PosiTrack School of Engineering Science • Burnaby • BC • V5A 1S6

April 20th, 2010

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, BC
V5A 1S6

Re: ENSC 440 Post Mortem for a System to Track Athlete Performance

Dear Dr. Rawicz:

The attached document outlines the PosiTrack team's efforts to create a diagnostic tool that tracks an athlete's performance. This product is called the PosiTracker and it can be used by large sports teams as both a broadcasting tool and a training tool.

This document outlines the current state of the proof-of-concept device, future plans for the device, deviation from originally planned design specifications, test results, and deviations from the budget and schedule. An individual review of the development process by each team member is also included.

PosiTrack Systems consists of four hard-working and motivated team members: Andreea Hrehorciuc, Jaime Valdes, Jeff Anderson, and Ryan Lynne. Should you have any questions or comments about the post mortem document please feel free to contact me by phone or email.

Regards,

Ryan Lynne

Ryan Lynne
1.778.840.9111
posi.track.systems@gmail.com
PosiTrack Systems

Enclosure: *Post Mortem for a System to Track Athlete Performance*

Post Mortem for a

System to Track Athlete Performance

PosiTrack Systems

A highly precise diagnostic tool to analyze all aspects of an athlete's performance

Project Team: Andreea Hrehorciuc
Jamie Valdes
Jeff Anderson
Ryan Lynne

Contact Person: Ryan Lynne
posi.track.systems@gmail.com

Submitted to: Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University

Issued date: April 20th, 2010

Revision: 8

Table of Contents:

List of Tables and Figures:

Glossary and Acronyms:

A/D	Analog to Digital
Actions	Athlete's movements and biological information such as heart rate
Anchors	A signal source with a known location
AOA	Angle of Arrival
AP	Access Point (see Anchors)
COM	Computer on Module
CPU	Central Processing Unit
Dead Reckoning	Using current direction and speed to determine future location

DLL	Dynamic Linked Library
DOF	Degrees of Freedom
DT	Diagnostic Tool
EM	Electro-Magnetic
GPS	Global Positioning System
GUI	Graphical User Interface
HDMI	High Definition Multimedia Interface
I2C	Inter-Integrated Circuit
ID	Identification
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
IP	Internet Protocol
OS	Operating System
RAM	Random Access Memory
RSS	Received Signal Strength
RSSI	Received Signal Strength Index
SD	Secure Digital
Sensor Fusion	To combine multiple measurements to increase measurement accuracy
SSH	Secure Shell
SPI	Serial Peripheral Interface
TCP	Transmission Control Protocol
TOA	Time of Arrival
UART	Universal Asynchronous Receive Transmitter
USB	Universal Serial Bus
VB	Visual Basic
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network

1. Introduction

The PosiTracker is a system that tracks the relevant statistics of athletes during game play or training. The system is targeted towards large sports teams and can be used for physical training, coaching or broadcasting. This system contains three main sub-components: the Diagnostic Tool, the Access Points, and the Graphical User Interface. The Diagnostic Tool is worn by the athletes in order to gather information such as location, acceleration, heart rate, and speed. Using the Access Points, the data is sent wirelessly back to a computer station running the Graphical User Interface. The Diagnostic Tool is able to determine it's location by measuring the signal strength of the various Access Points.

Over the course of the past 4 months, this system has been in development by a team of four Engineering students: Andreea Hrehorciuc, Jamie Valdes, Jeff Anderson, and Ryan Lynne. The following document outlines the current state of the proof-of-concept system, future plans for the system, deviation from the design specifications, test results, and deviations from both the budget and schedule. The teams group dynamic will also be summarized. To conclude, an individual review of the development process by each team member is included.

2. Current State of the Project

2.1. System Overview

The PosiTracker System is briefly outlined below in Figure 1.

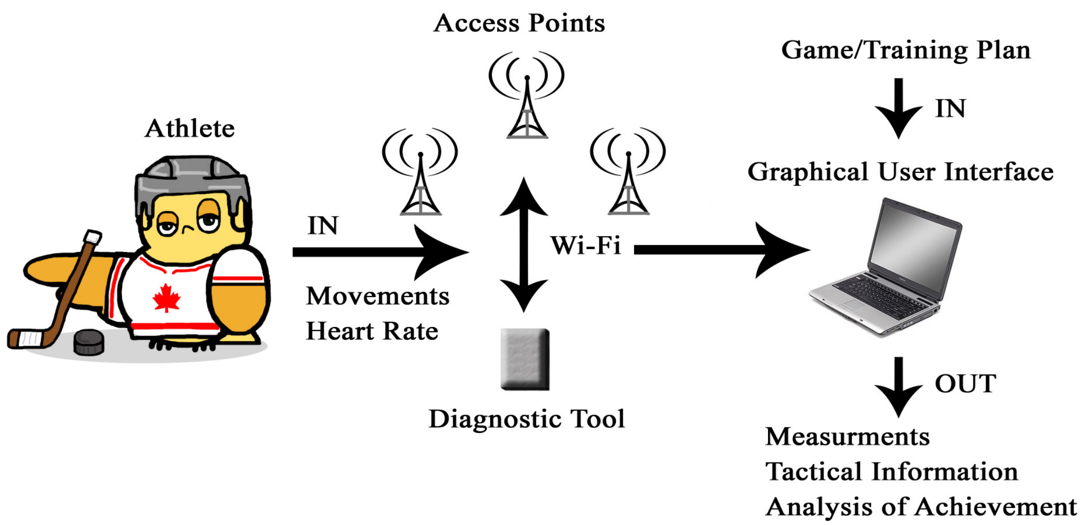

Figure 1: Design overview of the PosiTrack system.

Our system to track an athlete's performance consists of a Diagnostic Tool (DT), Access Points (APs) and a Graphical User Interface (GUI). The DT is a small electronic device that is placed on the athlete, it performs the desired measurements. APs are wireless routers that conform to the IEEE 802.11 Wi-Fi protocol. The DT measures the Wi-Fi signal strength from APs placed around the playing area to determine position at a given time. The DT also measures the linear and rotational acceleration of the athlete. The raw data generated by the DT will be sent via a WLAN to a host computer. This host computer will be running the GUI software and receive the data in real time. The GUI takes the acquired data and generates useful information such as location, speed and acceleration.

2.2. Diagnostic Tool

The current state of the diagnostic tool, or the athlete's module, is completed and meets all of the major and vital functional specifications. Some minor changes were made to the functionality such as processing of the data was moved from the module to the GUI. Also, the measurement interval of the RSSI data was only 130ms when we hoped to achieve 100ms. However, with further development time this could have been achieved.

Below is a short summary of the key specifications that have been met:

- Can measure RSSI from multiple AP's at 130ms intervals (req was 100ms)
- Can measure 3-axis acceleration at up to 10ms intervals
- Can measure 3-axis angular rotation at up to 10ms intervals
- Weighs 105 grams (requirement was 150g)
- Dimensions 11 x 5 x 2cm (requirement was 10 x 5 x 3cm)
- Automatically works when turned on
- Battery life of 2.5hours
- Recharge time of 5 hours
- Maximum range of 300ft

Figure 2 below shows a picture of the completely assembled device as described in the functional/design specifications.

PosiTracker Prototype

Dimensions: 110 x 51 x 21 mm

Weight: 105 grams

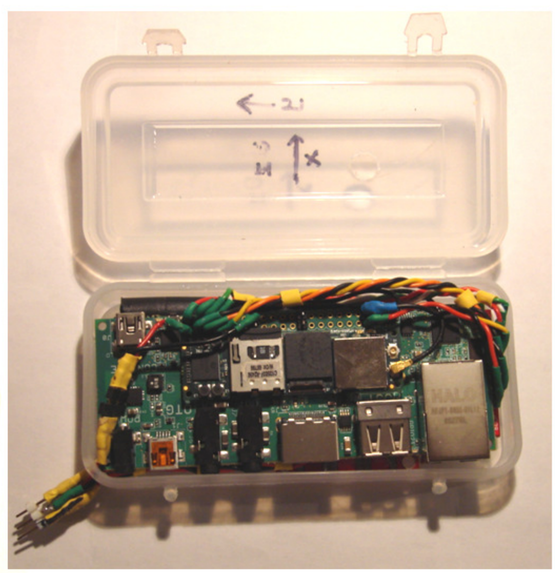

Figure 2: The assembled module for the PosiTrack system.

2.3. Software

The software is comprised of three parts, the GUI itself, the DT code and the DLL. The software schematic is shown in Figure 3. Currently the state of the software is as follows:

- DT:
  - Able to receive signal strength and inertial measurements at desired sampling rate
  - Able to receive and send commands from the DLL
  - Packaging and sending data packets is functional

- GUI:
  - Able to facilitate program start up and shut down procedures
  - Currently displays linear acceleration and rotational velocity

- DLL:
  - All DLL function calls have been implemented and can be called by the GUI
  - Message passing between the DLL and the DT works as intended

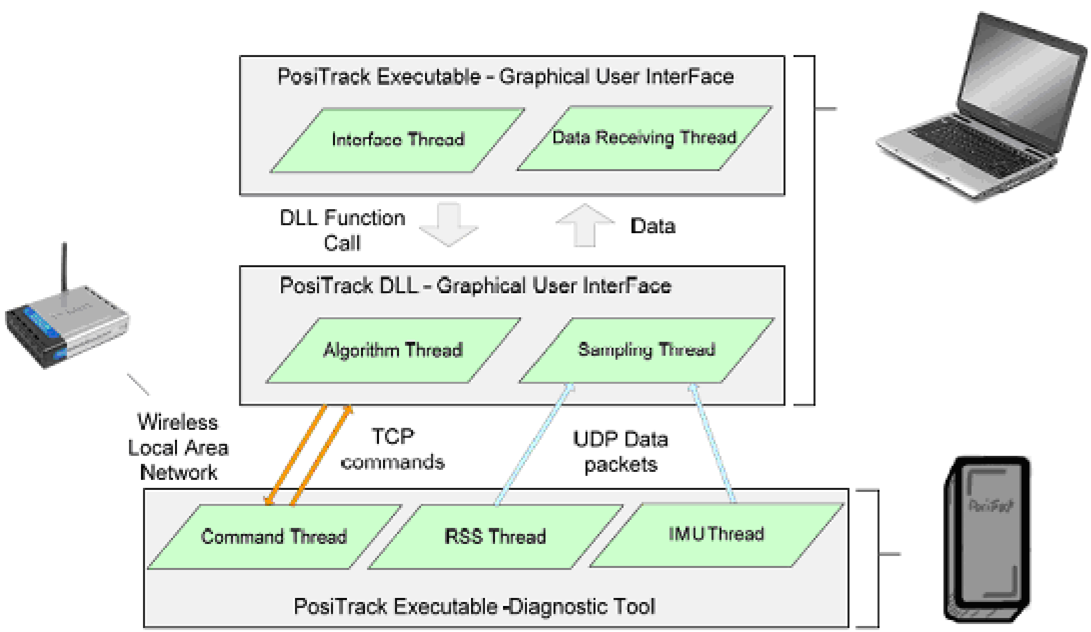

Figure 3: Software diagram for the PosiTrack system.

2.4. Algorithms

The multiple algorithms to derive the player's location, and speed were tested but not fully implemented to software. The following outlines the current status of the algorithm development:

- Dead Reckoning:
    - This algorithm of double integration to find the velocity and location has been implemented and tested through the saved data on to excel spread sheets.

- RSS localization:
    - All propagation models have been developed for the wireless APs
        - The geometric algorithm that uses the propagation models to triangulate position through averaging AP distance radii is implemented in excel
        - The probability distribution algorithm that uses the propagation models to triangulate position through using AP Gaussian probability radii is implemented into the DLL software
    - The WAP mapping technique has been competed for a 5x12m indoor space
        - This technique has been tested and implemented with data in excel

- Probability Filters:
    - Future work is needed to combine the dead reckoning algorithm and the RSS localization into a highest probability localization filter

2.5. Deviations

In terms of functionality, we have achieved a base level of operation as planned. Some features which were outlined in the design and functional specifications were not able to be realized due to time constraints. These deviations are described in this section.

Our team deviated from the functional specification by creating a centralized not a distributive system. Requirements R36-37, R39 imply that the algorithms should be calculated on the DT however we changed the system to run the algorithms in the DLL to increase ease of implementation.

The functional specs R48-50 were not met due to time constraints. These functionalities of having a an external on/off button and a USB charging port were unnecessary and time consuming functions for a proof of concept model which is why they were ignored.

The requirement R56 states the DT module is not affected by other Wi-Fi traffic / noise. This functionality is accomplished in terms of message passing. However for received signal strength localization the algorithm will always be affected by noise from the Wi-Fi signals.

We changed the decision to run the GUI on Linux to run on Windows. This change conflicts with functional requirement 73 but satisfies 74.

A deviation from the design that occurred is using UDP message passing to send the data in smaller and more frequent packets rather than using larger less frequent TCP packets. The reason for this change is we were having large retransmission lag delays when sending TCP packets. This change to UDP packets solved the problem.

The design specification did not consider the multiple threads needed for parallel processing the measured RSS and acceleration data.

3. Results

This section outlines the results we achieved using the device and the developed algorithms. Below range tests and Wi-Fi propagation models are discussed. This information is used in one and two dimensional localization test. Dead reckon is also discussed in this section.

One of the first tests that we performed was to find the maximum range of the DT. To do this we placed one of the wireless access points in the street and then measured the wireless signal strength every 25 feet. This setup can be seen below in Figure 4. The results of this test are shown in Figure 5 where the test was performed three times and the maximum distance was found to be 300ft. This test was to ensure that the DT could maintain a wireless link in large open areas such as a hockey rink.

Figure 4: Test setup to find the WiFi range of DT.

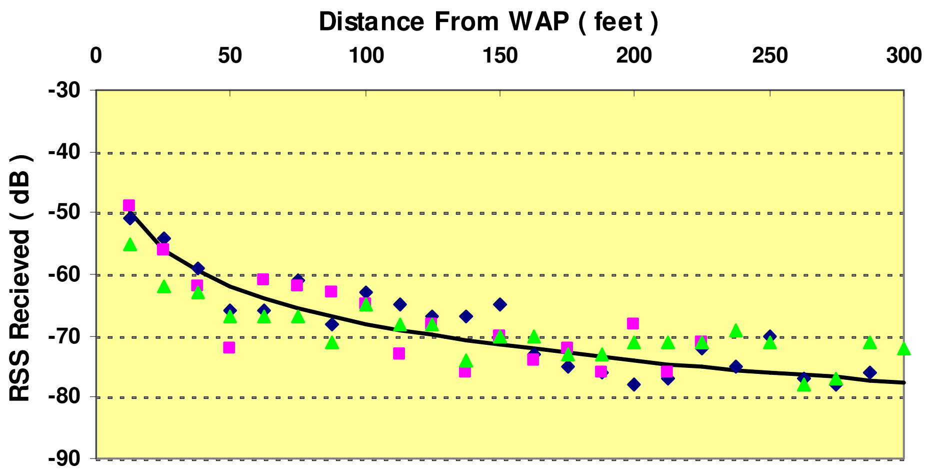

Figure 5: Results showing the maximum range of the Gumstix to be 300ft.

The next test that we performed was to more carefully observe the attenuation of the signal strength from multiple wireless access points. In this test we placed two access points 8 meters apart and measure the signal strength at every 0.25 meters in-between. This test setup can be seen below in Figure 6.

Figure 6: Test setup for signal strength attenuation, Gumstix middle, access points on either end.

The data from the tests conducted in Figure 6 and in Figure 16 was used to develop an exponential model to calculate distance from the AP from the RSS. The model and calculated constants are given in Figure 7. This propagation model is used in the following localization test.

•R = Exp(P + φ)/(β)

AP	φ	β
PosiTracker0	48.2	9.7
PosiTracker1	44.4	8.3
PosiTracker2	52.1	3.6
PosiTracker3	48.6	5.5

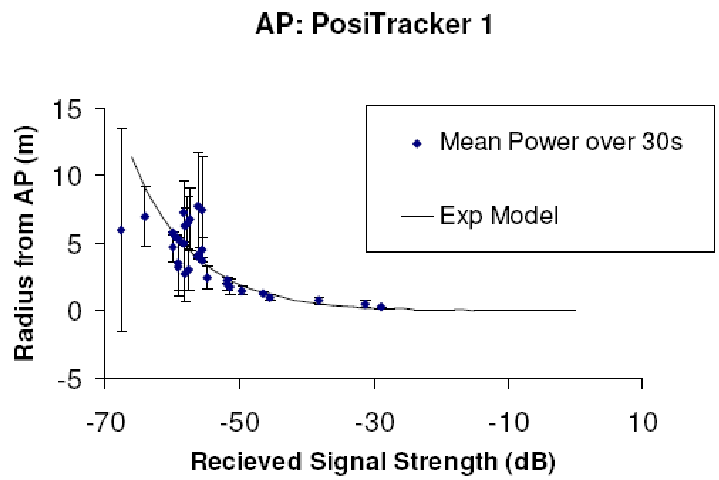

Figure 7: The exponential propagation model developed

A one dimensional localization and dead reckoning test were prepared with the test setup shown in Figure 6. The athlete walked with the DT back and forth across the 8m span. The path as a function of time is shown in Figure 8. The measured signal strength is shown in Figure 9; this data is used with the propagation model to give the location from the wireless AP shown in Figure 10.

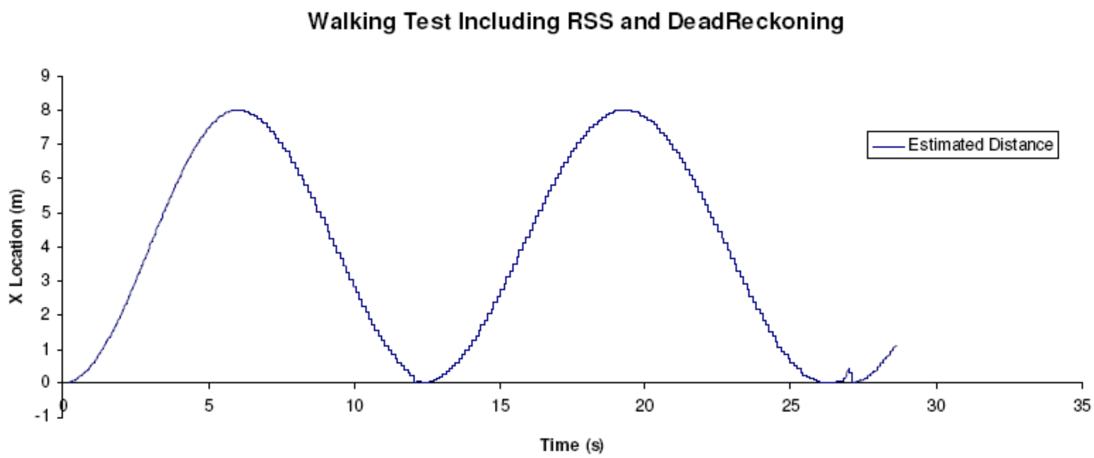

Figure 8: Estimated path taken by the athlete. The path is shown in Figure 6.

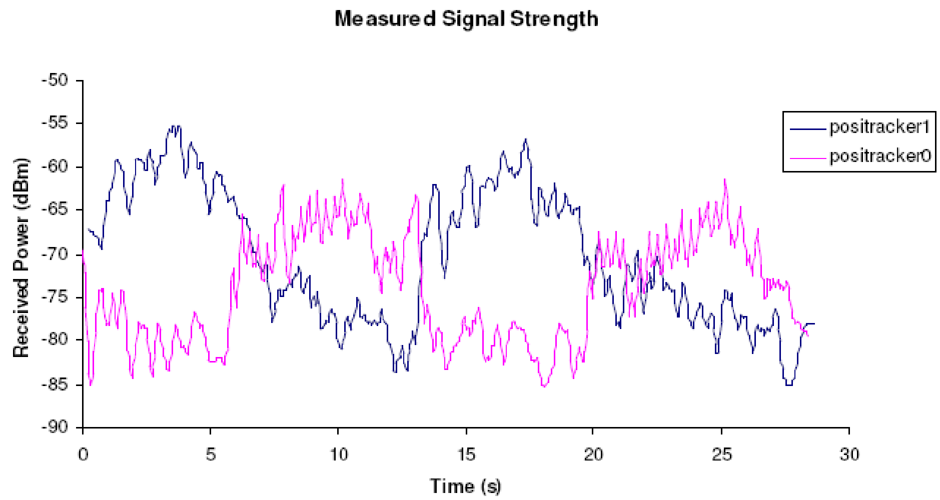

Figure 9: Measured signal strength of the APs at either end of the path shown in Figure 6.

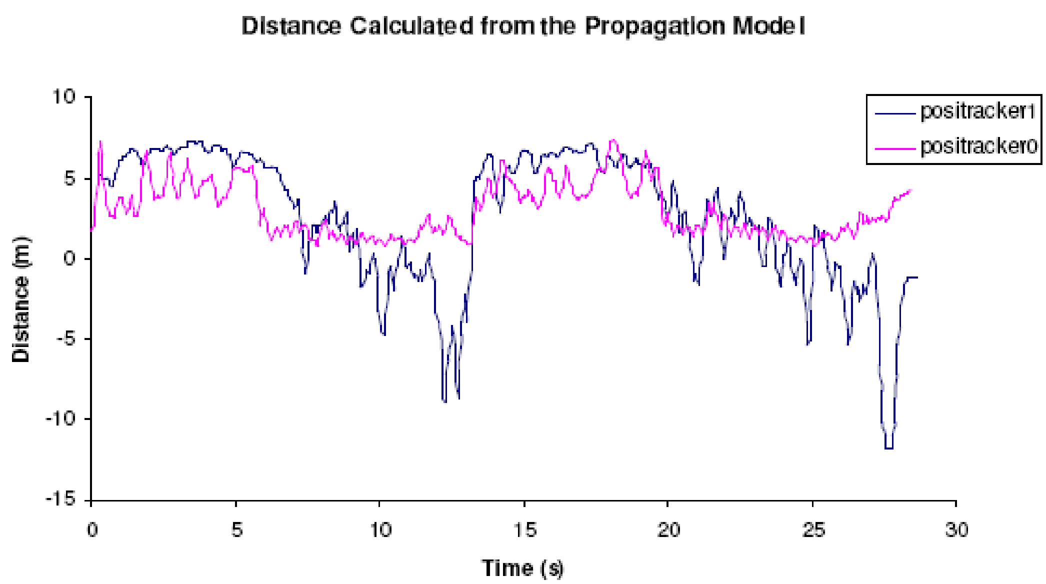

Figure 10: Calculated DT position from the APs at either end of the path shown in Figure 6.

The measured linear acceleration and rotational velocity is shown in Figure 11 and Figure 12. This data is used to dead reckon the location and velocity. The position result is shown in Figure 13.

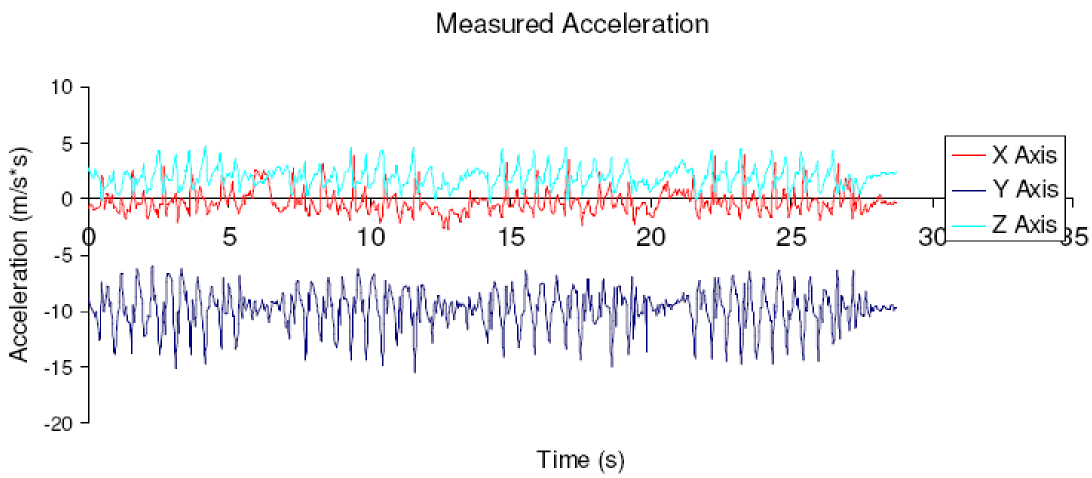

Figure 11: Measured acceleration from the inertial measurement unit.

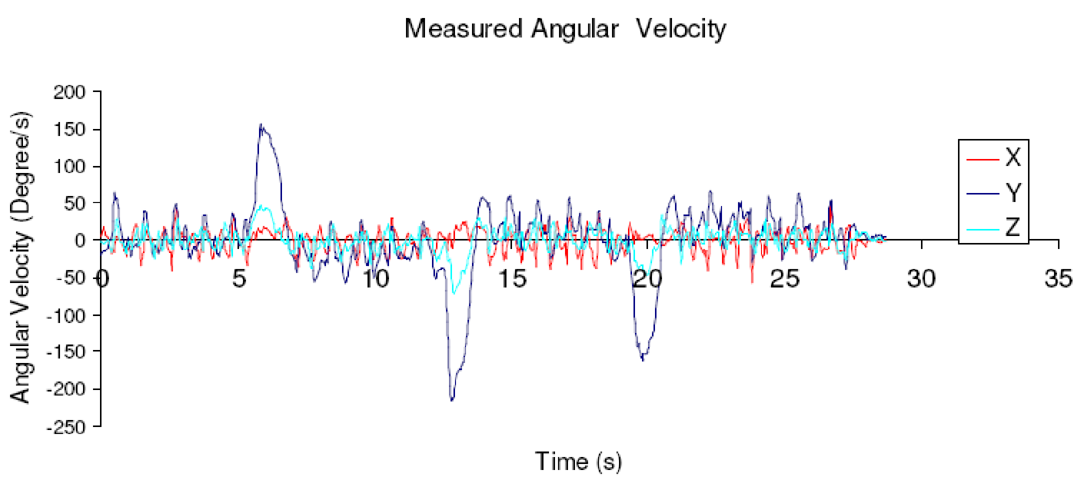

Figure 12: Measured rotational velocity from the inertial measurement unit.

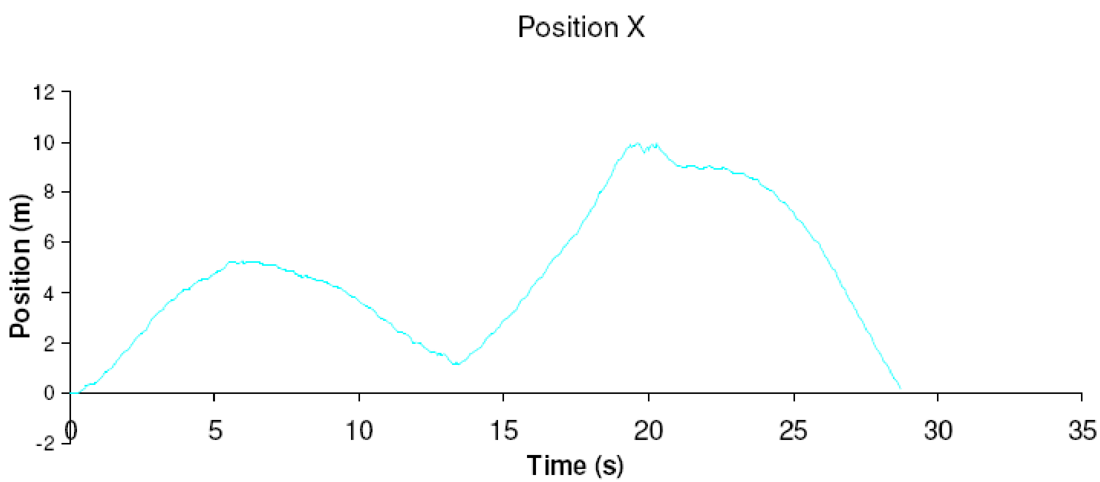

Figure 13: Calculated X position from the data measured by the inertial measurement unit. See Figure 11 and Figure 12

Figure 14 shows the comparison between the distance calculated with the RSS localization and the IMU dead reckoning. Taking the averages of this data shows the path in Figure 15. Comparing this path to the accual path give an average error of less than 2 meters.

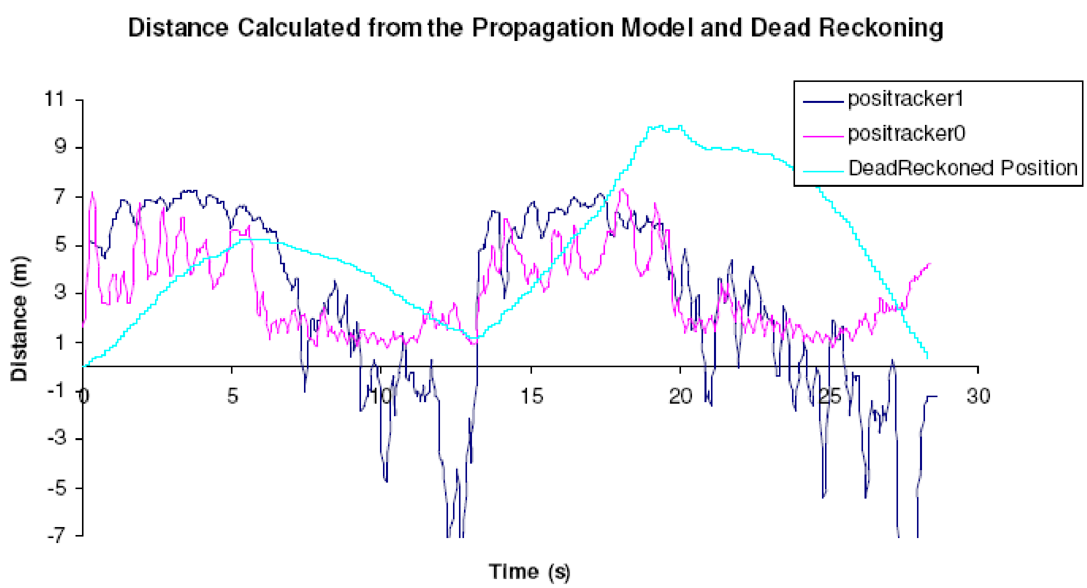

Figure 14: Overlaying the three calculated position data from RSS localization and dead reckoning

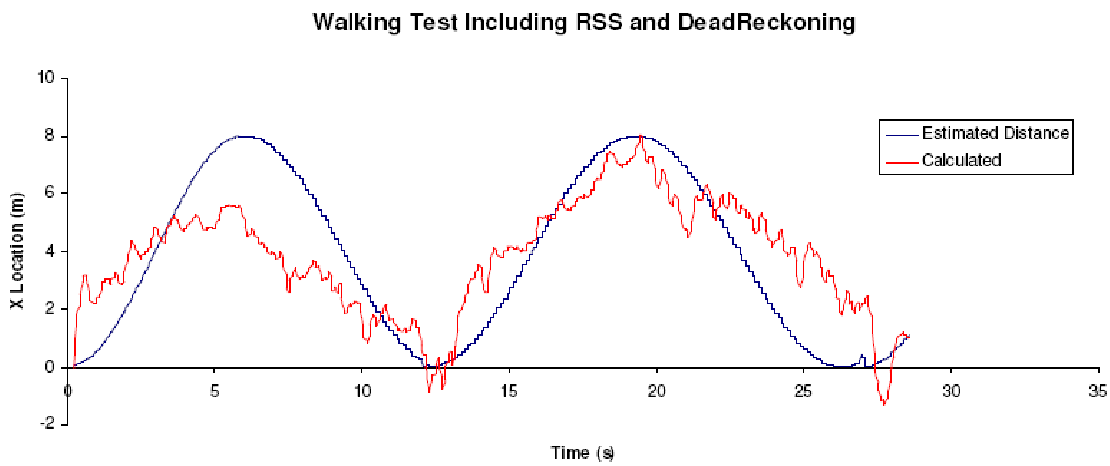

Figure 15: The highest probable location (red) compared to the estimated test distances (blue)

The next tests were done in a two dimensional space to compare different algorithms for RSS localization. The algorithms included geometric localization, probability distribution localization and Wireless Access Point (WAP) power mapping. All tests were completed in the ASB Atrium shown in Figure 16. Four APs were placed in the corners of a 5x12 meter space.

To test the geometric algorithm the athlete walked from one end of the space to the other and back. This path is shown in Figure 17 and results in a location error of less than 3 meters.

Figure 18 shows the location probability map generated using algorithm that adds the probability radii to achieve that graph. This method resulted with an average error of 3 meters.

Both the geometric method and location probability map rely on the logarithmic propagation models that were developed. These models are inherently wrong due to the EM wave reflections in the indoor space. The next algorithm, WAP power mapping, tries to solve this problem with a space dependant propagation model.

Figure 16: Signal strength mapping of 4 access points in a 5m x 12m area in a 1m x 1m grid.

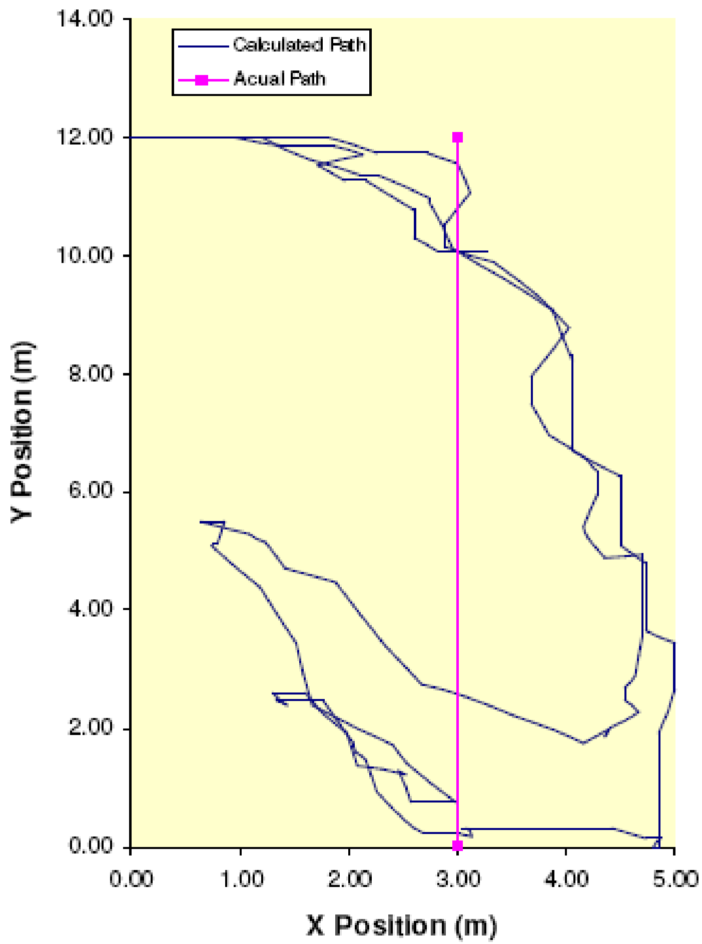

Figure 17: The results from the geometric algorithm in the space shown in Figure 20.

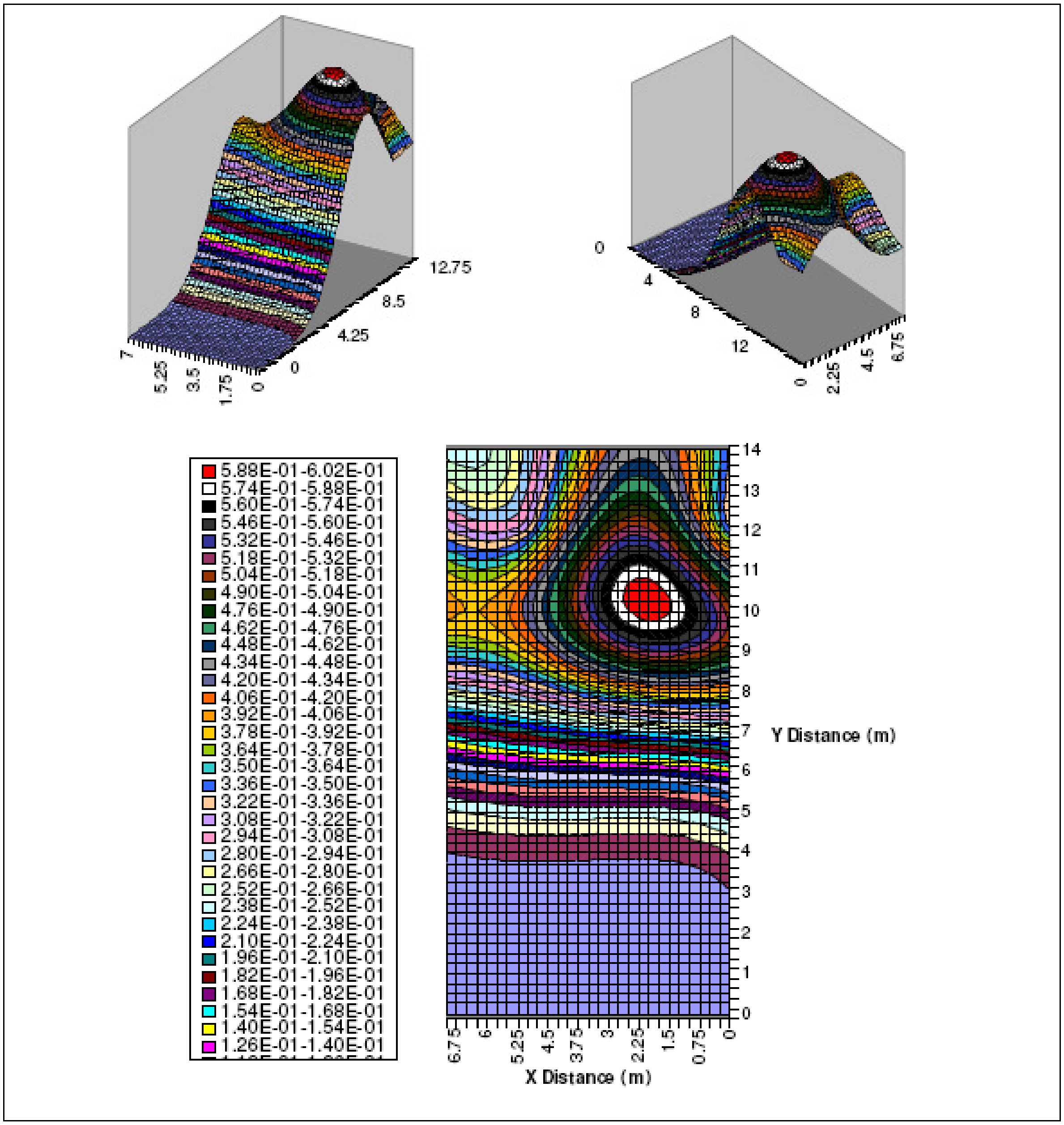

Figure 18: Location probability graph showing the highest probability of the X and Y coordinate at an instant in time. Real location at X = 2m Y = 12m.

One of the most revealing tests that we performed was in the final days before our presentation, and it was the signal strength mapping of a 5 by 12 meter test area outside of ASB 9896. Pictures of the test setup can be seen in Figure 16. In this test, the signal strength for every access point on channel 1 was recorded for 5 seconds at each 1 x 1 meter location. This resulted in over 21,000 different samples that were then used to create the following power maps show below in Figure 19, Figure 20, Figure 21, and Figure 22.

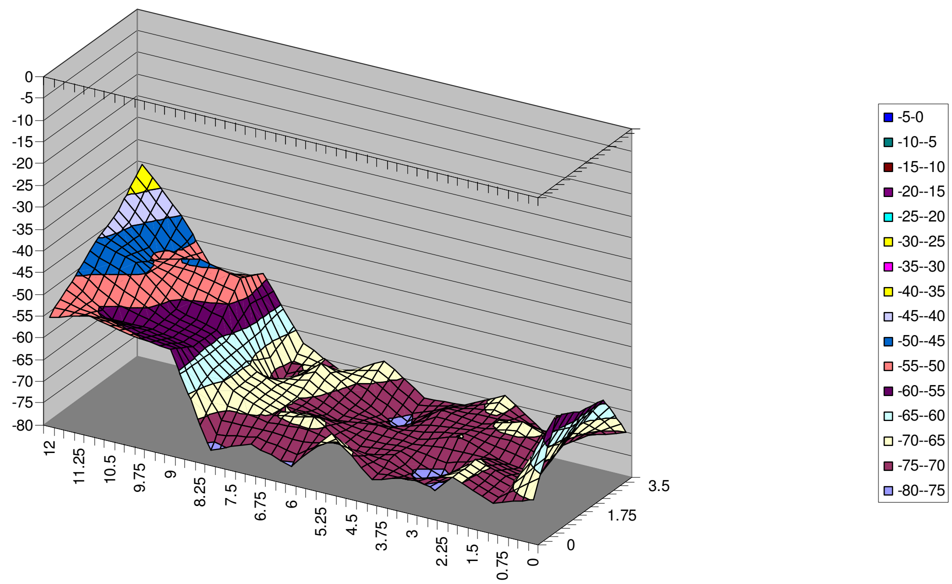

Figure 19: Signal strength map for back right access point.

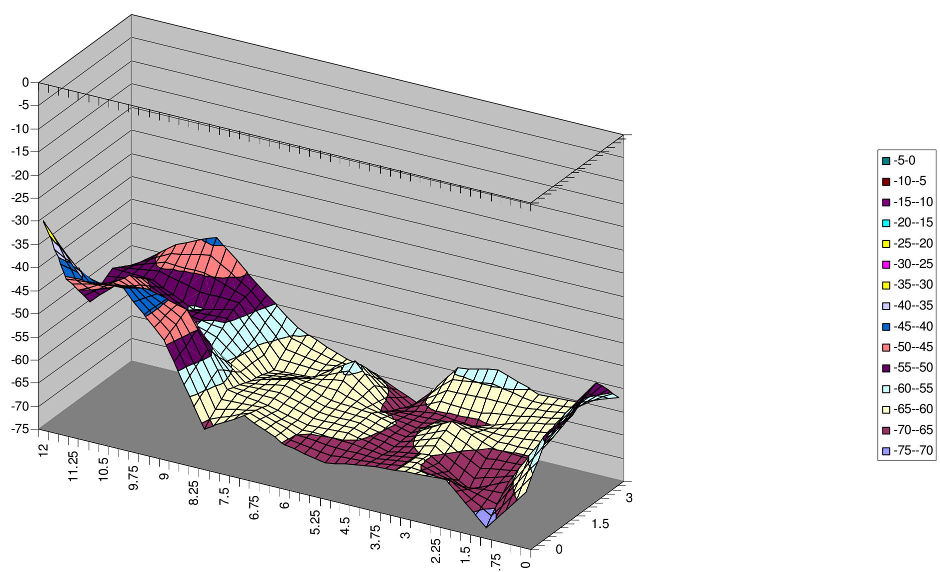

Figure 20: Signal strength map for back left access point.

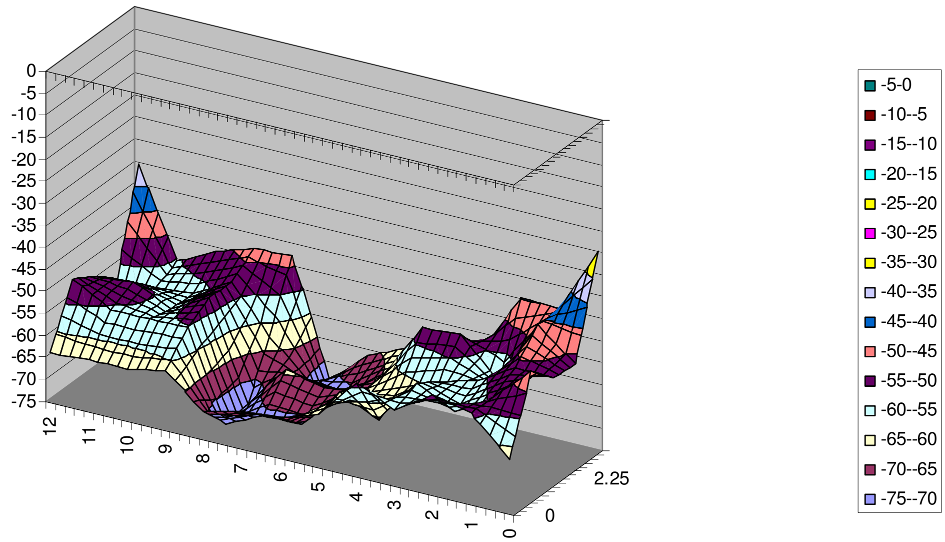

Figure 21: Signal Strength map for front right access point.

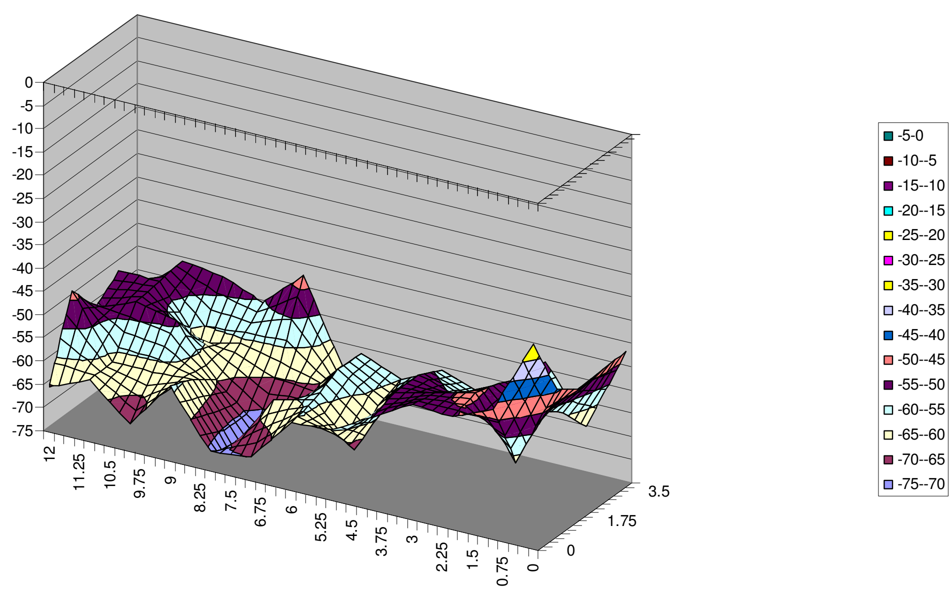

Figure 22: Signal strength map for front left access point.

Another way to view these power maps for each router is in topographical form where instead of elevation signal strength in dB is used. The topographical power maps for each access point can be seen below in Figure 23. The yellow portion indicates the location of highest signal strength or rather the location of the access point.

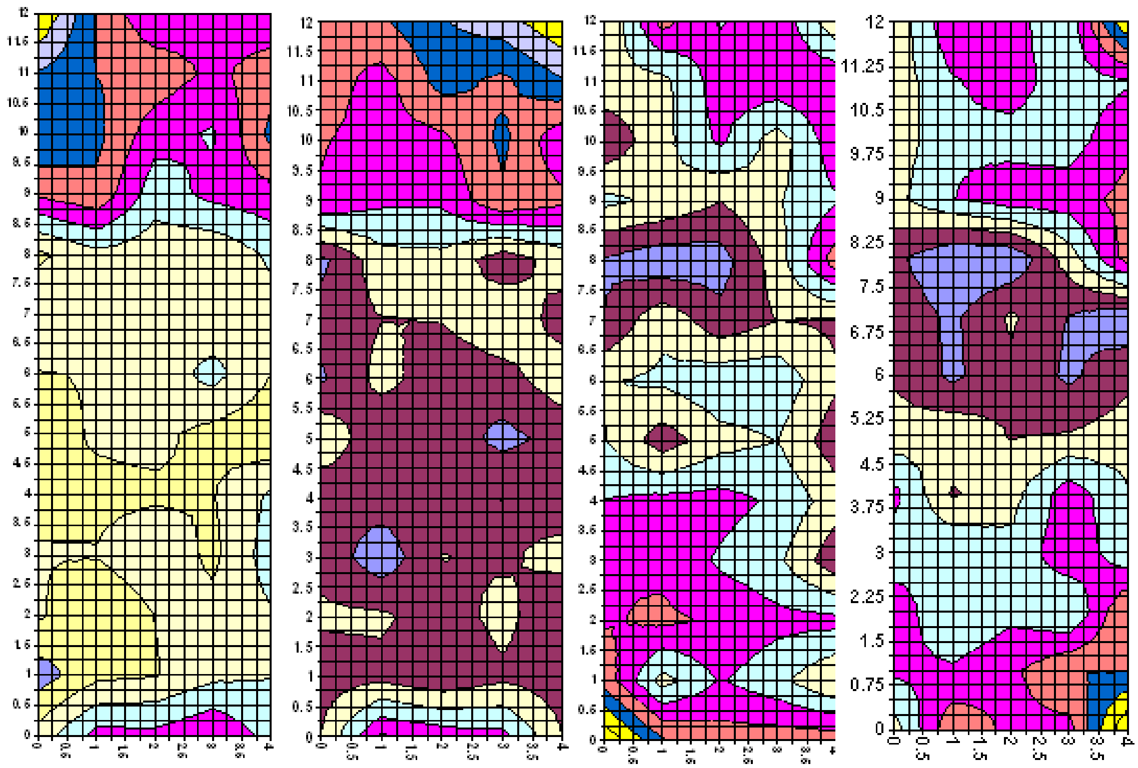

Figure 23: Topographical view of signal strength for each of the access points. From left to right: back left, back right, front left, and front right.

One method of localization we tried was to overlay the probable location from each access point to create a map of the most probable locations. This is illustrated in Figure 24 below. When a person is standing at a certain location in the test area, there are 4 different signal strengths (one for each router). For each router, certain signal strengths correspond to certain locations in the test area (ideally would be circular bands extending from the access point). By adding all of these probable locations up for each router/signal strength pair we can produce a graph of the most overall probable location. In Figure 25, Figure 26, and Figure 27, the most probable location has been calculated using this method using real data and 3 different locations. The most probable location in these graphs is either the highest point in the topographical result or the biggest overall “mountain”. Using this description of the most probable location it is easy to see that the accuracy is around 1-3 meters from the actual location indicated by the red dot.

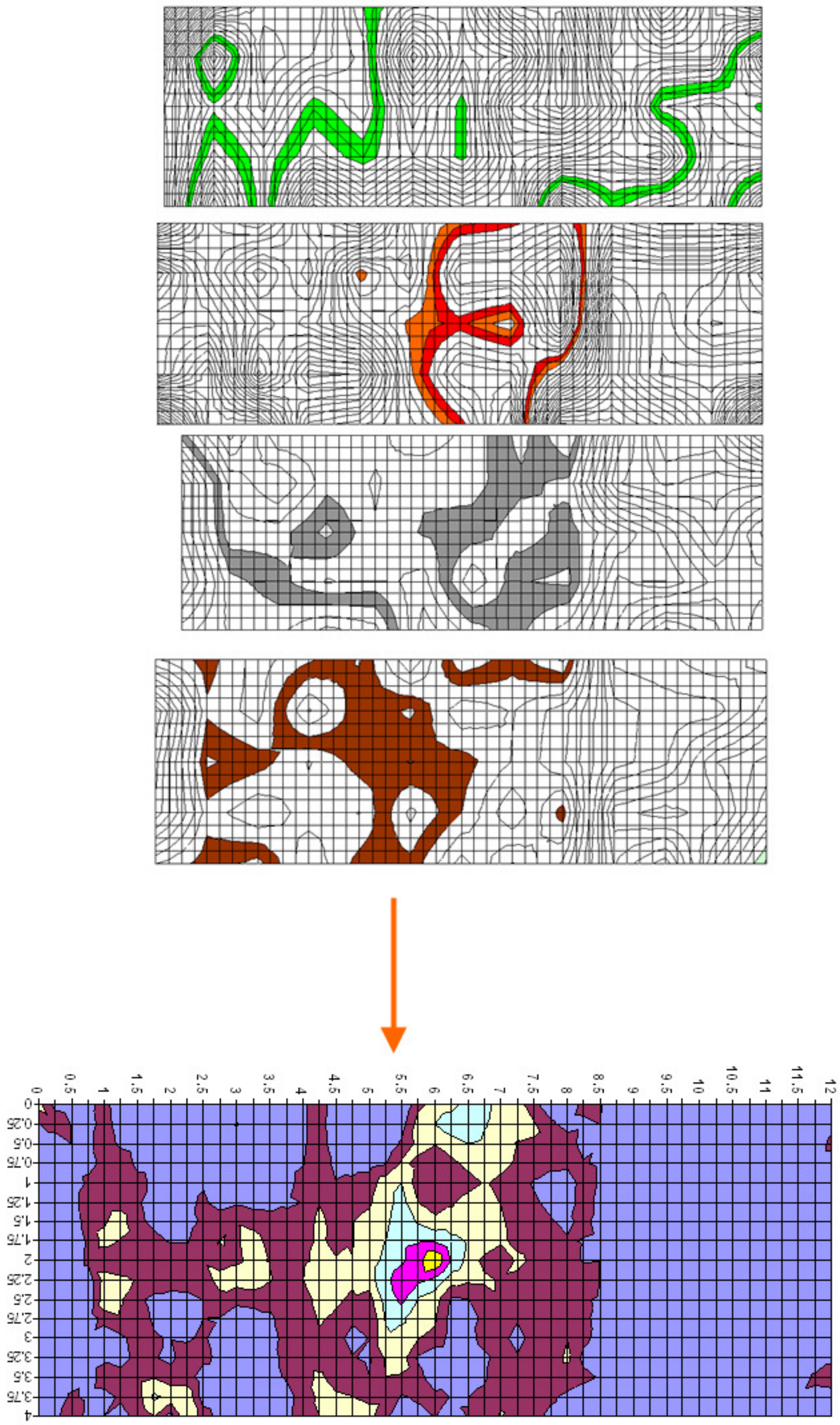

Figure 24: Illustration of the overlay method; by adding the most probable location from the signal strength map of each access point the most probable location can be found, in this case it is the middle.

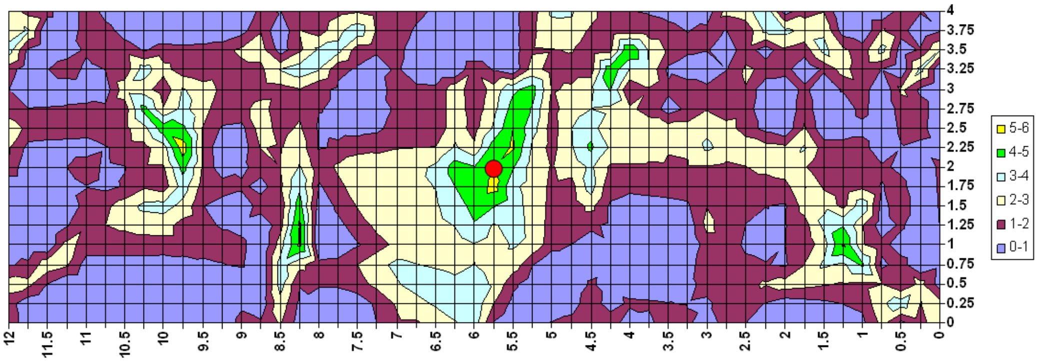

Figure 25: Overlay method shows most probable location to be approximately in the middle. Red dot indicates actual location.

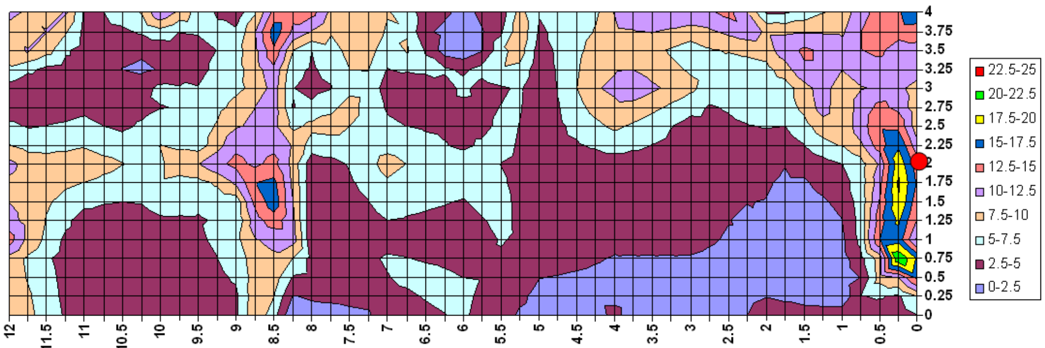

Figure 26: Overlay method shows most probable location to be approximately middle far right. Red dot indicates actual location.

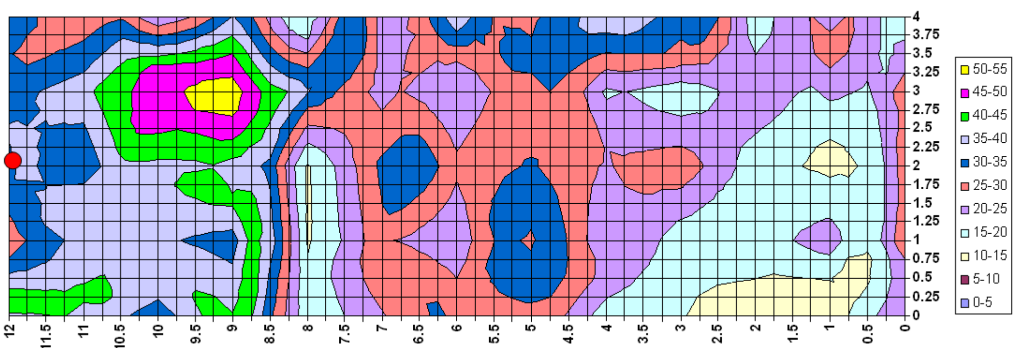

Figure 27: Overlay method shows most probable location to be approximately middle top left. Red dot indicates actual location.

The overall conclusion of the result section shows that RSS localization with a logarithmic propagation model or a power map gives location accuracy within 3m. This result agrees with our design specification. To increase this accuracy we need to use sensor fusion. Adding the IMU data in a probability filter with the RSS location generated will increase accuracy. This is discussed in the next section: future plans.

4. Future Plans

The current state of this project allows for further research into Wi-Fi localization and acceleration based dead reckoning. The progress of this project also allows for the ability to add more features to create a market product for coached and broadcasters.

- Research:
  - Create a more complex propagation model that considers environmental factors
  - Add probability filter to combined dead reckoning and RSS localization
  - Test the system at various locations including sports areas
- Product Development:
  - Decrease power consumption
  - Expand to track multiple units
  - Reduce size of the diagnostic tool (create custom chip layout and board)
  - Create a more user friendly GUI
  - Integrate heart rate monitor and other biometric measurement tools
  - Integrate video feed
  - Create other modules including the physical and tactical training modules

Once the technology is at a level for sale to our potential customers, the PosiTrack team will begin marketing the product to various training groups and sports teams. Another option would be to approach our competitors with the product and see if they have interest in purchasing the technology. All future decisions will be based on the economic advantages and disadvantages of the presented options.

5. Final Schedule

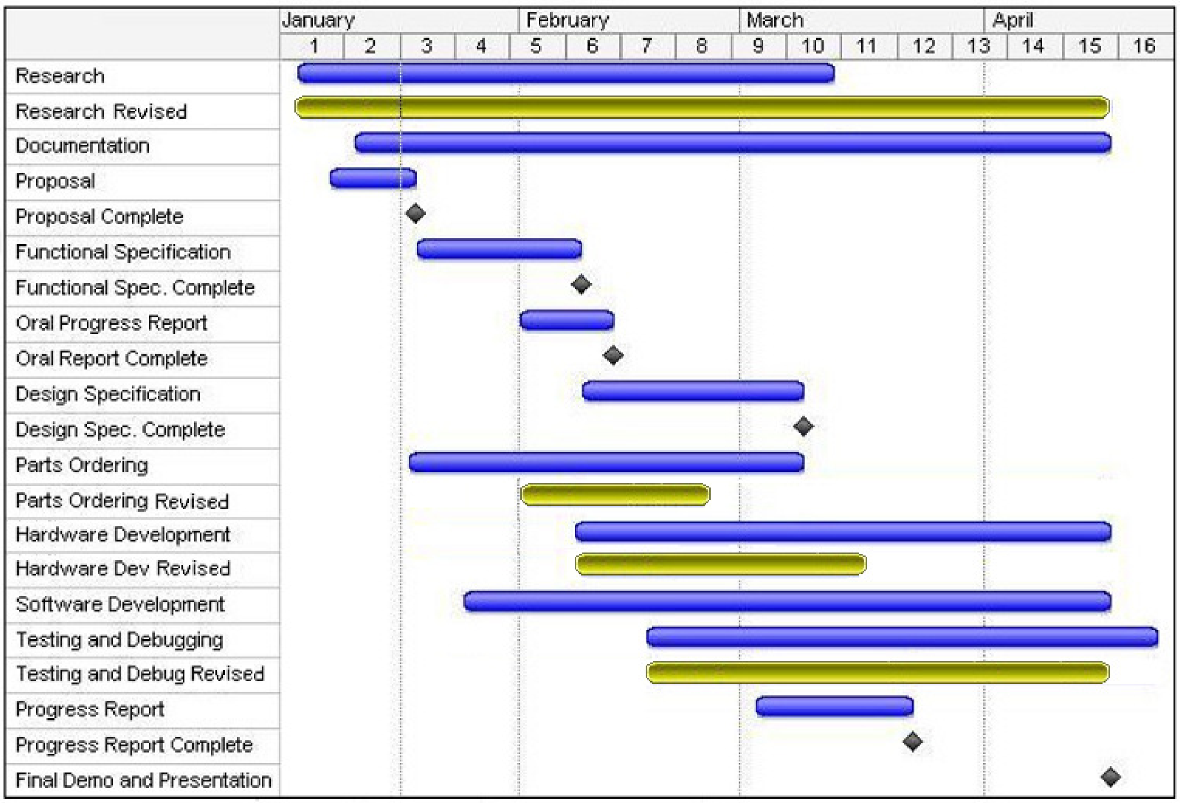

Figure 28: Gantt chart for the development of the PosiTracker.

Our updated and final Gantt chart is presented above in Figure 28. Blue bars indicate our original estimated schedule at the time of the proposal whereas the green bars indicate the actual scheduling of the tasks in development. We overestimated in our original plan to allow for unforeseen delays during the development stages. As seen, we were very efficient in developing the hardware and only used half of the time allotted by our original estimate. Research was initially thought to be completed roughly halfway through development but was continued throughout due to the complexity of the localization algorithms.

6. Final Budget

Expense	Source	Unit Cost
Overo Air Gumstix	www.gumstix.com	$219.00
Tobi Expansion Board	www.gumstix.com	$69.00
WiFi Antenna	www.gumstix.com	$10.00
5V AC Wall Adaptor	www.gumstix.com	$10.00
MicroSD Card 2GB	www.gumstix.com	$20.00
LiPoly Charger	www.sparkfun.com	$16.95
IMU 6DOF Razor	www.sparkfun.com	$89.95
1000mAh Li-Poly Battery (x2)	www.sparkfun.com	$23.90
Misc Parts / Costs		$46.00
Shipping Costs		$61.75
Final Cost:		**$566.55 USD** **$581.90 CAN**

Table 1: The final cost of developing the PosiTracker prototype.

Table 1 above shows the total cost of developing the PosiTracker prototype which came in under our initial cost estimate of $612.54. The high cost of shipping was offset by borrowing wireless access points which was a key point in meeting the original budget. The funding that was received only covered $450, the remaining balance of $131.90 was split amongst group members.

7. Group Dynamics

This section outlines how the members of PosiTrack systems worked as a team. Here outlines how the tasks were allocated and the technical difficulties encountered with each of them. The interpersonal difficulties are discussed as well as the recommendations for future projects with respect to group dynamics.

7.1. Task Allocation

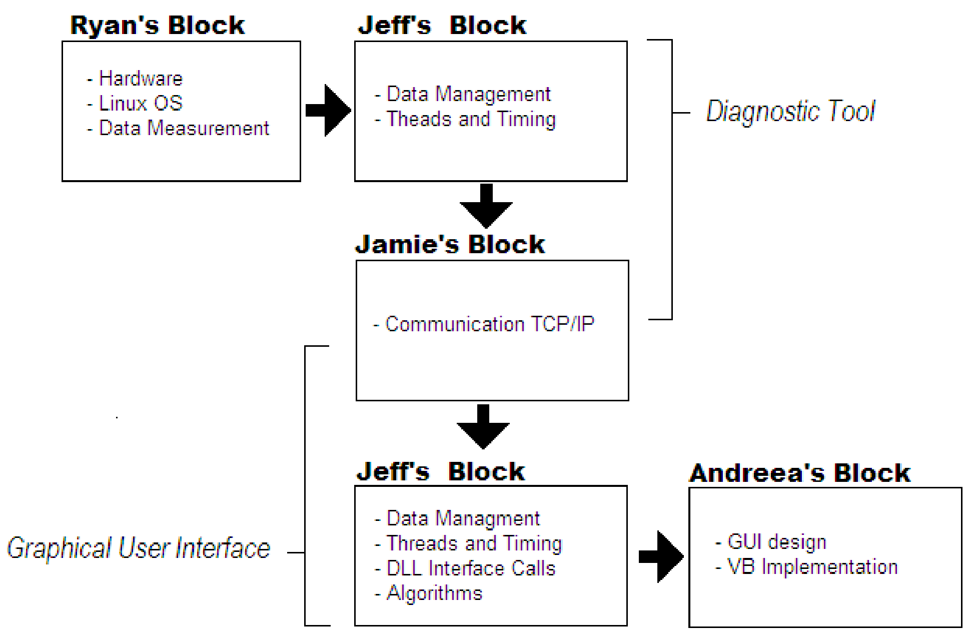

Figure 29: Shows the blocks of work that were allocated to each individual in the team

The above Figure 29 shows a diagram drawn at the beginning of the semester outlining the tasks delegate for each group member. The choices of tasks were based upon the skills of each group member. This diagram held true for a majority of the project. Close to the end of the course Ryan and Jamie finished their task blocks and moved to help Jeff test and implement different algorithms.

7.2. Difficulties Encountered

This section outlines only some of the major technical and interpersonal difficulties that arose from completing this project.

7.2.1 Technical

In Ryan's task block a large issue was the speed of sampling the Wi-Fi signal strength. The problem was the Linux command iwlist did not support the functionality that we desired. The solution was to create a custom iwlist function by modifying the source code.

A hardware issue that Ryan encountered was one of the AToD ports did not function properly. The manufacture was a fault for this problem and switching boards solved this problem.

Another issue was when soldering a glob fell onto our Gumstix board possibly rendering it useless. Luckily heating up the solder and using a CO_2 cartage to blow the solder off the board solved the problem.

In Jeff's first task block the major problem was facilitating the measurements of both Wi-Fi RSS and acceleration. This was solved using simple threads. Another problem that arose in this block was the portability of data between Linux and Windows OS. This was solved by creating custom data packing and unpacking algorithms.

In Jamie's task block an issue became present when transferring large amounts of data quickly over the TCP message passing software. The problem was that the TCP method will retransmit damaged data. This retransmission slows all data transmission. This problem was solved by sending smaller data packets over UDP. This creates small losses in data but allows for real time transfer.

Creating and implementing algorithms became an issue due to the very nature of the challenge. Noise and environmental factors created inaccurate propagation models. These models affected the accuracy of the localization. Also the complexity of the algorithms consumed two much computing power. A potential solution to these problems is to create similar location algorithms with more complex propagation models.

In Andreea's Task block the major issue was the method to display the graphs on the screen. The two methods that were considered were writing directly to the COM port or using the bit map function in VB. The solution chosen was the bit map method to save time and increase functionality.

7.2.2 Interpersonal

In our group the main issues arose due to the communication short falls and nonparallel production of the product. The speed of production was also an issue which affected the end product. The main reason for these communication and production problems was the lack of test infrastructure. Because this team had a very limited budget we had a limit testing infrastructure. This limitation created a non parallel production flow. At some times group members were not able to test on the correct platform or with the right tools which created delays in production. This limitation also created communication errors while interfacing the software modules.

7.3. Recommendations for Future Projects

The recommendations to be made for future projects when considering the issues in section 7.2 are discussed below. Our team recommends when managing a team you allocate tasks that allow for the most parallel production but make sure that you have enough testing infrastructure to complete this parallel production as rapid as possible. To solve the communication problem our group should have adopted more frequent and smaller mile stones. Our team believes that this would increase production speed and decrease communication errors.

8. Individual Reflection

Andreea Hrehorciuc

I was mostly in charge with designing the GUI. Having so many options to choose from for the IDE, I had to spend some time at the beginning of the semester just to get familiar with some of them and to see which one would be most suitable for the project we had decided on. I looked at several options for Linux IDEs and a couple of IDEs for Windows, but eventually chose an IDE that would make the product look more professional, at the expense of more work to do.

During the 4 months working on this project, I have learnt how important integration testing is. Due to the nature of our project, there were 4 parts to it: algorithm research, programming for Windows (GUI), programming for Linux (the diagnostic tool) and putting hardware together. We have worked independently on these sub-sections, for the most part, and the integration of the (almost complete) parts turned out to be non-trivial. If I were to turn back time, I would probably integrate all the small accomplishments in each of the 4 areas, before moving on to the next part.

The most important thing I have taken away from this experience is, perhaps, the benefit of working in a team. My group members come from other specializations than myself, and that was helpful when I encountered difficulties.

Jamie Valdes

My main role in the project was working on the communications between the GUI and the diagnostic tool and many small tasks after that. Through lots of trial and error with coding, I learned a great deal of researching and testing techniques as well as the importance of constant integration. I now appreciate the importance of standardization across an entire product, especially in programming.

In the beginning stages of the project, I got experience with selecting components by considering the system requirements, budget and time constraints which I had not had to have to work with previously.

I'm very lucky to have had the opportunity to work with such a talented group of people. Every individual in the team brought different skill sets and expertise to make a very diverse team. I learned a lot about team dynamics, how to deal with partners in a group effort, how to work with others' ideas and how to contribute my own ideas. From working in this group, I feel that I am more capable of tackling large design and implementation projects in the future.

The most valuable experience in this project which will be taken away is the realization of a product from an idea to a functioning prototype which probably is the most applicable lesson to be applied in a future career in engineering.

Jeff Anderson

Looking back at the past 4 months I see an experience that was humbling and educational. If I had a conversation with myself at the beginning of January it would consist of my past self listing off all the great things this product will be able to do. And current self would have to interject with all the limitations and barriers that past self would meet. The largest barrier would be the constraint of time and the second largest barrier would be the environmental aspects affecting our technology.

My role in this project was to create software that interfaced between Ryan's, Jamie's and Andreea's code which facilitated and accomplished finding the position, velocity and acceleration of a player in real time. This task put me in a position to manage the surrounding tasks and communicate how the interfaces should connect. I found that because everyone's task was very interdependent we needed to be on top of each others work and progress. This connection lead to a group oriented experience that was very communication dependant.

I also found this project to be an addictive experience. As a veteran of the university class work, I know how to learn and write tests. The goal for this process (of writing tests) is to take knowledge and commit it to memory. This goal, after 5 years, can be very mundane and boring; however the goal to create a project like this is very different and task dependant. I found myself determined to accomplish the next task at hand by any means necessary. This is a very different process than school work and I found it very addictive.

I also enjoyed the freedom that Co-op work could not offer. This project, having no real guide lines, was a breath of fresh air. The fact that we could choose any problem and dive into any potential solution titillated the mind to levels that no other class has before. If it wasn't for the memory of sleepless nights in the ASB I would almost want to take this course again.

I would like to thank everyone who worked on this project as well as everyone who worked hard to facilitate ENSC 305/440.

Ryan Lynne

My major role in this project was to create the hardware/software for the module located on the athletes back. In order to create this module I first had to juggle the tradeoffs between cost, size, weight, power consumption, speed, and time to market when I was selecting the appropriate hardware. In the end I chose the Gumstix Overo Fire as the heart of the module and based the other design decisions off of it. From the beginning of the project I preached to the other group members that a software solution versus a hardware solution would greatly benefit us due to our short development period (or our version of time to market). By choosing the Overo we were able to spend most of our time coding instead of developing hardware and in the end allowed us to meet almost all of our goals. This experienced for me again reinforced the tradeoffs between a software and hardware solution; a hardware solution should be used for high performance or high volume sales whereas software should be used adequate performance or low volume sales.

This decision to use "essentially" a software approach however was not exactly an easy path to take as I originally expected. What I expected, more like a pipe dream, was to attach the battery and accelerometers/gyroscopes to the Gumstix Overo and instantaneously we would have a WiFi connection, WiFi signal strength, gyroscope measurements, and accelerometer measurements. If I were to do this project again this would probably be the case but there was a large learning curve for using Linux as an embedded development platform. The biggest difference that I experienced when using Linux was that it was a lot harder to play in "Godmode", that is the operating system limits what you can do in userspace and to have "Godmode" you must place your code in the Kernel.

Another challenge that I had was to set up a development platform to allow for us to actually write code for the Gumstix and gain access to it. I would not have been able to do this without the help from my father who has been using the Gumstix as a development quite sometime. With his help I was able to create a bootable SD card with Linux and setup cross-compiling with the ARM tool chain to compile programs for the Gumstix. For a then novice Linux user this would have taken me forever to do alone, even with his help it took a week to get a "hello world" program running. Simple things like transferring files to the Gumstix, improperly linked header files, and setting up the wireless connection to an access point took easily 1-2 weeks of my time. One of the reasons I also choose the Gumstix was that I could leverage my fathers experience if I needed it and this proved to be an excellent decision! Many of the problems I experience were because of the sometimes steep learning curve associated with Linux.

Once the Gumstix was ready for development I had to write/modify software drivers to gain access to the accelerometer/gyroscope data and the wireless signal strength. The access to the A/D ports to read the accelerometers/gyroscopes proved to be pleasantly easy and only took two days. However, to calibrate the accelerometers proved to be tricky as the center voltage and sensitivity are ratiometric (vary with input voltage). It took another several days to calibrate them properly (i.e. reading gravity properly). One major mistake I made was to assume my desk at home was level, which it apparently was not. Once I starting using a level my results improved greatly. The most painful part of this project for me was to read WiFi signal strength from multiple access points. The software drivers for the wireless chipsets nowadays do not support fast reading of this data. On average, it takes 2-4 seconds for a typical software driver to scan the signal strength of multiple access points. This is terrible resolution as we wanted this data about every 100 milliseconds. I knew we could achieve this but I didn't know if we could do it without having to write our own software driver which would have been far beyond the scope of this project/course. I initially spent about a month trying to get a reasonable scan time but gave up after I had spent to much time. I came back to the problem about 2 weeks before the presentation and found a simple but effective solution that gave us a scan time of about 130 milliseconds! In the end, I had to only change fewer than 10 characters in the software driver to do this. The solution was to make the driver only scan 1 channel for a total of 20 milliseconds. Originally the driver would scan all 16 channels for 100 milliseconds which would give a scan time of 1.6 seconds plus processing time. To get access to the kernel information I hacked apart the program iwlist to do exactly what we wanted. This was the hardest part of the project for me and it was a great moral victory when it was finally working.

After I had completed the hardware module, I moved on to help Jeff with the algorithms to locate and give the statistics for the athlete. This proved to be quite tricky and we unfortunately ran out of time spending many late nights in the ASB. At this point in time I can't say whether this product can actually be marketable as we would need more time to test localization algorithms and actually do full size tests in a hockey arena or field. One thing I will point out is that the accuracy we were seeing was in line with that of GPS meaning we could compete with our competitors and ours would work indoors unlike their products. An important thing to remember about this product is that the actual location is not as important as the distance travelled, speed, acceleration, and heart rate. For example location is not that important for broadcasting and training purposes. The localization in this product is to simply correct the drift associated with dead-reckoning. I believe that this product could deliver these statistics fairly accurate and at least in line with the competitors products.

In the end I learned extensively about developing an embedded system under linux: cross-compiling to different platforms, interfacing to the kernel, software drivers, kernel modules, and differences between userspace and kernel space. Additionally I gained a new interest in wireless technology and just how amazing it is. The smallest signal picked up by modern WiFi can be up to a billions times smaller than the original signal.

One note about the course in generally is that I spent an enormous amount of time doing documentation (~4 weeks) which I would rather have spent developing our product. One recommendation for this course would be to spread out the course over 8 months, or possibly make ENSC 305 a prerequisite for ENSC 440. Even if this meant eliminating some other academic course I would strongly still recommend it as these two courses give hands on experience versus book knowledge.

Thank you to everyone who helped out, especially my father for his help and the ESSEF for their generous financial contribution.

I would also recommend the Gumstix as a development platform, definitely a cool and well thought out product!

9. Conclusion

At this point in time the PosiTracker is not a marketable product as the algorithms we are using for signal strength localization are underdeveloped and in their infancy due to the complexity. Full scale tests and further time is needed to develop the localization algorithms in order to first gauge the possibility of making this product a reality. However, the current system can be used as a platform to conduct these tests with great ease.

The current tests results that we obtained were comparable to that of GPS which is what our competitors are using in their end product. This possibly means that with further development our product can upstage our competitors in an indoor environment. A lot of development time was spent on localization but in reality the location is not as important as the distance travelled, speed, acceleration, and heart rate with respect to broadcasting and training purposes. If we were to continue development, localization in this product would be to simply correct the drift associated with dead-reckoning. We believe that this product could deliver these statistics accurately and at least in line with the competitor's products.

