McMaster University
DigitalCommons@McMaster

EE 4BI6 Electrical Engineering Biomedical Capstones

Department of Electrical and Computer Engineering

4-23-2010

Laparoscopic Tool Trajectory Data Acquisition and Analysis

Jiten Mistry
McMaster University

Follow this and additional works at: http://digitalcommons.mcmaster.ca/ee4bi6

Part of the Biomedical Commons

Recommended Citation
Mistry, Jiten, "Laparoscopic Tool Trajectory Data Acquisition and Analysis" (2010). *EE 4BI6 Electrical Engineering Biomedical Capstones.* Paper 47.
http://digitalcommons.mcmaster.ca/ee4bi6/47

This Capstone is brought to you for free and open access by the Department of Electrical and Computer Engineering at DigitalCommons@McMaster. It has been accepted for inclusion in EE 4BI6 Electrical Engineering Biomedical Capstones by an authorized administrator of DigitalCommons@McMaster. For more information, please contact scom@mcmaster.ca.

Laparoscopic Tool Trajectory Data Acquisition and Analysis

by

Jiten Mistry

Electrical and Biomedical Engineering Design Project (4BI6)

Department of Electrical and Computer Engineering
McMaster University
Hamilton, Ontario, Canada

Laparoscopic Tool Trajectory Data Acquisition and Analysis

by

Jiten Mistry

Electrical and Biomedical Engineering
Faculty Advisor: Dr. Patriciu

Electrical and Biomedical Engineering Project Report
submitted in partial fulfillment of the degree of
Bachelor of Engineering

McMaster University
Hamilton, Ontario, Canada
April 23rd , 2010

Copyright © April 2010 by Jiten Mistry

Abstract

Laparoscopic surgery involves complex techniques and many challenges, and thus has a steep learning curve for the novice surgeon. By using tracking methods during training, we can objectively assess the surgeon through trajectory analysis and virtual training. Implementing am inertial measurement unit onto the laparoscopic tool will provide an active way of measuring the position and rotation of the tool over a period of time. A trajectory measured from an IMU is capable of having a very high resolution, and gives us the ability to apply curve matching and analysis algorithms to quantify the performance of the surgeon.

Acknowledgements

I'd like to thank Dr. Patriciu for guiding us through this project. I'd also like to thank my partner Calvin Gan, as well as family and friends, for providing support throughout the year.

Table of Contents

Abstract ..... **ii**

Acknowledgements ..... **iii**

Table of Contents ..... **iv**

List of Tables ..... **v**

List of Figures ..... **vi**

Nomeclature ..... **vii**

1 Introduction **1**

1.1Background 1

1.2 Objective 3

1.3 Methodology 4

1.3 Scope 5

2 Literature Review **6**

2.1 Laparoscopic Surgery Training 7

2.2 Curve Matching Algorithm and Analysis 8

3 Statement of Problem and Methodology of Solution **8**

3.1 Problem Statement 9

3.2 Solution Methodology 10

4 Experimental and Design Procedures **11**

4.1 Analog to Digital Converter 11

4.2 Data Communication 12

5 Results **13**

6 Conclusion and Recommendations **14**

Appendix **15**

References **20**

Vita **21**

List of Tables

Table 1: Comparison of different laparoscopic surgery training solutions…….7

List of Figures

Figure 1: Laparoscopic Surgery……1

Figure 2: System Overview…...……5

Figure 3: Single SPI Block Diagram…..……..10

Figure 4: FTDI FT4232 Mini module **...**……..10

Figure 5: UART Timing Test……...13

Nomenclature

IMU – inertial measurement unit

ADC – analog to digital converter

SPI – serial peripheral interface

UART – Universal Asynchronous Receiver/Transmitter

ICP – Iterative Closest Point

Box trainer – device used to mimic a laparoscopic surgery

Ax, Ay, Az – accelerometer output from IMU

Gx, Gy, Gz – gyroscope output from IMU

Chapter 1

Introduction

1.1 Background

Surgeons are continually looking at ways to improve the effectiveness, safety and efficiency of the procedures they perform on patients on a daily basis. Open surgery, the traditional approach to performing responsive care to patients, had been practiced for centuries. Continual improvements were made to the procedures, but not until the introduction of minimally invasive surgery (MIS) did we see a significant change to the surgery room atmosphere.

MIS is less commonly referred to as laparoscopic surgery. A laparoscope is a specialized endoscope and a crucial component of MIS. It is placed into the body through a small incision, while small tools are placed into the body from other incisions in a similar way. The surgeon views a video screen which displays the image from the laparoscope, to move his surgery tools into the appropriate positions. The use of a camera or a light source to either look into the body for [1] diagnosis or to perform surgery has been practiced for centuries, though with very little frequency when compared to open surgery methods. Open surgery methods have always provided the surgeon a quick and direct way of performing the job, but it has always had significant implications for the patient. He or she would leave the surgery with a large incision on part of their body, which would always turn into a significantly sized scar. Recovery times for the patient have always been significant, since the hospital would not only have to monitor the patients recovery from the surgery, but to make sure the consequences of the open surgery would not affect the patient. This includes blood loss, which in modern practices is not so much of a problem anymore, but we are now seeing significant detrimental effects from surgical gauze which are left in the patient to soak up blood.[2] While open surgery

continues to be performed and provide benefit to many, there are many aspects of it which can be improved upon.

Minimally invasive surgery was not a procedure that developed over night, but rather a collection of methods of performing certain tasks, which were developed over a few centuries, were aggregated to be able to perform a full surgical procedure. [2] In 1585, Dr. Aranzi focused sunlight through a flask of water to be able to see into the nasal cavity of a patient. In 1869, Commander Pantaleoni used a modified cystoscope to cauterize a hemorrhagic uterine growth. [3] The in the 1950's, Dr. Hopkins, developed a rod lens system, which was the foundation of using video during a laparoscopic surgery. In 1977, Kurt Semm demonstrated a technique for endoloop suturing in laparoscopic surgery. With all these contributions and many more, surgeons began to experiment with laparoscopic surgery, and slowly, different open surgery techniques were being performed in a minimally invasive manner. Initially, only surgeries such as cholecystectomy and gynecology related procedures were performed. In the 1980's a rapid increase in the use of laparoscopy was seen, and with the improvement of video technology, by the early 1990's many doctors and hospitals were performing laparoscopic surgery on a regular basis.

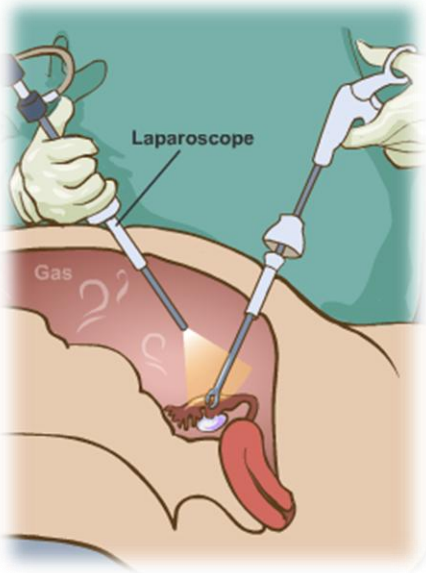

Figure 1: Laparoscopic Surgery

At this point laparoscopic surgery techniques have been developed and practiced for a wide range of treatable problems. The procedure itself has increased patient safety, has reduced their recovery time from 1 to 2 weeks down to as little as one day, and in many cases it has reduced the surgery time, freeing up hospital resources to be used more efficiently.

Laparoscopic surgery took a significant amount of time to develop, for mainly two reasons. The first reason, as described above, was in part due the lack of necessary technology. As the technology developed over time, there was an increase frequency of laparoscopic surgeries. The second limiting factor, which still exists today, is the significant learning curve associated with laparoscopic surgery. MIS, just like any other procedure, requires a significant amount of training and planning, but it adds in another level of complexity because of the nature of the way the tools are inserted in the body with minimal incisions, the surgeon is unable to see in front of him the area of interest. The degrees of freedom are also reduced from 6 to 4, once the tool is placed through the trocar, the opening the body.[4] The tool can now only move in the z direction (in and out), and the 3 rotations on each major axis. This provides a significant disadvantage to surgeons who are transferring from open surgery. In addition, the ability of surgeons to transfer their existing mastered skills in laparoscopic surgery to another surgeon has been proven to be quite difficult when compared to open surgery. Considering the complexity of most laparoscopic surgeries, a method of judging a surgeon on his skills objectively is required, to gain a measurable way of determining his skill level and improvement over time. Current methods of judging are subjective based on the trainer's experience and do not necessarily guarantee a fair assessment. In this report we will discuss solutions of how we can overcome this issue.

1.2 Objective

The ultimate goal of this project is to create a system to be able to quantifiably judge the performance of a surgeon during training. The system will consist of a tracking device, which is connected to the laparoscopic surgery tools when a surgeon is training. During

his training period, the trajectory in which he guides his tools will be recorded in real time and stored onto a PC. We will then be able to compare this recorded trajectory to another trajectory. This could be a trajectory that the same surgeon recorded previously, or compared to a more another surgeon. This will allow us to objectively and quantifiable measure the performance of the surgeon when practicing on a phantom. While similar solutions do exist, the approach we will be taking will be a bit different. The current solutions on the market are being used, but are not perfect as they still require The solution should be easy to use and setup, and should not interfere with the surgeon while he is completing a trial. In addition, the system will integrate with an existing solution that our supervisor, Dr. Patriciu, has already developed. His existing solution uses a stereoscopic camera and infrared LEDs. In the end, the solution will give a significant improvement over current practices in training the surgeon. In addition, the surgeon will walk away from the training with better skills and understanding of his abilities. After a period of time, when the surgeon were to take the training again, he would be able to measure how his skills have improved or decreased since his last training. The availability of this system will also be beneficial as it will remove the requirement of an experienced surgeon being present during the whole training. The computer will already contain the “solution,” and the trainee will be able to view in order to understand what he did wrong. Overall, this solution, will not only provide a more reliable way of training the surgeon, it will decrease training costs and will provide a long term training and assessment platform.

1.3 Methodology

The project will consist of a 6DOF of inertial measurement unit (IMU).This unit will be attached to the laparoscopic tool. The IMU is the key component of how we will acquire the trajectory of the laparoscopic tool. The unit itself will consist of an accelerometer for each direction (3), as well a gyroscope for each axis (3). These 6 voltage signals will be fed into an analog to digital converter. The ADC will be a part of the microcontroller, and will preferably be over 8 bits. It will use the internal clock to ideally perform the conversion for all six values simultaneously. The goal is to achieve several hundred

samples per second. Once the values are determined, they will be send in a single packet over the SPI communication protocol to the host computer. SPI is chosen for it's potential speed of upto 10 Mhz. Once the values are read through the communication port on the host computer, the values will be fed into an algorithm. The algorithm will take in the 6 voltage values from the IMU, and calculate x,y,z positional data, as well as the rotation angles on each axis. The host computer will store the values locally, while storing the data sequentially with the positional data from the stereoscopic camera. Below is a diagram outlining the overall system.

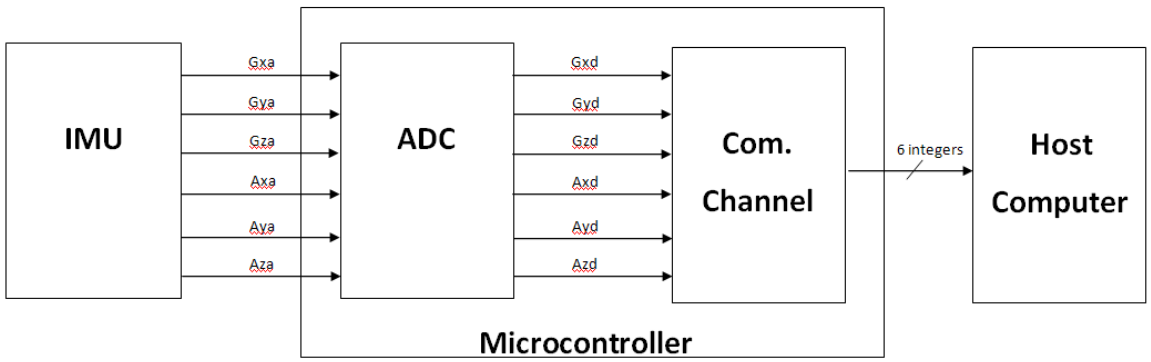

Figure 2: System Overview

Once the complete trajectory was stored, the goal was to display the trajectory in a user friendly manner, and apply a curve matching algorithm to compare it with other trajectories. Using the values obtained from the curve matching algorithm, a quantifiable result value would be given to the user, to sum up the differences between the recorded trajectory and existing stored trajectory.

1.4 Scope

The overall project, as described above, is a tracking device for laparoscopic surgery tools for training purposes. This solution, if completed and accurate, will have a use in medical schools across the world. It'll be a tool required to train surgeons' laparoscopic techniques. It will of course require input from an experienced surgeon in order to record

their trajectories for evaluation. The application of this project is very specialized, and currently has no application outside the medical schooling realm.

In order for the solution to be valuable, it'll require a high level of accuracy, in terms of the reading from the IMU. The IMU unit itself should not interfere with the operation of the laparoscopic tool. The trainee should have the impression that nothing has been added to the tool. The goal is to continue training the surgeon as though he is an operating room.

This project will be split up into two sections with my group member, Calvin Gan. He will be concerned with aspects related to the IMU and the algorithm which will be used to calculate relevant values from it. I will be handling the operation of the ADC, communication over SPI, and any data processing after the IMU algorithm which may need be completed.

Chapter 2

Literature Review

The research completed for this project can be divided into 3 sections: Laparoscopic surgery training research papers, USB and communication protocol, and curve matching. The 3 sections will be described separately below. The research completed in each of these sections allowed us to properly select the devices we would be using and justifying our methods. USB and communication protocols will not be discussed in this section as the research associated with it was conducted through non academic sources.

2.1 Laparoscopic surgery training

A part of the motivation behind this project was inspired by the numerous papers written on laparoscopic surgery training. M. K. Chamarra et al. [4] discussed in their paper the various solutions currently available for laparoscopic surgery skills training. While the list was not exhaustive, it did provide a detailed comparison of several leading products. At the time, the tracking devices used for laparoscopic surgery skills training, were based on mechanical, optical, acoustic or electromagnetic technologies. Optical tracking systems would use a system with cameras and a set of light emitting diodes (LED) to measure the position and rotation of the object. Acoustic tracking devices use ultrasonic sound waves. Electromagnetics technologies which are similar to what we are working with, use small wire coils oriented orthogonally in each of the 3 axes, within a low frequency electromagnetic field. Below is a table that compares the training systems:

Table I. Overview of tracking systems*

Name	System	Mechanics	DOF	Lap. instr.	Environment	Portability	Feedback	Accuracy	Availability
ProMIS	P	–	3	+	box, VR	–	+		+
UMS	P	–	4	+	box, VR, OR	–	+		+
UWPS	P	–	1	+	OR	–	+	<40 μm	–
LSW	A	gimbal	4	–	VR	–	+		+
VLI	A	gimbal	4	–	VR	+	–		+
LIE	A	joystick	4	–	VR	+	+		+
CELTS	A	gimbal	4	+	box, VR	+	+		–
ADEPT	A	gimbal	4	+	box		+	± 0.5 mm	+
Simendo	A	gimbal	4	–	VR	+	–		+
BlueDRAGON	A	bar passive	4	+	box, OR	–	+		
Patriot	A	–	4	+	box, VR	+	+	2.54 mm, 0.75°	+
Xitact ITP	A	PantoScope/ LinRot	4	–	VR	+	–		+
Xitact IHP	A	PantoScope/ LinRot	4	–	VR	+	+		+
IOMaster5D	A	cable drive/ pantograph	4	–	VR	+	+		–
IOMaster7D	A	cable drive/ pantograph	7	–	VR	–	+		–
TrEndo	A	gimbal	4	+	box, VR	+	+	>95 %	–

* DOF – degrees of freedom; UMS – ultrasound measurement system; UWPS – ultrasound wireless positioning system; LSW – laparoscopic surgical workstation; VLI – virtual laparoscopic interface; LIE - laparoscopic impulse engine; P – passive; A – active; OR – operating room; VR – virtual reality trainer; box – box trainer.

Table 1: Comparison of different laparoscopic surgery training solutions[1]

The systems were split into two groups, passive and active. Active systems, the sensors which measure the actual movement are attached to the tool, such as the IMU we will be using, and passive systems localize the tool using markers that have been placed on the

instrument. The passive systems had the disadvantage of needing a line of sight in order to track the tool, which made active in one sense superior. Active, though more accurate, generally requires the tool to be physical connected either to a stand, or by wire. This restricts motion in some ways. Overall the emphasis was on having systems in which acquire the most accurate trajectory, while simulating the surgery using a box trainer as closely as possible. Very few of the solutions evaluated used electromagnetics, and that is why that approach was explored in this project.

2.2 Curve matching Algorithm and Application

Curve matching has been a well-studied topic for several years now, and researchers have showed interest in it so that it could be used to quantify laparoscopic surgery training. In Leong et. Al [5], the use of Hidden Markov models to train a system with trajectories into different skill level groups. The output was the similarity of a given subject's trajectory compared to the groups. It was successful in ranking the quality of each subject's trajectory, based on a model. Hidden Markov models are an advanced way of doing curve matching. A entry level way to conduct curve matching would be to use the popular iterative closest point (ICP) method. An improved version of the algorithm, named TrICP [6], is discussed and evaluated by Chetverikov et al.

Chapter 3

Statement of Problem and Methodology of Solution

This chapter will outline the technical objectives of the project, and provide the background information which will be required for putting together the completed solution. The tools and resources required to develop the completed project will also be discussed in this chapter.

3.1 Problem Statement

In order to derive any value out of the use of the IMU, a reliable and effective way of communicating the data to the host computer would be required. Numerous protocols and methods are possible, including serial or parallel communication, wired or unwired connection, and encrypted or unencrypted. For this specific project, a solution that would be able to communicate a packet of 6 bytes at 1000 packets/ a second is what we would be looking for, after the overhead of hardware components and software are accounted for. Numerous factors could go into efficiency of the microcontroller, mainly the hardware selection and code implementation. The main requirement of the hardware aspect of the microcontroller will be the ability to read in 6 analog values simultaneously. This will be a key requirement, since we are required to read 6 analog values at a time from the IMU. The microcontroller will also require built in support for communication with a host computer. With the communication built into the microcontroller, it will require less clock cycles, and this will allow the solution to run faster overall. The microcontroller will also require the ability to read 6 analog values at a time. This will require a versatile and fast ADC. Ideally a 12 bit ADC will be utilized to achieve a decent accuracy when measuring analog values.

For the curve matching portion of the project, an algorithm will be chosen on it's ability to quickly align two curves, and be able to extract parameters from the solution. These parameters will be crucial as they allow us to quantify the similarities and differences of the two curves we are analyzing. What these parameters are and how they will be used will be discussed in chapter 4.

3.2 Solution Methodology

Our initial solution involved using Serial Peripheral Interface (SPI) as our communication protocol standard. It was chosen for its speed capabilities, which would leave us a room for error, and its two channel communication packet handling. In addition it is synchronous, providing more reliability. The two way communication would also help reduce the errors, and if time permitted, allow us to implement extra code that would fully utilize this feature to ensure our data was perfect, in terms of time intervals, as well as when combining the data with the positional and rotational data coming from the stereoscopic camera. Below is a simplified diagram of how SPI works:

Figure 3: Single SPI Block Diagram

SPI requires 4 connections between the master device and the slave device. The slave select pin on the master device selects which slave device it will be communicating with. In the event there is only a single slave device, the $\overline{SS}$ pin on the master can be ignored and the $\overline{SS}$ can be set to ground. The SPI Master provides the clock to the slave device so they are both synchronized. The remaining two pins on each device are the input and output pins.

For this aspect, we selected the FT4232H Hi-speed USB chip, manufactured by Future Technology Devices International Ltd. The chip came with as an evaluation product on a mini breakout board. It is pictured below:

Figure 4: FTDI FT4232 Mini module[1]

The evaluation kit came with access to all pins of the chip. The device is capable of being USB-powered, and requires 4 pins to connect to an SPI module. It is only capable of running as an SPI Master device, which could pose a problem for a selection of the microcontroller. The code for this section will be written in C++ using the Visual Studio environment.

The PIC24 [2] was initially identified as a microcontroller to use to test our embedded code and extract initial results from the IMU. The solution id used for the Computer Engineering 2DP4. It comes integrated onto a PCB with a USB serial connection and access to the majority of the pins. The device does not include SPI, but it does support other communication protocols. The code for this part of project will be developed using MPLab environment, and will be loaded onto PIC24 using the bully bootloader[3] serial programmer.

For the trajectory analysis aspect of the project, the iterative closest loop (ICP) algorithm was identified as an easily approachable method to conduct curve matching. Details on ICP are in the literature review chapter. The code will be written in C++ using the Visual Studio environment.

Chapter 4

Experimental and Design Procedures

4.1 Analog to Digital Converter

A considerable amount of time was spent learning how to configure and use the ADC on the PIC24. The microcontroller's internal voltage (3.5V to 0) was used as the reference

[1] www.ftdi.com
[2] www.microchip.com
[3] www.reesemicro.com

when taking in our analog inputs. The IMU unit, used a 3.5V reference also, so there was no circuitry placed between the IMU and the ADC, and there would theoretically be no saturation issue. A major drawback of the ADC on the given PIC was that it could only sample 4 analog channels at a time. This required us to reconfigure the PIC for every sample set of 6 signals. Initially, the PIC would be set up to read 4 channels simultaneously. This would read pins AN12, AN3, AN4, and AN5, which were ax, ay, and az, and gx, respectively. The PIC would then be reconfigured to read 2 channels simultaneously. This would read pins AN9 and AN10, which were gy, gz respectively. After this was complete, the cycle would repeat. The ADC was required to be configured in 10 bit mode, if simultaneous sampling was enabled. In configuring the ADC, the parameters were set to maximize the amount of samples which could be read over time. The time between the start of the sampling, and the start of conversion, was set to 2 , which is the minimum time allowable for the ADC in 10 bit simultaneous mode. While it wasn't ideal to set these values since it didn't leave time to average the ADC samples over a set, but it was required to increase the sample rate. While the ADC created 10 bit integer values, they were stored as 16 bit integers.

4.2 Data Communication

The USB chip incorporated into the PIC24 module we used did not support SPI, so a temporary measure was made to use UART to initially test the ADC function. The USB to serial chip used on this module, was also manufactured by FTDI, the make of the SPI capable USB chip selected earlier on. It used the same dll library to allow the USB device to act as a virtual com port on the host computer. This save a considerable amount of time in setting up the USB connection. Once the communication port was established, the .NET SerialPort() Class was used in C++ to obtain our values from the com port. (Please see appendix for the code.) Due to the nature of the library and the values being sent, the values coming through the com port were read one byte at a time. This was unfortunate as it took more time to read then initially expected. After a control signal was sent from the PC to start sending values, bytes were read in sequentially. Every 2 bytes were saved

as one integer, to reform the 16bit integer sent over UART. Once each set of six values were saved, it would begin to read the next six values. It would continue until the communication port was closed.

Chapter 5

Results

Unfortunately, due to problems with the hardware that was selected, significant results were not obtained, and this hindered our ability to move further into the project. The primary aspect of the data communication portion, the USB controller, was unable to achieve the desired sampling rate of the IMU data. This was a significant hindrance, since without a significant throughput of data from the IMU, there was no motivation to continue on to the next step where we could begin using the data. The PIC controller, after a considerable amount of time of code manipulation and tinkering, was unable to achieve a rate of 35 samples / sec, which was well below our goal of achieving 1000 samples /sec. An image of the test is shown below. The timing was calculated using the C++ clock() library.:

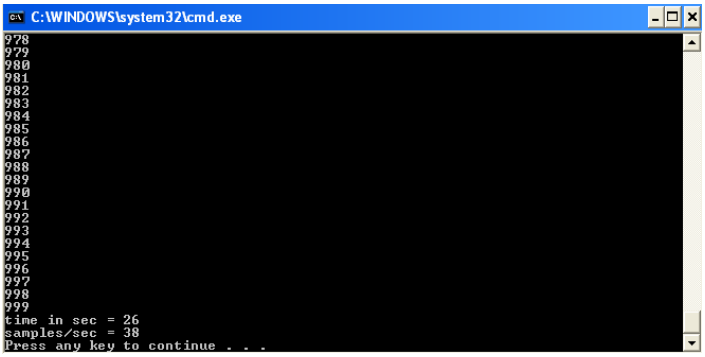

Figure 5: UART Timing Test

An alternative microcontroller with a higher quality ADC was regrettably not pursued. With the low sample rate we achieved, application of the ICP was not carried out, nor was the use of SPI.

Chapter 6

Conclusion and Recommendations

Although conclusive results were not obtained, it was clear that the IMU would eventually provide significantly improved trajectory measurements over passive and other active tracking methods currently employed. The use of the ADC on the PIC24 was a major drawback, considering its inability to read 6 analog channel simultaneously, and should have discontinued use in favor of a ADC capable of reading 6 channels simultaneously. A major recommendation for the future, would be to use a high performance ADC for this project. This will aid in speed issues, and more importantly, accuracy. The AD7656 by Analog devices is a surface mount chip with the ability to sample 6 channels simultaneously[4]. With the investment into a ADC, the use of a microcontroller high a higher clock rate may be a something to also consider.

[4] http://www.analog.com/static/imported-files/data_sheets/AD7656-1_7657-1_7658-1.pdf

Appendix A: Supplementary Figures

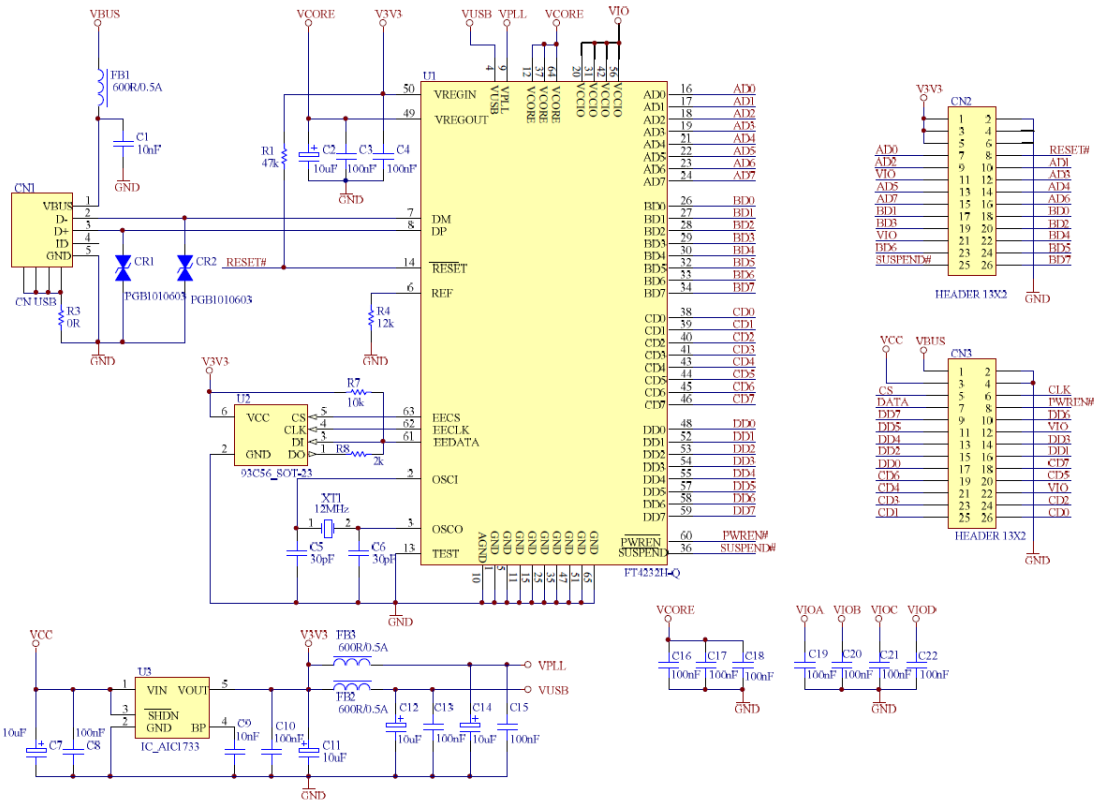

Figure A.1: Schematic of FT4232 Mini Module [5]

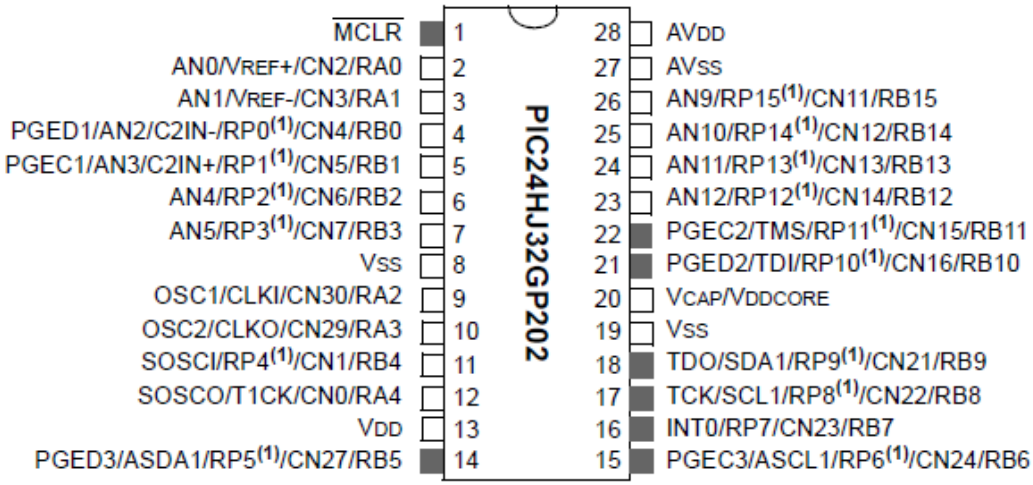

Figure A.2: PIC24HJ32GP202 Pin out [6]

[5] www.ftdi.com

Appendix B: PIC Embedded Code

B.1: PIC Firmware Code (Developed using MPLab)

```
int main (void) {
  uint8   u8_i;
  uint16  u16_ax;
  uint16  u16_ay;
  uint16  u16_az;
  uint16  u16_gx;
  uint16  u16_gy;
  uint16  u16_gz;

  uint32  u32_ticks;
  //float   f_pot;

  //configBasic(HELLO_MSG);

  // make RA0/AN0/VREF+ a digital input to kill the pullup and
  // set the TRISA bit, then make it ANALOG so the ADC will work
  CONFIG_AN3_AS_ANALOG();
  CONFIG_AN4_AS_ANALOG();
  CONFIG_AN5_AS_ANALOG();
  CONFIG_AN12_AS_ANALOG();
  CONFIG_AN9_AS_ANALOG();
  CONFIG_AN10_AS_ANALOG();
  CONFIG_AN11_AS_ANALOG();

  CONFIG_LED2();

  u8_gotData = 0;
  // configure T2/T3 as 32-bit timer to trigger every 1/64 second
  T3CONbits.TON = 0;
  T2CONbits.TON = 0;
  T2CON = T2_32BIT_MODE_ON | T2_PS_1_1 | T2_SOURCE_INT;
  TMR3 = 0;
  TMR2 = 0;
  u32_ticks = usToU32Ticks(15625, getTimerPrescale(T2CONbits)) - 1;
// # of ticks for 1/64 seconds
  PR3 = u32_ticks>>16;
  PR2 = u32_ticks & 0xFFFF;
  T2CONbits.TON = 1;
  int i = 0;
      int time = 0;
  clock_t start = clock();
  clock_t end;



if(inChar() == '1'){
```

[6] www.microchip.com

```
  //while (1) {
  for(i = 0; i < 1000; i++){

      configADC1_Simul4ChanIrq(12, ADC_CH123_POS_SAMPLEA_AN3AN4AN5,
ADC_CONV_CLK_12Tcy );
    SET_SAMP_BIT_ADC1();
      outUint16Decimal(i);
      //outChar('\n');
    while (!u8_gotData) {
      doHeartbeat();
    }
    u8_gotData = 0;
    //for ( u8_i=0; u8_i<4; u8_i++) {
    u16_ax = au16_sum[0];
      u16_ay = au16_sum[1];
      u16_az = au16_sum[2];
      u16_gx = au16_sum[3];
    //  f_pot = (3.3 / 1023 / 64 ) * u16_pot;
     //printf("ax %d ay %d az %d gx %d  ",  u16_ax, u16_ay, u16_az,
u16_gx);
   outUint16Decimal(u16_ax);
   outUint16Decimal(u16_ay);
   outUint16Decimal(u16_az);
   outUint16Decimal(u16_gx);
   // } //end for()

   //endof while()

  configADC1_Simul4ChanIrq(12, ADC_CH123_NEG_SAMPLEA_AN9AN10AN11,
ADC_CONV_CLK_12Tcy );
  SET_SAMP_BIT_ADC1();

   while (!u8_gotData) {
      doHeartbeat();
    }
    u8_gotData = 0;
    //for ( u8_i=0; u8_i<4; u8_i++) {
    u16_gy = au16_sum[1];
        u16_gz = au16_sum[2];

    //printf("gy %d gz %d ",  u16_gy, u16_gz);
    outUint16Decimal(u16_gy);
    outUint16Decimal(u16_gz);
      outChar('\n');
   // } //end for()
   //printf("\n");

 } //endof while()
}

end = clock();
time = (end-start)/CLOCKS_PER_SEC;
printf("start = %d\n", start);
printf("end = %d\n", end);
printf("time = %d", time);
} // endof main()
```

B.2: C++ Serial Port Input Code

```
 // This is the main DLL file.

#include "stdafx.h"

#include "serial_port.h"
#include "windows.h"
#include "stdio.h"
#include <iostream>
#include <time.h>
#include <string>
//#include "system.dll"
using namespace System;
using namespace System::IO::Ports;
using namespace std;


int main(array<System::String ^> ^args)
{
	typedef char uint8;
	uint8 f1;
	uint8 f2;
	uint8 f3;
	uint8 f4;
	uint8 f5;
	uint8 f6;
	uint8 f7;//the new line character gets stored here. (if there is
one)

	char temp[7];//this is where we store the incoming number.

	int val = 0;
	int vals[6][1000];//this is the positional and rotational
information.
	int valCount;
	int Valtemp;
	string sTemp = "";
	//SerialPort^ serialPort = gcnew
SerialPort(L"COM14",57600,Parity::None,6,StopBits::One);
	SerialPort^ serialPort = gcnew SerialPort("COM14",57600);
	serialPort->Open();


	valCount = 0;
	//serialPort->Write("Send");
	ead in 6 ints at a time, + 1 for the new line character currently
being dumped out.
	serialPort->Write("1");//this means send Data
	clock_t start1 = clock();
	clock_t end1;
		for(int setCount = 0; setCount < 1000; setCount++){
		//r	cout<<setCount<<"\n";
```

```
        for(int valCount = 0;valCount < 6;valCount++){

              switch(valCount){
              case 0:
                    cout<<"ax ";
                    break;
              case 1:
                    cout<<"ay ";
                    break;
              case 2:
                    cout<<"az ";
                    break;
              case 3:
                    cout<<"gx ";
                    break;
              case 4:
                    cout<<"gy ";
                    break;
              case 5:
                    cout<<"gz ";
                    break;
              }
                     f1 = serialPort->ReadByte();
                    //printf("%c", f1);
                    temp[0] = f1;
                    f2 = serialPort->ReadByte();
                    temp[1] = f2;
                    //printf("%c", f2);
                    f3 = serialPort->ReadByte();
                    temp[2] = f3;
                    //printf("%c", f3);
                    f4 = serialPort->ReadByte();
                    temp[3] = f4;
                    //printf("%c", f4);
                    f5 = serialPort->ReadByte();
                    temp[4] = f5;
                    //printf("%c", f5);
                    temp[5] = '\0';
                    vals[valCount][setCount] = atoi(temp);
                    cout<<"\n"<<temp<<"\n";
                }
        }
        end1 = clock();
        int time = (end1-start1)/CLOCKS_PER_SEC;
        cout<<"time in sec = "<<time<<"\n";
        cout<<"samples/sec = "<<1000/time<<"\n";
      serialPort->Close();

      //return 0;
}
```

References

[1] “Laparoscopic History”, laparaoscopyhospital.com, [Online]. Available https://www.laparoscopyhospital.com/history_of_laparoscopy.htm

[2] “Surgical Objects Accidentally Left Inside About 1500 Patients in US Each Year”, Dec, 2007. [Online] Available http://www.sciencedaily.com/releases/2007/12/071208171847.htm

[3] “Laparoscopic Equipment Details”, laparaoscopyhospital.com, [Online]. Available http://www.laparoscopyhospital.com/laparoscopic_equipments_detail_2.htm

[4] M. K. Chmarra, C.A. Grimbergen, and J. Dankelman, “Systems for tracking minimally invasive surgical Instruments”, Minimally Invasive Therapy and Allied Technologies, 2007, 16:6,328 — 340

[5] J.J. Leong, M. Nicolaou, L. Atallah, “HMM Assessment of Quality of Movement Trajectory in Laparoscopic Surgery”, Computer Aided Surgery, 2007, 12:5, 335 -346

[6]D. Chetverikov, D. Svirko, D. Stepanov, “The Trimmed Iterative Closest Point Algorithm”, Pattern Recognition, 2002, 3, 30545

VITA

NAME : Jiten Mistry

PLACE OF BIRTH: Ashton Under-Lyne, UK

YEAR OF BIRTH: 1987

SECONDARY EDUCATION: Pickering High School (2001 – 2005)

POST SECONDARY EDUCATION: McMaster University (2005 – 2010)