Engineering Quality in the Robotic World

Vicki Niu, Ethan Takla, Ida Chow, MacLean Freed, Jeffery Wang and Kingsum Chow
{nanites4092}@gmail.com, {kingsum.chow}@intel.com

Abstract

This paper explores the challenges of attaining engineering quality in robotics through the experiences of the Lincoln High School robotics team, the Nanites.

In robotics, engineering quality in hardware-software co-design is important. While software is written in the abstract world, it is executed on robot's hardware, which interacts with the physical world to accomplish its missions. The quality of the interactions between the robot's sensors, motors, and the physical world is critical to the success of the missions.

Testing software quality on a robot is different from testing software on an information system. In the physical world, the state of a robot is never precisely known. Sensor readings inevitably contain errors and often provide insufficient information. The robot's motions, no matter how well controlled, are inexact. As a result, accomplishing a task using a robot requires tolerance of some error. Testing software quality on a robot in the physical world thus requires a different mindset than testing software quality in the information technology world.

Through the case study of building and testing a robot with motors and sensors, this paper characterizes the uncontrollable and variable environmental factors encountered in the physical world and the uncertainties from sensor readings, especially those regarding the location of the robot.

The contributions of this paper are:
1. To share the lessons learned from striving for engineering quality in robotics and hardware-software co-design.
2. To demonstrate that high school students are inspired to learn advanced sciences and mathematics not simply for exams, but for robotics competitions, and the sake of learning.

Biography

The authors are members of the Lincoln High School robotics team "The Nanites". They presented a paper at PNSQC two years ago as a FIRST LEGO League team. In 2010, they received the first place programming award in the FIRST LEGO League World Festival in Atlanta, Georgia. The Nanites became a FIRST Tech Challenge robotics team in 2010, upon entering high school. In addition to striving for hardware and software excellence, the Nanites are very involved in educating youth in the Portland metropolitan area about robotics through camps, workshops, and demos. More information about the team is available at http://nanites.zymichost.com

Kingsum Chow is a Principal Engineer at Intel. He works in the System Software Division (SSD) of the Software and Services Group (SSG). He joined Intel in 1996 after receiving his Ph.D. in Computer Science and Engineering from the University of Washington. Since then, he has been working on performance, modeling and analysis of software applications. Currently, he leads the Java Middleware performance optimization on Intel architecture. He holds more than 10 patents and he has presented more than 40 technical papers. In his spare time, he volunteers as a coach for multiple robotics teams, bringing the joy of learning about "Science, Technology, Engineering and Mathematics" to the students in his community.

1 Introduction

This paper explores how a high school robotics team, the Nanites, overcame the obstacles in attaining engineering quality in robotics. The Nanites are a FIRST TECH Challenge robotics team. FIRST TECH Challenge (FTC) is a high school robotics competition that is becoming increasingly popular.

The Nanites' work on quality assurance through robot competition dates back to 2005 when they first competed in FIRST LEGO League. Other FIRST programs include Jr. FLL and FIRST Robotics Challenge (FRC). Jr. FLL, for kids ages 6-9, consists of a challenge similar to the theme of FLL for that year. Teams create a LEGO model based on research that they do relating to the year's theme. In FRC, teams of high school students build larger robots and compete on a much larger field than in FTC.

1.1 FIRST LEGO League

First Lego League (FLL) is a robotics program for students ranging from elementary school to middle school. The students utilize a LEGO NXT brick, LEGO Mindstorms pieces, and NXT-G programming software to program an autonomous robot to complete various missions on a FLL table for two minutes and thirty seconds. Figure 1 shows the FLL field setup for the 2011-12 season. Each team builds a robot to accomplish missions while navigating around obstacles on the table. This experience introduces participants to critical thinking, team building, presenting ideas, and applying math and science to solve real world problems. FLL team members learn skills that will benefit them for the rest of their lives.

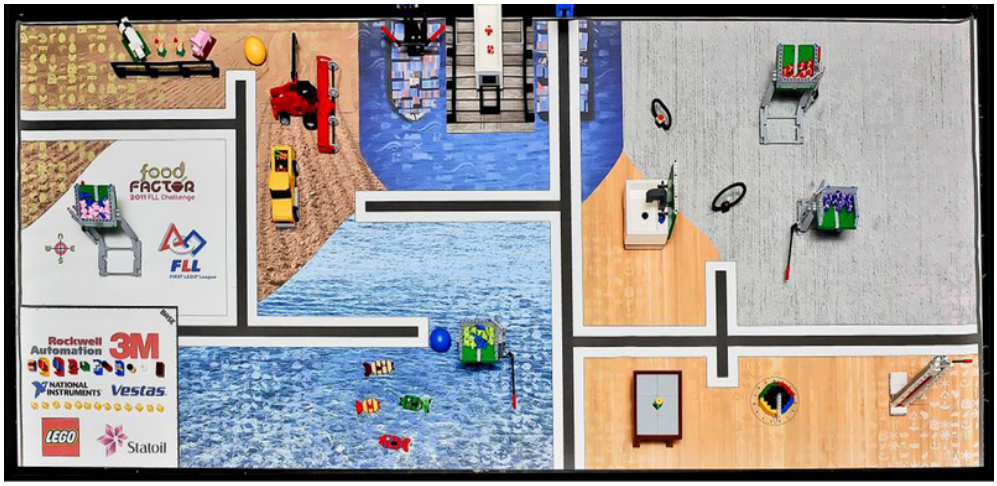

Figure 1: FLL Mat

The first years in FLL were predominately a learning experience for the team as they had no prior experience with engineering methodology. Over the years, the Nanites learned about programming, teamwork, sensors, and many important life skills. The team attempted to achieve quality assurance in progressively more complex tasks, from having an effective way to connect motors to the robot to having the most effective and reliable line-following algorithm. Such experiences led the team to discover that a group effort is better than an individual one.

1.2 FIRST TECH Challenge

In 2010, the Nanites progressed into the FIRST TECH Challenge program. There are many important differences between the FLL competitions, for elementary and middle school students, and the FTC competitions, for high school students.

In FLL, the competition field consists of a tabletop field measuring 8' x 4' [1] while in FTC, the competition field is 12' x 12' [2]. On the FTC field, four 18" x 18" robots compete against each other and must be able to navigate around each other without breaking when hit by another robot. Conversely, a LEGO NXT robot does not require the same amount of durability as the FTC robots do because it has the field to itself, leaving room for more unwieldy contraptions on the robot. This is an added complexity of FTC, as the variable of three other robots' actions are introduced, while an FLL robot is always the sole player on its field. Figure shows the field setup for the 2011-12 FTC season. Each alliance team of 2 robots moves from their bases, the squares in the lower right and upper left corners, to the field and performance various actions such as picking up racquetballs and putting them into the crates.

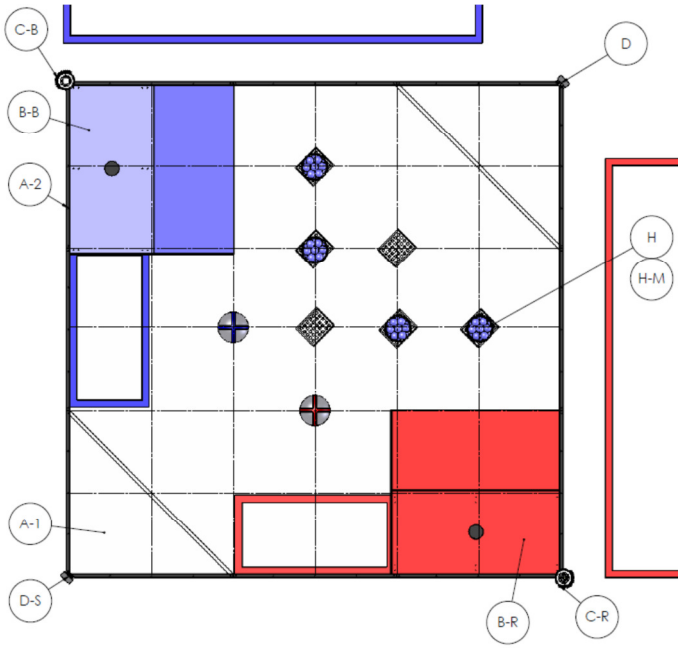

Figure 1: FTC Field Setup for the 2011-12 season

In FLL, the robots' actions are completely automated. In FTC, there are 30 seconds of autonomous robot action followed by 120 seconds of tele-operated (TeleOp) control, when two team members drive the robot with Logitech controllers to complete other goals on the field [3].

There are limited LEGO sensors available for use in FLL, such as a touch sensor, a sonar sensor, and a light sensor. In FTC, a prototype board may be used to utilize any number of specialized sensors, giving teams a virtually unlimited range of sensors that may be used [3 §4.2]. FTC teams can thus use custom sensors that allow more freedom and better results than the prescribed LEGO sensors.

Many FLL teams use NXT-G, a graphic-based programming language that has restricted capabilities. In FTC, a C-based system, RobotC, is used most widely. RobotC has data logging [5] and computational capabilities far more advanced than those of NXT-G, making it better equipped to analyze many sensor readings. RobotC also includes the Virtual Worlds package, where programs can be tested before being put on an NXT block and used with a physical robot, increasing the quality assurance of programs put on the robot.

1.3 Sensor Reliability

FTC is a good case study of software quality. Although consumer-level sensors have become more accurate in recent years, most sensors are still erroneous to some degree. These errors, if uncorrected, can result in unreliable and low-scoring robot performance. The quality of software written to compensate

for these errors can be tested in the physical world using qualitative observations and/or a robot localization system. Many sensors included with the FIRST TECH platform show much inaccuracy; to attain reliable performance, the students may choose to develop their own, more accurate, custom sensors.

Through the case study of building and testing a robot with motors and sensors in FTC, this paper characterizes the variable environmental factors encountered in the physical world, many of which are uncontrollable. The paper characterizes the uncertainties from sensor readings, proposes solutions to address these uncertainties, and then evaluates the effectiveness of those solutions.

2 Software Quality Issue: Achieving a Known State

The autonomously controlled period in FTC requires that the robot know its location. As stated in Cox 1991, "using sensory information to locate the robot in its environment is the most fundamental problem to providing a mobile robot with autonomous capabilities." In competition, there is infinite variability and complexity, so the robot is incapable of ever truly knowing its position.

Determining the robot's location is difficult because the factors in the physical world are constantly changing. Recognizing this, the authors used sensory technology in an attempt to gain information with regards to the robot's environment. Although this did increase the robot's certainty of its own position to some degree, many new complications arose due to the unreliable nature of the sensors. The commonly-used sensors in FTC are problematic. A LEGO-provided ultrasonic sensor, which uses ultrasonic waves to measure the distance from a target, was tested for its accuracy and reliability. As seen in the figure below, the readings have large amounts of error, making for unreliable performance

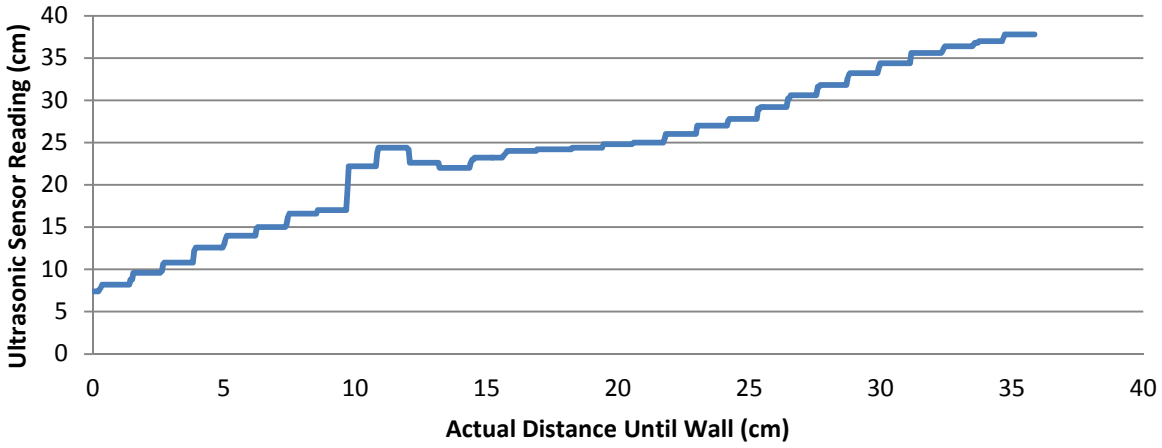

Figure 2: Ultrasonic Sensor Reading Error

Other sensors used were quite reliable, such as the LEGO light sensor, but simply insufficient. The robot's tasks required different types of information than those provided by the sensors used. In a playing field largely the same color, light sensors contributed little to the robot's knowledge of its location. The robot needed information to let it know, in as direct a manner as possible, of its orientation. The team attempted to use HiTechnic-provided compass sensors which take information with regards to the strength of the earth's magnetic field along three axes and output a heading, but this heading proved to be inaccurate and not tilt-compensated. A magnetometer, which returns the strengths of the three axes of the earth's magnetic field in milligauss, not a heading, was also tested, but its values were greatly altered by both hard and soft iron distortion.

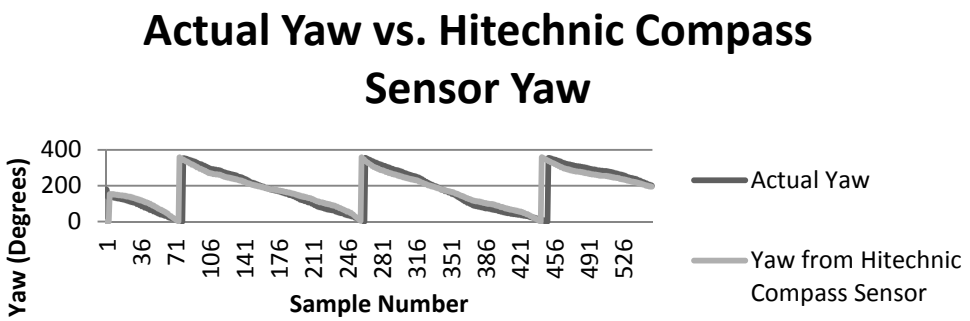

Figure 3: Graph of HiTechnic Compass Sensor Yaw Error

HiTechnic accelerometers, which measure the acceleration of the robot along three axes, were also evaluated in an attempt to get accurate roll, pitch, and yaw values. While the sensor was accurate while stationary, it exhibited errors of up to 50 degrees while moving, rendering it essentially useless. With the HiTechnic gyroscopes, which output a rotation rate in degrees around each of the three axes, there was found to be a slight drift over time that was difficult to compensate for as the offset was constantly changing. Although these sensors provided the correct information, they did so unreliably. The ultimate complication in achieving a known state is finding a precise, accurate sensor that provides relevant information.

3 Solutions to Software Quality Issues

3.1 Attitude and Heading Reference System

The magnetometer error was mitigated by taking the magnetometer values in a variety of orientations, and then calibrating the sensor to correct for the hard and soft iron distortion.
Although it was found that the accelerometer and gyroscope alone cannot give accurate attitude values, it is possible to use both sensors to achieve the desired results. This can be done by using a direction cosine matrix (DCM), a computationally efficient attitude estimation algorithm [10]. It integrates the gyro values in order to obtain an angle, but also detects any drift by the gyro and compensates for it using a PI feedback loop.

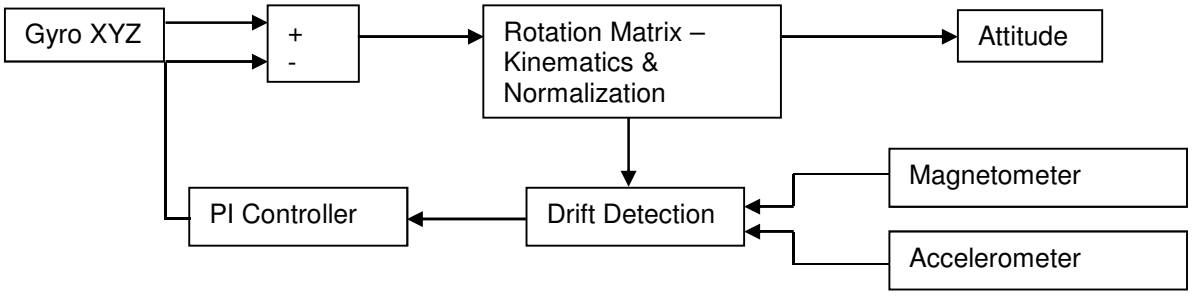

Figure 4: PI Feedback Loop

This consolidated system is called the attitude and heading reference system, or AHRS. It fuses 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer values to obtain readings of the robot's roll, pitch, and yaw.

3.2 Image Tracking Using Cameras

To localize itself, the robot may also be equipped with two cameras that feed visual data to an onboard processor which identifies landmarks and calculates the distances to them using a semi-global block matching (SGBM) algorithm.
Most low cost cameras today do not have perfect lenses, and do not have the same internal mounting, meaning that two camera images can have 2D offsets and distortion. In order for the SGBM algorithm to work, the two cameras need to be rectified so that their image outputs are aligned. The calibration was done using a chessboard, as the corners are easily recognizable. The algorithm first identifies the corners on the chess board, then uses those points to calculate and correct for distortion and offset.

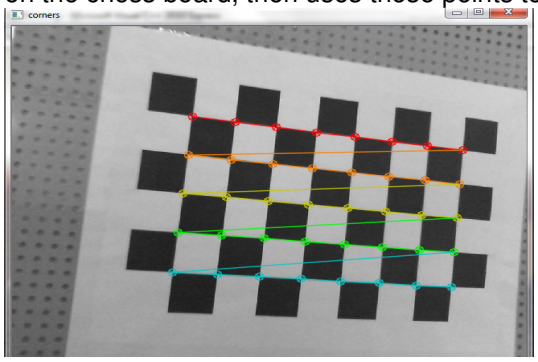

Figure 5: Chessboard Calibration of Camera

Using rectified image pairs from the left and right cameras, the SGBM algorithm finds a cluster of pixels from the right camera image and matches it to the most similar section of the left camera image. The SGBM algorithm then calculates the pixel disparity for those clusters and calculates the distance from the camera using the parallax effect; a closer object will have greater pixel disparity than a further object. Below is a pair of rectified camera images, and their corresponding disparity map.

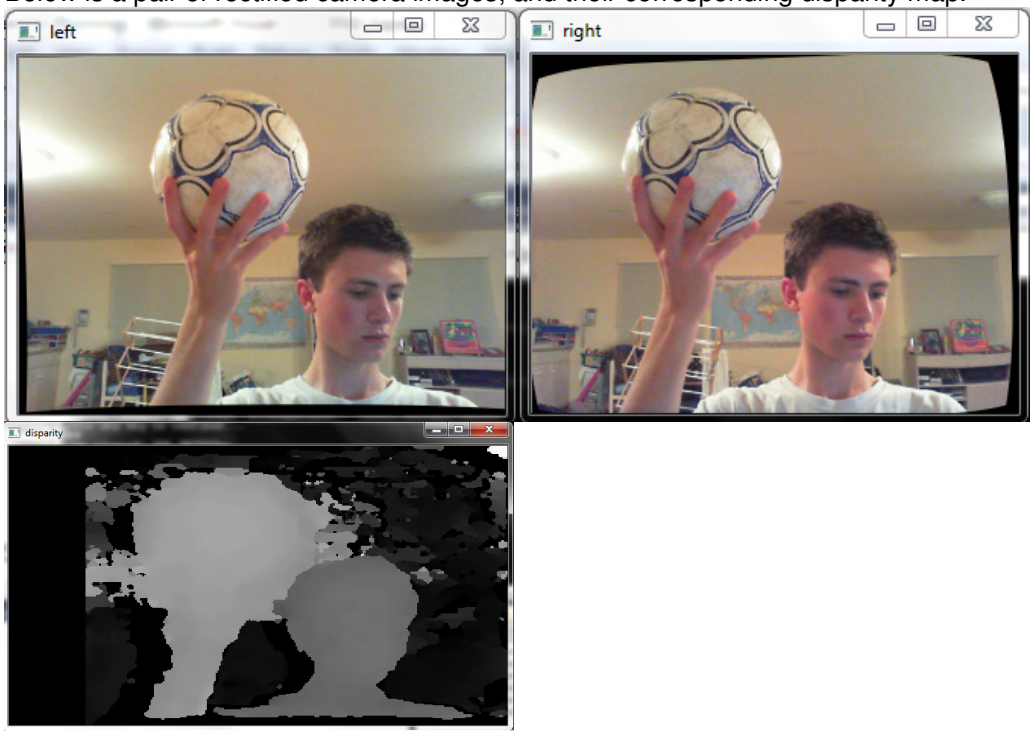

Figure 6: Calculated Disparity Map from Above Images

In order to calculate the distance from an object using the disparity map from the SGBM algorithm, one must first measure the distance between the two camera lenses. An object is then placed in front of the two cameras at a known distance, and the SGBM algorithm is run. The resulting disparity of the object is then used to calculate a calibration parameter that enables distance calculation.

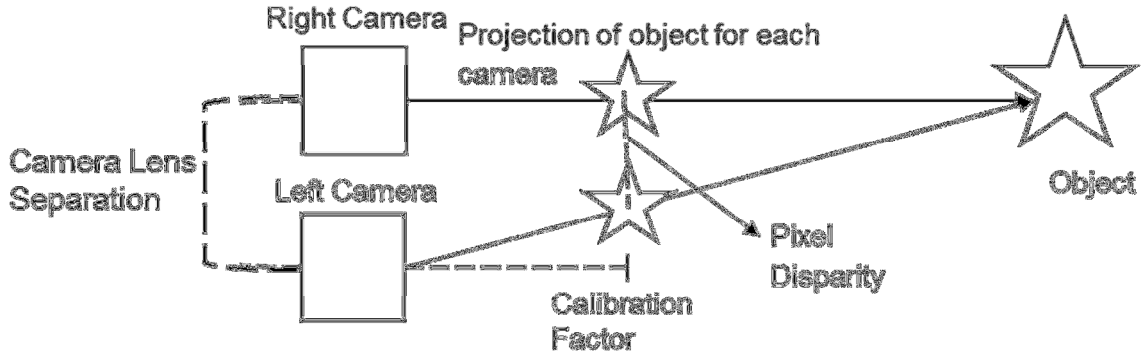

Figure 7: Distance Calibration Setup

4 Evaluating Efficiency of Solutions

4.1 Efficiency of Achieving a Known State Solution

The attitude and heading reference system was proven to be much more reliable than the HiTechnic Compass sensor. Due to the calibration, the AHRS returns an accurate, tilt-compensated heading. This can be seen in the graph below.

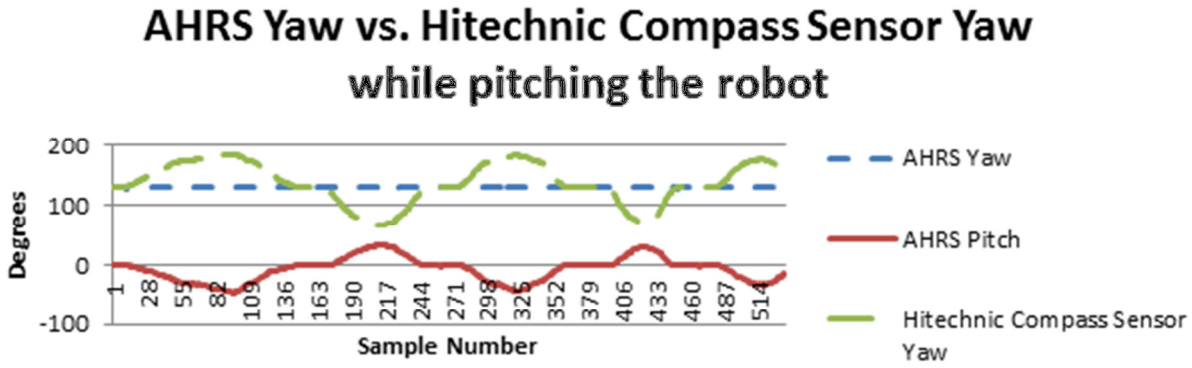

Figure 8: AHRS vs. HiTechnic Sensor Yaw

With accurate information with regards to its roll, pitch, and yaw, the robot is much more able to determine its position and orientation in the field. Without interference from hard or soft iron distortion, or misrepresented values due to acceleration, the process of truly knowing where our robot is position is much better facilitated.

5 Future Plans

With respect to the image-tracking algorithm used, there is still much work to be done. The team has yet to implement this solution on the robot and test it in real-time in a competition scenario. There will be much work to do in improving the efficiency of the system and fine-tuning the algorithm. The team plans to continue testing with this idea so that the efficiency of using cameras to localize the robot can be evaluated and further progress can be made.

While these endeavors with achieving higher levels of software quality have been fruitful, there is yet much work to be done. The efficiency of the proposed solutions has been evaluated, and there is much room for growth. Furthermore, there are many avenues leading to the known state and efficient movement which have not been discovered, and thus the team plans to explore these in times to come.

Acknowledgments

The authors wish to thank the FIRST Tech Challenge community, especially the Oregon Robotics Tournament and Outreach Program, for providing such a wonderful opportunity for high school students to learn about technology. The authors would also like to think Mr. Richard Vireday and Mr. Aaron Akzin for their invaluable suggestions to improve this manuscript.

References

[1] FIRST LEGO League, “Food Factor Challenge,” [Online], Available: http://firstlegoleague.org/sites/default/files/Challenge/FoodFactor/FLL2011_Complete_Challenge.pdf

[2] FIRST Tech Challenge, “Bowled Over! Challenge,” [Online], Available: http://www.usfirst.org/sites/default/files/uploadedFiles/Robotics_Programs/FTC/Game_Info/2011/Complete-Build-Guide.pdf

[3] FIRST Tech Challenge, “Bowled Over! Game Manual,” [Online], Available: http://www.usfirst.org/sites/default/files/uploadedFiles/Robotics_Programs/FTC/Game_Info/2011/Bowled-Over-Game-Manual_Rev%285%29.pdf

[4] RobotC, “RobotC NXT Curriculum,” [Online], Available: http://www.robotc.net/education/curriculum/nxt/

[5] V. M. B. K. Pirabakaran, “PID Autotuning Using Neural Networks and Model Reference Adaptive Control,” *Proceedings of the 15th IFAC World Congress, 2002*, 2002. [Online]. Available: http://www.nt.ntnu.no/users/skoge/prost/proceedings/ifac2002/data/content/01467/1467.pdf. [Accessed: 15-Jun-2012].

[6] W.-yong Han, J.-wook Han, and C.-goo Lee, “Development of a Self-tuning PID Controller based on Neural Network for Nonlinear Systems,” *Control*, pp. 979-988, 1999.

[7] F. Lin, R. D. Brandt, and G. Saikalis, “Self-tuning of PID controllers by adaptive interaction,” *Proceedings Of The American Control Conference*, vol. 5, no. June. American Autom. Control Council, pp. 3676-3681, 2000.

[8] D. Brutzman, “From virtual world to reality: designing an autonomous underwater robot,” 1992.

[9] Y. Kuroda, K. Aramaki, and T. Ura, “AUV test using real/virtual synthetic world,” *Proceedings of Symposium on Autonomous Underwater Vehicle Technology*, pp. 365-372.

[10] S. O. H. Madgwick, “An efficient orientation filter for inertial and inertial / magnetic sensor arrays,” Report xio and University of Bristol UK, vol. 2011, p. 1--32, 2010.

[11] Oliver J. Woodman, “An introduction to inertial navigation,” University of Cambridge, p. 1--37, 2007.