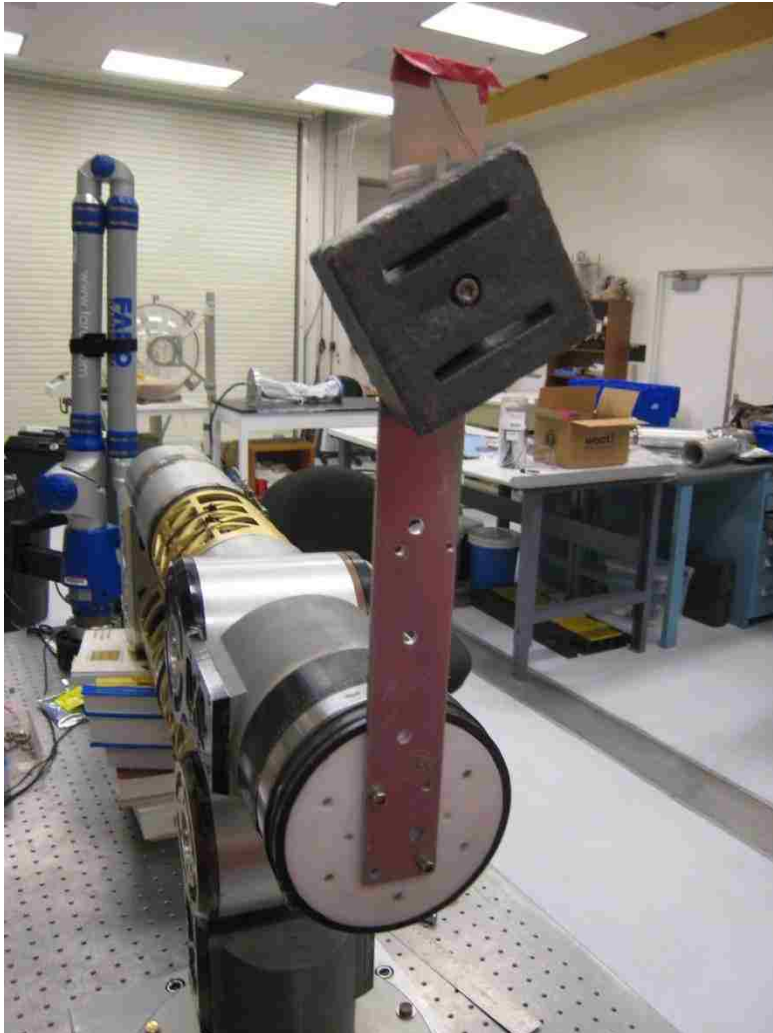
Figure 30 – Endefector Offset Load Test Configuration

The test runs each started with the lever arm and mass held vertically above the actuator. The starting position was determined to less than 1° through measurement and a small pendulum that was added to the lever arm as shown in Figure 31. The arm then rotated through approximately 2.25 rotations in the counter clockwise direction with a motor speed of 200 RPM.

Figure 31 – Pendulum Used for Alignment in Continuous Rotation Test

Results

The results of the continuous rotation testing did not show a systemic trend of error in terms of load as would be expected if the system was unable to cope with more massive offset loads. Table 5 shows the results of the testing including RMS of the velocity. The error is very low and the velocity tracks well with the command, but

76

the noise is significantly higher. This suggests that additional filtering of the velocity

loop would yield a performance improvement.

| Mass	Mean Velocity	Mean Error	RMS
554.5	2.002	0.085 %	4.82 %
2212	2.001	0.074 %	4.42 %
6577.5	2.003	0.154 %	4.74 %
4368.5	2.001	0.031 %	5.07 %
8766.5	2.005	0.227 %	4.60 %

Table 5 - Continuous Rotation Velocity

The current output from the driver is shown in Figure 32 through Figure 36.
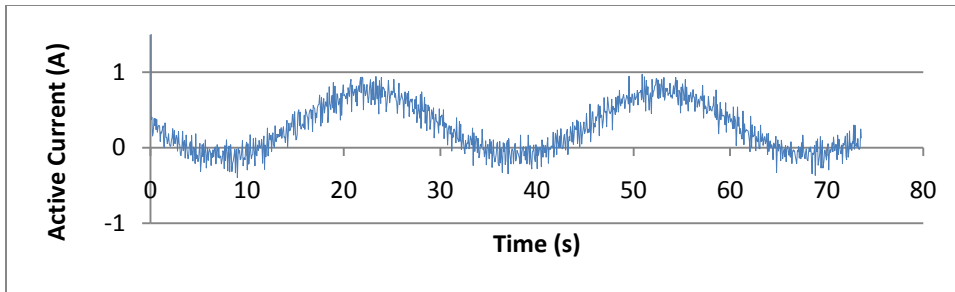
The data shows that the testing with small masses, and thus small differences in

torque, is dominated by the friction internal to the joint. As the mass is increased, the

exterior torque on the joint becomes the dominant driver for the active current.
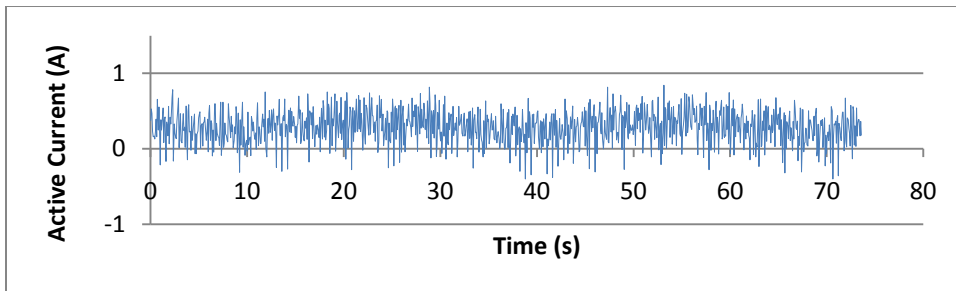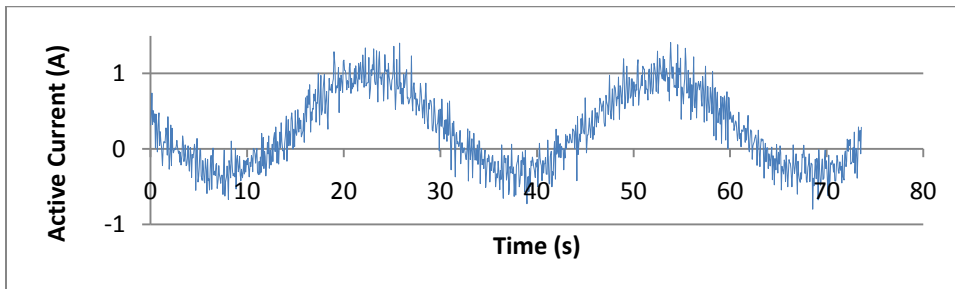

Figure 32 - Current for 554.5g Continuous Velocity Test
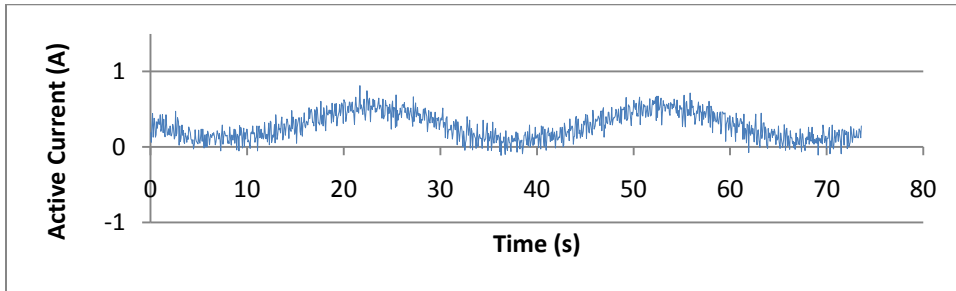
Figure 33 - Current for 2212g Continuous Velocity Test

Figure 34 - Current for 4368.5g Continuous Velocity Test

Figure 35 - Current for 6577.5g Continuous Velocity Test
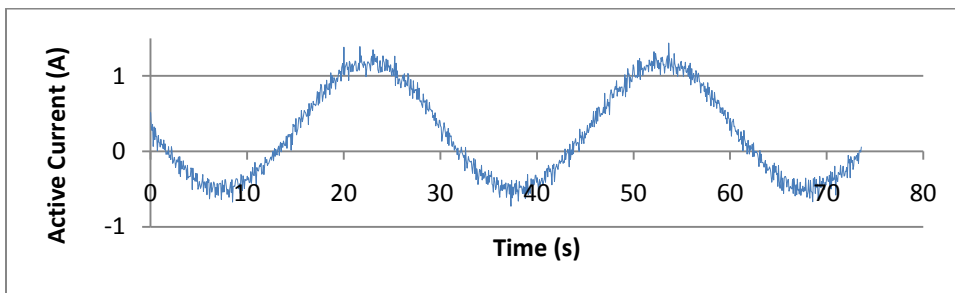
Figure 36 - Current for 8766.5g Continuous Velocity Test

Fractional Rotation Test

Overview

In typical usage, it is very rare for a manipulator joint to be driven continuously at a
set velocity. The exception to this is a tool drive, which often is tasked with motions
such as rotating a screw or brush. On the ranger NBV-I Camera Arm there are 6
joints, with three being "continuous rotation," and three having hard stops that keep

them from rotating a full revolution. Even of the continuous rotation joints, two of them are not rotated more than a single revolution as doing so may damage the arm by putting tension on the internal wiring. It should be noted that the discussion of this "fractional" rotation test refers to a 180° rotation of the manipulator but nearly any large translation results in multiple rotations of the motor because of the 100:1 or 200:1 gear ratios in the harmonic drives.

The torque applied during the continuous rotation testing was beginning to flex the structure of the manipulator, as it was not designed for significant operation with an offset tool load such as the test configuration. It was decided that it would both be safer for the robot and more representative to complete the heavier testing using a single joint that was designed to take a heavy offset load.

Test Setup

The actuator used in the fractional rotation velocity testing was joint 2 of the NBV-I Camera Arm. This joint consists of a Kollmorgen RBE-01812A motor in a pitch configuration. The gearing used in this actuator is a 200:1 harmonic drive so the torque applied to the motor is half of that compared to the testing done on joint 6, but the speed is reduced by 50% as well. The mass consists of larger lead blocks that are attached to the end of the same skeletonized segment used for the torque/current testing as shown in Figure 37.
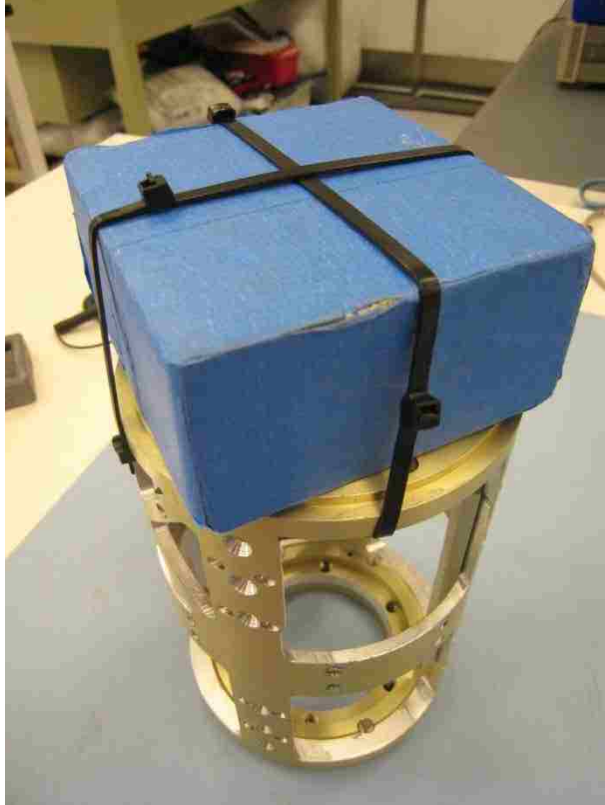
Figure 37 – Mass Attachment for Fractional Rotation Velocity Test

Results

Table 6 - Partial Rotation Velocity Results

Mass(g)	Mean Velocity (RPM)	Mean Error	RMS
0	2.000	-0.0028 %	1.58 %
6255.5	2.000	0.0067 %	1.63 %
11170.5	2.001	0.0491 %	1.57 %
11610	2.000	0.0199 %	1.57 %
23594	2.001	0.0415 %	1.54 %
35496.5	1.999	-0.0457 %	1.39 %

Once again, the test results did not show any breakdown in performance.

Unlike what was seen in the earlier testing, there was a more significant trend towards

a lower variation in the velocity with the heavier loading. This can be attributed to a

larger portion of the load resulting from the mass and less from the friction internal to

the actuator. Additionally, as the actuator was geared at 200:1 instead of 100:1 and the velocity was similarly doubled, the resulting error is less on a percent scale.

Controls

Overview

The research presented in this thesis is an overview from a system level on the architecture and the motor drivers chosen. In that context, the controllers utilized and their performance is a significant aspect of the overall performance and applicability for use in serial link manipulators. One of the important questions is how well the autotune functions for the drives work for highly geared robots such as RANGER. This section attempts to address these items and prepare a foundation for a detailed controls analysis and optimizations as part of future research and design.

The control solution integrated in to the Whistle modules can be broken down into three separate areas: a current controller, a velocity controller, and a position controller.

Current Controller

The current control loop is used to optimize the performance of the current flowing through the motor when it is commutated. The SimplIQ drivers utilize a PI controller for both the active and reactive currents, the Q and D controller respectively. The Q, or active component, controller input is the desired torque command and the D or reactive controller always has a desired value of zero. Both of these loops operate in

a similar fashion with the KP and KI gains being divided by the voltage to generate the PWM duty cycle and thus the motor drive voltage.

The KP and KI gains are set transparently to the user by the automatic tuner included in the configuration software. They are determined by sending step current commands and a system identification process. The whole process is automated and takes less than 5 minutes to run on a PC connected to either the serial port or CAN bus. The only area where the configuring engineer has any input is the power supply filter. The filter defaults to a bandwidth of 50 Hz, but may be adjusted up or down depending on the impedance and the ripple coming from the power supply. For all testing conducted as part of this thesis, this filter was left at the default 50 Hz value.

The values of KP and KI are optimized for a specific brushless DC motor. This means that while motors of the same make model will be very close, different motors with different windings will differ significantly in the appropriate gains values. In the older architecture using motor drivers such as the DDC PW-82520, this was an analog controller block with two resistors and one capacitor installed during board assembly. This analog PI regulator is shown in Figure 38.
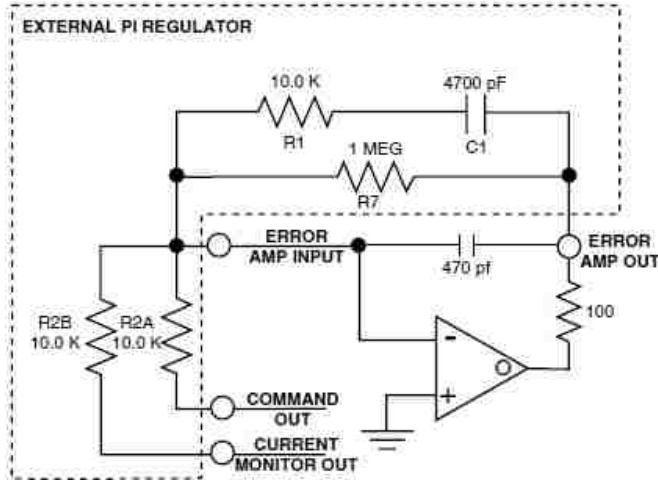
Figure 38 – Analog Current Gains Setting on PW-82520(13)

Not only does this mean that there was a time consuming (and less precise) trial and error tuning method, but it means that a motor driver card optimized for one specific motor/model would have diminished performance on a different motor, limiting the modularity of the hardware and increasing the issues relating to porting hardware or replacements.

Velocity Controller

The velocity auto tuning process consists of six steps that are simply identified by number during testing.

1. The software determines a simple, low bandwidth, velocity control loop that will be used for all further steps during the system identification process.

2. The module uses the low bandwidth loop identified in the first step to drive the motor at a constant velocity. At this point, the current command is tested at different frequencies. These commands as well as the velocity and position responses are recorded for use in step three.

3. The frequency responses collected in step two are used to calculate the open loop transfer function for the actuator. This transfer function is for the entire mechanical drive from the motor to the encoder and includes mechanical and sensor effects and delays.

4. The drive transfer function is calculated and the gain schedule table is filled in for a range of velocities up to the maximum specified for the motor.

5. A sample step response, either from default values or those specified by the operator, is commanded and the resulting velocity and current graphs are generated for the user. Additionally, using an integrated portion of MATLAB, the transfer function and bode plot is displayed for the user.

Position Controller

The position controller implemented in the Whistle and all of the SimplIQ drives consists of a nested configuration of a proportional position loop that outputs a speed command for the velocity loop that was described in the previous section. From a position loop perspective, this results in a controller that is functionally equivalent to a PID controller while remaining modular and facilitating an easier operation in both velocity and position modes. It should be noted that this means the controller maps to a PID implementation as shown in Equation 8, but does not allow PID gains to be set directly, potentially complicating the implementation of PID controllers in future research.

$$PID \begin{cases} K_I \\ K_P = K_P^{Pos} \times K_P^{Speed} + K_I^{Speed} \\ K_D \end{cases} \begin{matrix} K_P^{Pos} \times K_I^{Speed} \\ \\ K_P^{Speed} \end{matrix} \Bigg\} PIP$$

Equation 8 – PID equivalence of PI velocity and P position loops

Velocity Controller Self-Tune Performance

In order to test the self-tuning capability of the Whistle motor drivers and the

Composer configuration software, a test was conducted using the gains generated by

the self-tuning procedure and then with gains modified by +/- 10%. For this test, the

first joint of the manipulator was tuned using the built in self-tuning feature in

Composer. The resulting step function is plotted in Figure 39, and the computed

gains are shown in Table 7, for 200 RPM were then identified and extracted from the

gain schedule that was automatically generated. At this point, gain scheduling was

deactivated and the extracted gains were programmed directly into the Whistle. The

Whistle was put into a diagnostic mode and commanded to run at -200 RPM and then

+200 RPM. The data was analyzed for the greatest overshoot (always the first one),

and the greatest undershoot. From the overshoot, the damping ratio was then

calculated using Equation 9. The test was then repeated with the modified gains.
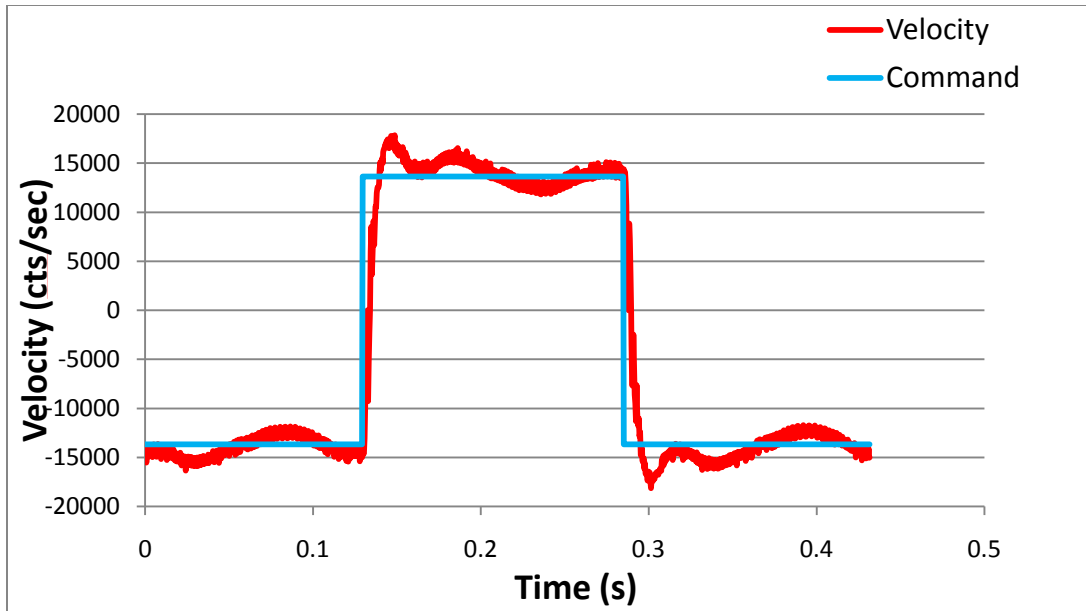
Figure 39 - Velocity Step Function

Table 7 - Self-tuning Velocity Gains

	Gain
K_P^{Speed}	88.343
K_I^{Speed}	8369.180

$$\zeta = \frac{-\ln\frac{\%OverShoot}{100}}{\sqrt{\pi^2 + \ln^2\left(\frac{\%OverShoot}{100}\right)}}$$

Equation 9 - Damping Ratio

Table 8 - Autotune Velocity Performance

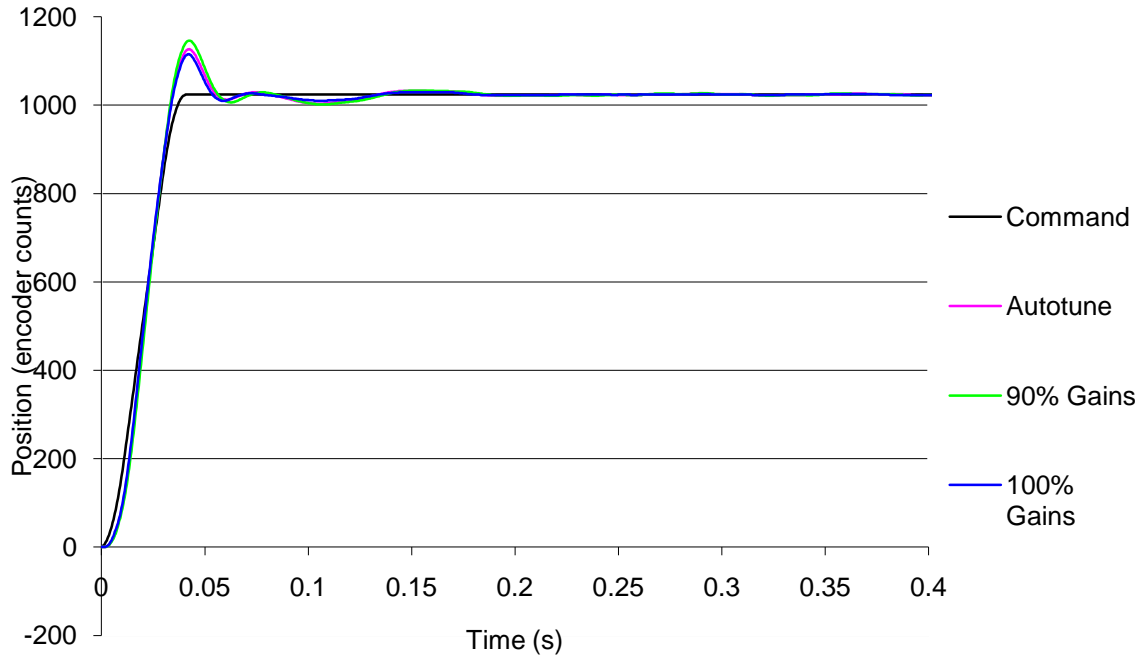| Gains	Overshoot (RPM)	Undershoot (RPM)	Damping Ratio
90% of Autotune	68.30	28.1	0.490
	17.08 %	7.014 %	
Autotune	61.7	27.0	0.511
	15.41 %	6.748 %	
110% of Autotune	64.31	33.9	0.503
	16.08 %	8.477 %	

The results in Table 8 show that the self-tuned gains performed better than the positive or negative modifications. Both overshoot and undershoot were lowest in the self-tuned gain case and the damping ratio was the highest. Also evident from this data is that the controller is rather robust and tolerates moderate adjustments in gains with a minimal hit to performance. On the other hand, the 15.41% overshoot and damping ratio of 0.511 is a little low for an ideally tuned system, suggesting that there may be some room for improvement.

Position Controller Performance

The same procedure outlined above was also carried out for the position controller. As shown in Table 9, the gains for the velocity loop remained the same, but Kp was also set for the position loop. The test case for the position loop was a commanded motion from rest to a position 1024 encoder counts in the positive direction. Commanding an instantaneous position change is problematic as it results in a full saturation of the motor current and high acceleration until the position is reached, with the current limit programmed into the controller restricting the motion, which then appears over damped. Instead, a high acceleration, high speed (500 RPM), trajectory was used.

Table 9 - Self-tuning Position Gains

	Gain
K_P^{Speed}	88.343
K_I^{Speed}	8369.180
K_P^{Pos}	84.774

Figure 40 - Autotune Position Performance

Table 10 - Autotune Position Performance

Gains	Overshoot	Undershoot	Damping Ratio
90% of Autotune	122	22	0.561
	11.9 %	2.15 %	
Autotune	103	21	0.590
	10.1 %	2.05 %	
110% of Autotune	91	14	0.610
	8.89 %	1.37 %	

The results shown in Figure 40 and Table 10 show a similarly tightly grouped set of

results. In this case, the 110% gain solution resulted in the best performance with

minimal over and undershoot and a damping ratio of 0.610. This does show that the

self-tuning results can be tweaked to realize performance improvements.

As a first order comparison of the performance, the loop was tuned using a modified

version of the Zeigler-Nichols tuning method. The Zeigler-Nichols tuning method

was originally introduced in their 1942 seminal paper. In this paper, they outline a method of tuning PID gains by measuring the "ultimate sensitivity" of the system. In order to do this, the proportional gain is raised until there is an overshoot and then until the overshoot is no longer damped and the resulting sine wave has a constant amplitude. At this point, the proportional gain is the ultimate sensitivity or gain, K_U, and the period of these oscillations is called T_U. From these values, the appropriate gains are defined as shown in Table 11.

Table 11 - Ziegler-Nichols PID Gains

K_P	$0.6K_U$
K_I	$\dfrac{2K_U}{T_U}$
K_D	$\dfrac{T_U K_U}{8}$

In order to implement this method on the Elmo Whistles, the manual position loop tuning interface was used. In order to adjust the proportional PID gain, as per Equation 8, K_I^{Speed} on the Whistle was increased. From this, the Zeigler-Nichols PID gains were calculated according to Table 11. One complication is that there is no real solution for converting the Whistle P and PI loops to a PID loop using Equation 8. Instead, the gains were chosen to fit as close as possible. The resulting controller was then used for a similar test to the autotune position loop and resulted in a damping ratio of 0.63, only a very minor improvement over the self-tuned values.

The basic evaluation of the controllers and the self-tuning performance demonstrates the capabilities of the self-tuning feature. Self tuning provides an accelerated

procedure that results in acceptable performance for a reduced configuration effort. The self-tuning is not a replacement however for a proper controls engineer and manual tuning for a robot that will be used long term and justifies the time investment. The autotuning is still useful as it provides a starting point for manual tuning. In addition to tuning the control loop gains, the harmonic drive adds significant noise to all the control loops, something that can likely be mitigated by adding a custom filter to the controller.

Chapter 6: Conclusions and Future Work

Conclusions

This research demonstrates that a COTS motor controller communicating over CANopen is fully capable of replacing a custom Local Processing Unit in the dexterous robots used by the Space Systems Lab. Specifically, this also suffices as a test of the Elmo Whistle motion controllers and paves the way for both their use and the use of other SimplIQ controllers in future systems. Testing was completed on current, velocity and position control modes with performance that meets or exceeds the high standards established by the RTSX development program. In the process of conducting this research, the NBV-I arm was transformed from a museum piece to a functional manipulator and the similar implementation on SAMURAI is allowing that program to move forward with underwater and sea trial planned for the coming months. Additionally, an evaluation of the self-tuning capabilities of the SimplIQ line has been conducted and lays the groundwork for future controls studies. This will significantly improve the time and cost required to implement the electrical systems in future robots and enable a larger focus on the capabilities and research dependant on the function of the whole robotic system.

Statement of Work

- Evaluated data bus options and chose CAN for network
- Evaluated COTS motor controllers and selected a unit
- Designed and fabricated the printed circuit boards and support circuitry to integrate with RANGER and SAMURAI

- Evaluated the performance of the Elmo SimpIQ Whistle modules

  - Current performance

  - Offset load velocity testing

  - Static Positioning

  - Repeatability

  - Evaluated self-tuning feature

- Implemented the first new electronics architecture at the SSL in over a decade

- Returned the NBV-I Camera Arm to operation

- Implemented the electronics for SAMURAI

Future Work

The future work related to this project can be broken down into two categories,

namely implementing features or improvements and the design of a follow on system.

Controls

The scope of this thesis only included a very basic controls implementation that could

be expanded upon and would improve the dynamic performance. The gains could be

manually tuned in order to reduce the overshoot which is not ideal at this time.

Additionally, the design of a filter to reduce the high frequency noise should be very

workable using the tools built into Elmo Composer.

Gains Scheduling

One of the features of the COTS modules that have not yet been applied is the gains

scheduling. While the controller that runs on each module is linear, gains scheduling

allows for implementing a family of gains similar to a lookup table where the gain is adjusted based on current or velocity.

Absolute Encoders

The main capability that is missing from the system demonstrated here is the ability to interface with the absolute encoders. The incremental encoders provide higher precision than any reasonable absolute encoders, but lack the ability to determine the initial pose of the arm. It should be noted that the SAMURAI arm lacks any absolute encoders and the NBV-I camera arm contains encoders but lacks documentation, and verbal reports from the original team indicated that the encoders may never have worked reliably due to interference from the motor. That said, the operations of a manipulator with absolute encoders (or a workaround) are far simpler as they do not rely on starting from an absolutely known position.

Known Position

The current method of determining the absolute position of the manipulators is to start them at a known pose (specifically, known joint angles) that all encoder measurements are then relative to. This method is highly dependent on the precision that the original pose is known. One option that has been used by the lab in the past is to develop a cradle that securely restrains prior to power up. Alternatively, with proper high level control, the manipulator can run a configuration motion and move its joints into hard stops, thereby identifying its position from an unknown starting point. Either of the methods can be improved to measure an absolute position

without additional hardware is to implement a homing procedure at startup. This procedure takes advantage of the index pulse that every nearly every optical (or simulated optical) encoder uses.

NBV-II Option

Inevitably, design is an iterative process. Anyone who completes a design will lay out either how they could have done better or how they would do it differently for the next incarnation of the product or process. On the other hand, many have pointed out that better is the enemy of good enough and one should make sure to temper one's desire to improve a design until and to the extent that any improvement is worth the cost in time, material and risk. This section is therefore dedicated to laying the groundwork for an improved version of the distributed control system described in this thesis, one that takes advantage of the work and experience already accrued and also one that will provide the headroom to move forward.

As mentioned in the communications section of this paper, many of the higher throughput data buses required either significant processing speed, an ASIC or FPGA to handle communications, or a combination of the two. At the start of this research, there were not accessible solutions that fit into the size and cost requirements required for the Space Systems Lab's dexterous robots. What 2 years ago was confined to a rack unit, is now just approaching the size requirements as prices have fallen and the support, both hardware and software, has increased dramatically. This is especially true of the EtherCAT standard that builds on common 100BASE-TX twisted pair Ethernet. The bus operates at a speed of 100Mbps compared to 1Mbps of the

CAN2.0B bus and is both deterministic and redundant (based on wiring used). While

EtherCAT traffic can coexist with standard TCP and UDP traffic, it operates

independently and has very tight tolerances with data being moved in and out of

frames in a deterministic manner by either an ASIC or FPGA with the appropriate IP

core.

The topography of an EtherCAT network is a daisy chain or ring compared to the

multidrop CAN bus. Instead of having a common physical wire on which each

module communicates, each node connects directly to its neighbors both up and

downstream. This is illustrated in and Figure 42. It should be noted that EtherCAT

also supports more traditional Ethernet tree topologies that may be useful for

applications other than serial link manipulators.
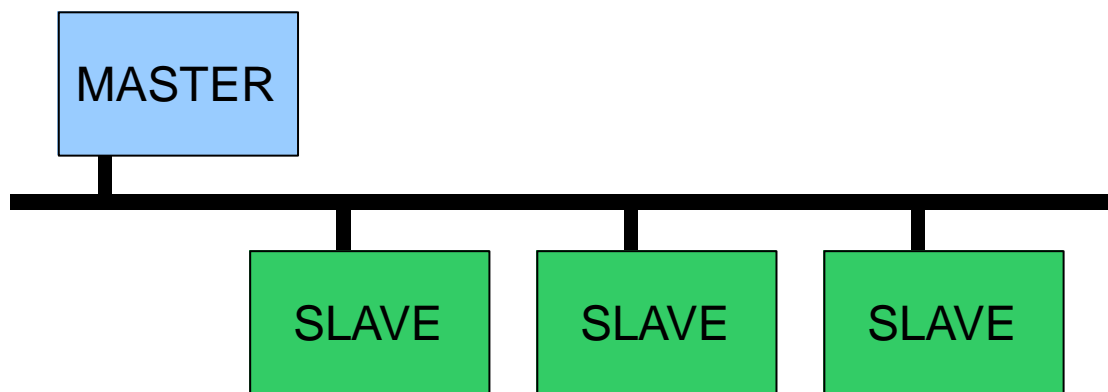
Figure 41 - CAN Wiring
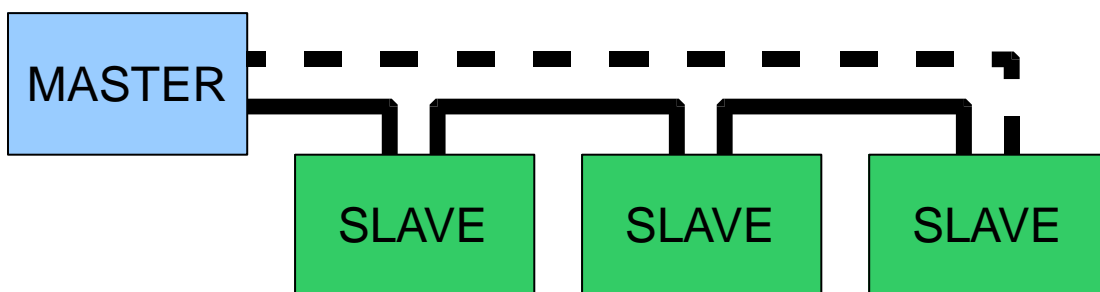
Figure 42 - EtherCAT Wiring

Nominally, switching to a new communications bus would necessitate significant changes in the DMU code to operate using the new protocol. EtherCAT has the ability to operate in the CoE, or CAN over Ethernet, mode where it utilizes the existing CANopen device profiles. In this manner, the driver would change on the DMU, but the rest of the software would operate in the same manner and pass the same data as the existing CANopen architecture.

This is not to say that there would be no changes to the code base per se as the increase in capability would demand similar improvements on the software side to keep pace.   The faster speeds, while still maintaining determinism and timing, would make the lack of real time control from the DMU a major impediment to reaching the full capabilities of the system. This would necessitate a reworking of some of the code to support real time operation and the switch to a patched version of Linux that supports at least soft real time operation.

Appendix A: Quadrature Encoders

The quadrature output is a simple 2 channel Gray code. In a Gray code, only one channel or bit changes at any one time. As only one bit may change at one time, this helps to prevent spurious or corrupted signals from triggering as counts. It should also be noted that the first and final bits in a gray code differ by only 1 bit and thus all Gray codes are also cyclical. As shown in Table 12, the Gray code improves in this manner over binary while still transferring data at the same efficiency. While a 1 bit gray code, or simple on/off pulse would be sufficient for velocity, a robotic manipulator requires direction information as well. The two bit Gray code can be used to determine direction based on the transition. In this manner, a $00 \rightarrow 01$ transition is one count in the forward direction and a $01 \rightarrow 00$ transition is one count in the negative direction.

Table 12 -Gray Code

Number	Binary	Gray
0	00	00
1	01	01
2	10	11
3	11	10

The basic design of the optical encoders, as shown in Figure 43, is an optical disk, usually made of glass, which has the quadrature pulses marked or etched in it. The etching of the second track is set 90° out of phase relative to the pulses of the first track. An additional channel contains only one pulse per revolution, operating as an index to establish the starting position and to ensure that any errors caused by dust or scratching do not add up over many rotations and cause position errors. There is also

97

an LED and photo sensors for each channel. Light from the LED shines through the

disk and is received by the photo detector and translated into a digital signal.
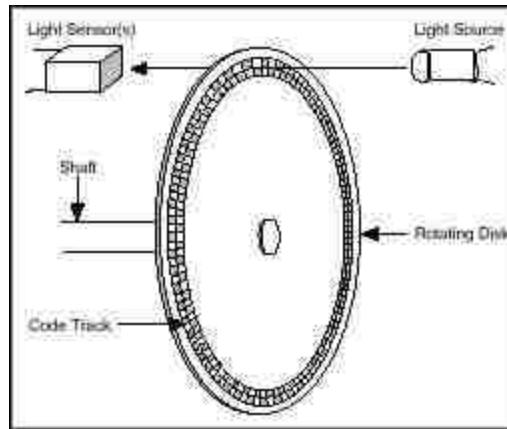
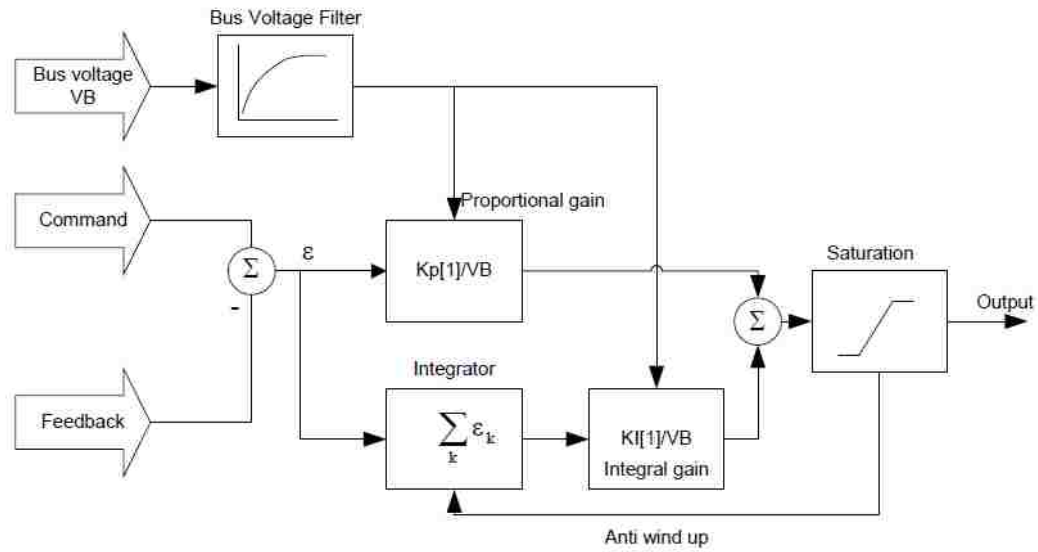Figure 43- Optical Encoder

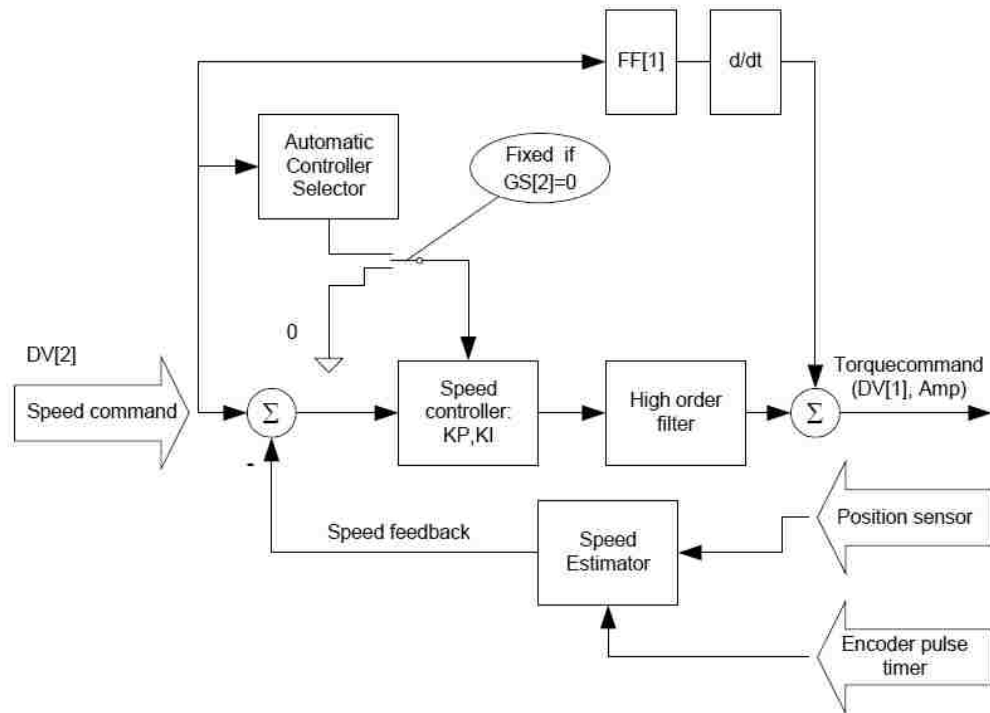# Appendix B: Internal Controllers

Figure 44 – Current Controller(14)

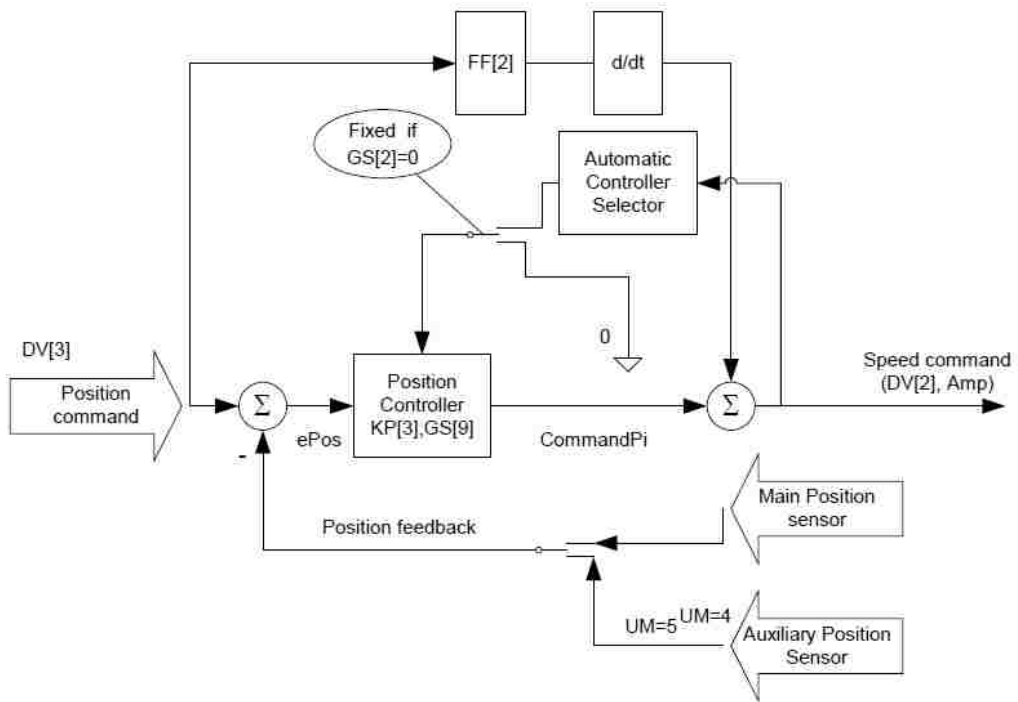**Figure 45 – Velocity control loop for Elmo Whistle(14)**

Figure 46 – Position controller used in Elmo Whistle(14)

Appendix C: Internal Controllers

Glossary

If needed.

Bibliography

1. *AN ARCHITECTURE FOR A NETWORK BASED ROBOT CONTROL SYSTEM.* **Young Shin Kim, Hong Seong Park, Wook Hyun Kwon.** Barcelona , Spain : IEEE, 1999. Emerging Technologies and Factory Automation. pp. 875-880 vol.2.

2. **Harmonic Drive LLC.** Operating Principles. *Harmonic Drive LLC.* [Online] 2010. [Cited: December 1, 2010.] http://www.harmonicdrive.net/reference/operatingprinciples/.

3. **Softing AG.** CAN Bus Arbitration Method. [Online] [Cited: 12 10, 2010.] http://www.softing.com/home/en/industrial-automation/products/can-bus/more-can-bus/communication/bus-arbitration-method.php.

4. **Murphy, Niall.** A-short-trip-on-the-CAN-bus. [Online] August 11, 2003. [Cited: December 6, 2010.] http://www.eetimes.com/discussion/other/4024614/A-short-trip-on-the-CAN-bus.

5. **J. Borenstein, H. R. Everett, and L. Feng.** *"Where am I?" sensors and methods.* s.l. : University of Michigan, 1996. Technical Report.

6. **Texas Instruments Incorporated.** *QUADRUPLE DIFFERENTIAL LINE DRIVER - AM26C31.* 2008. Data Sheet.

7. **Craig, John J.** *Introduction to Robotics: Mechanics and Control (3rd Edition).* s.l. : Prentice Hall, 2004. ISBN-10: 0201543613.

8. **D'Amore, Nicholas John.** *DEVELOPMENT OF A REUSABLE TOP-LEVEL CONTROL ARCHITECTURE FOR A ROBOTIC MANIPULATOR.* College Park : University of Maryland, 2010. Masters Thesis.

9. **American National Standards Institute.** *American National Standard for Industrial Robots and Robot Systems - Point-to-Point and Static Performance Characteristics - Evaluation.* New York : American National Standards Institute, 1999. ANSI/RIA R15.05-1-1990 (R1999).

10. **Danaher Motion.** *RBE(H) Series Motors.* 2003.

11. **Schafer, I, et al.** *SPACE LUBRICATION AND PERFORMANCE OF HARMONIC DRIVE GEARS.* Limburg, Germany : Harmonic Drive AG.

12. **Space Systems Laboratory.** *Ranger Telerobotic Shuttle Experiment: Critical Design Review.* 1999.

13. **Data Device Corporation.** *PW-82520/21N 3-Phase DC Motor Torque Controller.* 2001. Data Sheet.

14. **Elmo Motion Control.** *SimplIQ Software Manual.* 2010.

15. *Development of a Network-based Real-Time Robot Control System over IEEE 1394: Using Open Source Software Platform.* **M. Omar Faruque Sarker, ChangHwan Kim, Jeong-San Cho and Bum-Jae You.** s.l. : IEEE, 2006. International Conference on Mechatronics. pp. 563-568.

16. **Akin.** Photos.