Critical Design Review

Ajay Joseph
Saba Rizvi
Steve Parkinson
Lee Fithian

TABLE OF CONTENTS

Problem Statement

We are to build a mobile robot to follow a pre-specified path using a synthesis of navigation methods. We will execute trial runs and record actual versus specified paths. We will analyze the results to determine the accuracy of the integrated system.

Objective

Construction: We plan to build a robot consisting of a chassis, power supply, servo motors, and a micro controller. We will use accelerometers, shaft encoders, a gyroscope and a compass as our movement sensors. We will write drivers for the robot using software compatible to the micro controller. We will interface between a personal computer and the robot using serial ports.

Experimentation: We will load a path for the robot to follow. We will layout a path on the experiment surface. We will then be able to measure the deviation in the paths.

Analysis: The experiment surface will be a grid. We will be able to measure the distance from the target for the different trials. We will then be able to make error distributions for the different navigation methods. From the error distributions, we will be able to create an integrated navigation system which takes into account the error inherent in each sensor. We will then run tests for this integrated system and find its error.

Algorithms

We have separated our program into three different functions, each of which has its own job.

Sensor-Interface Module – This module's job will be to obtain the data from the sensors and convert it so that it can be manipulated.

Navigation Module – This module will use the obtained data to calculate what the robot's next move should be. This is the module responsible for the navigation of the robot.

Movement Module – This module is responsible for moving the robot in the desired direction using the motor and wheels.

We have come up with Sensor Interface Modules for the accelerometer, gyroscope, and encoders. No matter what sensor we are using, we will have three outputs that the Sensor Interface Module will give to the Navigation Module. Those three output are:

1. Heading – the actual heading relative to the starting position.
2. X component – of the distance traveled
3. Y component – of the distance traveled

Accelerometer and Gyroscope (Sensor Interface Module)

The accelerometer measures acceleration on clock cycles using T1 and T2.

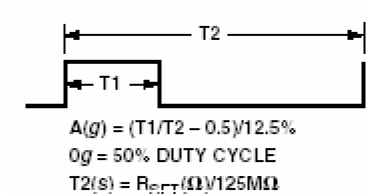


Figure 1: Duty Cycle of Accelerometer

The algorithm will repeat every time period, which will be set by us. A probable value is 5 ns.

The basic idea will be to reset the counter while the output is low.

General Formula for Position:

Position = Starting Position + $(A * t^2)/2$

Error Formula – To be used to predict error over time

Error(Position) = $(A * t^2)/2$

1. Sample axis counters x and y -> store count

c = counter value, f = frequency in Hz of clock, n = bits of counter

A 8-bit counter gives the following resolution assuming the nominal 12.5% duty cyle.

$$4/ (2^n * .5) \Rightarrow \text{resolution in g}$$

$$4/ (2^8 * .5) \Rightarrow .03125 \text{ g/bit resolution with 8-bit}$$

2. Calculate distance traveled since last sample

$$A = ((T1/T2 - .5))/12.5\%$$

$Dist_x = (A_x * t^2)/2$ (Formula for g based on $T1_x$, T2 is constant)

$Dist_y = (A_y * t^2)/2$ (Formula for g based on $T1_y$, T2 is constant)

3. Add distance to last distance since program started.

4. Sample value of Gyroscope

$$s_g = \text{sampled value}, n_g = \text{nominal value}$$

$$s_g - n_g / n_g *150 = \text{Angular Value} = V_{ang}$$

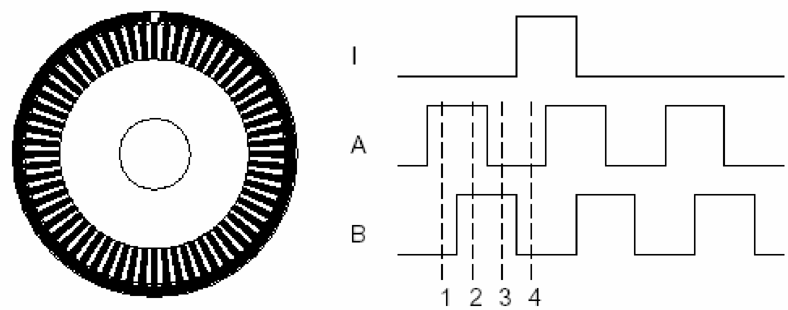
Angle traversed since last sample = $V_{ang} (t_{current} - t_{previous}) = \Delta\text{Angle}$

5. Add angle ΔAngle to Heading.

Shaft Encoders

Sensor output

The output of the shaft encoders is a sine wave. Depending on the resolution, a certain number of pulses are generated for each revolution. Our encoder will generate 256 pulses per revolution; half of this resolution is due to the 2 channels of output that are 90° out of phase. The shaft encoder has 5 pins, of which two are used to output the two sine waves.



State	Ch A	Ch B
S_1	High	Low
S_2	High	High
S_3	Low	High
S_4	Low	Low

Figure 2: Incremental Encoder Output

Connected to the shaft encoder, we will have an encoder to counter interface board. The purpose of this interface will be to provide a counter with a data that it will understand directly from the encoder. It will take as inputs the sine waves generated by the encoder. It will output either an up or down clock. The clock will be used to run the counter.

We will have a basic up/down counter to accumulate the number of pulses generated by the encoder. The input will be from the encoder/counter interface and provide the clock for the counter. The counter will then increment as the wheel rotates. When the counter is full, an interrupt will be generated and the information that the maximum value of the counter number of rotations have occurred.

From the output of the counter we will be able to determine the robot's position.

The position algorithm has been determined before and the equations are well known[1]. The basis for the algorithm is counting the number of revolutions and translating that into distance.

The data produced by the counter will be the number or fraction of wheel revolutions. When the counter is full, a certain amount of distance will have been traveled. This distance will be the length of the arc covered by one pulse multiplied by the number of pulses(the value from the counter).

We will then be able to determine the distance traveled by each encoder by mutiplying the total number of pulses by the length of the arc. Once we have the distance traveled by each encoder, we will be able to determine the total distance, D, traveled by the following equation:

$$D = (\text{Left D} + \text{Right D}) / 2$$

We will be able to determine the orientation of the robot using the left and right distances along with the robot's wheel base, *b*. The following equation calculates the orientation, Θ (in radians):

$$\Theta = (\text{Left D} - \text{Right D}) / b$$

Finally, we can determine the x and y coordinate positions. To do this, we must use the total distance, D, along with the orientation, Θ, of the robot. The following equations determine the position:

$$X = D * \cos(\Theta)$$

$$Y = D * \sin(\Theta)$$

These equations allow us to determine the current location of the robot.

From here navigation decisions can be made and executed. If the robot's heading is not towards the target, the robot can turn towards it and then move. To determine if the robot's

[1] http://www-personal.umich.edu/~johannb/shared/pos96rep.pdf.

heading is toward the target, we can compare the headings from subsequent measurements and correct for any deviations that occurred. This assumes that the robot is at some time, the robot's orientation is towards the target and that this is the basis for each heading correction.

Merging the data

Navigation with one or two sensors has been experimented and implemented many times over. We want to try and merge the sensor inputs we have into a single navigation system. The primary sensors that we want to merge are the accelerometer, encoders, and gyroscope.

Our first thought to merge the sensors is to do an average. Determine the estimated position based off of navigation algorithms for each sensor and then find the position in between the two estimations. This very basic scheme will not account for cases in which one sensor might be more accurate or one sensor has failed and is completely incorrect.

A more sophisticated approach would be to find a weight for each sensor and use them to find the position of the robot. We could run trials to determine the error each sensor-based navigation algorithm produces. We would then calculate the position based on each sensor. From these positions we could then weight them based upon their error. Then, we could sum the weighted positions to find an accurate position.

We could find out what kinds of errors are produced from each sensor. Then, we will know which measurements we can trust from each sensor (i.e. We can trust the distance traveled from one sensor, while another gives us a more accurate heading). Next we would determine how far off each measurement is for each sensor. We would then scale the measurements for each trusted measurements. Then we could have the different parts of the position calculated more accurately based off of which sensor we find to be more accurate and the scaling we would perform on it.

We have not developed these ideas as thoroughly as we intend. We only recently realized that this was the intended problem being researched and experimented upon. We are still doing research on these topics; we plan to elaborate and refine them as we proceed with the project.

Changes

Since the project proposal was submitted, there have been numerous changes made to the overall project. This was due to many factors. The team was given a relatively short amount of time in order to come up with a complete design of the project for the initial proposal. The project was also given many critiques by the advisors. Also, more research has been conducted since the proposal submission.

Acting on the suggestion of the project advisors, the robot will now have a compass and a gyroscope. These sensors will be added to aid in dead-reckoning and to help distinguish this robot from various other known projects. The compass chosen to be used was the CMPS03 Digital Compass made by Devantech. It has a resolution of 0.1 degrees with an accuracy of 3-4 degrees. The gyroscope chosen was the ADXRS300 from Analog Devices.

The processor on the robot will be upgraded to the OOPic II+. This provides for more objects and better computing power. This upgrade also brings the memory count up to 256 kb EEPROM, which will be more than enough for our purposes.

A prototype board will be added to the robot. This will allow the group to easily interface the sensors to the processor. Items such as the compass, accelerometer, gyroscope, and the counters for the encoders will be placed here.

The robot kit that was ordered arrives unassembled. Therefore, time will be added on to the schedule to achieve construction of the robot. Much more soldering will be required than originally anticipated.

Initially, normal alkaline batteries were going to be used for the robot. It was later found out that rechargeable batteries could be used. NiMH batteries have been chosen to be the batteries of choice.

In the proposal, the project was set out to perform two trials, one with each original sensor. One trial would be conducted with the encoders and the other would be performed with the accelerometer. The project advisors also proposed that there should be a third trial implemented which would include both of the aforementioned sensors at the same time. Each sensor algorithm would be weighted so that the most accurate reading of the robot's position and heading was to be calculated.

Due to time constraints and our parts not arriving, we changed around our Gantt chart.

	Week									
Task	First	Second	Third	Forth	Fifth	Sixth	Seventh	Eighth	Ninth	Tenth
Order Components	■									
Components Arrive					■					
Odometry algorithm	■	■	■							
Accelerometer algorithm	■	■	■							
Putting robot together					■	■				
Coding the OPIC					■	■				
CDR				■	■					
Debug						■	■			
Analysis								■	■	■

Table 1: Gantt Chart

Problems

Throughout the course of this project, many problems have arisen. This sections explains each one and how we have handled them.

One of the biggest problems the group had was the misunderstanding of the project's goals with the advisors. It was eventually decided that the robot will need to use all the sensor readings in the same trial to achieve the most accurate dead-reckoning system. The algorithm to make this happen would have to be weighted with respect to the level of error each sensor accumulated. The sensors to be used in the algorithm are the encoders, accelerometer, gyroscope, and the compass. Initially, each sensor was only going to be analyzed in its own separate trial.

After finally deciding on the encoder to use for the robot, a specific part number was to be determined based on the attributes required for the robot. The encoder part number required that an option for an index to be included. When submitting the part number to the ordering department, that option was not included. This caused the company in question, US Digital, to indicate that an incorrect part number was given and needed to be corrected. The part number was corrected and the order finally went thru.

It has already been 5 weeks since the project was assigned and we have no parts to work with other then batteries and the charger. This obviously will pressure the group for time in the future. Many aspects of the original schedule will have to be altered in order to accommodate for this delay.

After selecting the encoders to choose for the robot, the new problem of interfacing the encoders to the processor arose. We figured out that the encoder to counter interface board we had ordered was not the correct model, therefore we will have to reorder it. Counters would have to be used to in order to compute the number of pulses. An adapter was found to attach to the encoder which would allow for easy connection to the counter inputs on the prototype board. Four bit counters were selected and will be attained soon.

Since the encoders will be mounted in a custom manner, we need to study the robot in depth to determine how this will be accomplished. Since we do not have the robot yet, that part of the design must be held off until the parts actually arrive and the robot is fully constructed from the kit. A plexiglass frame is still the material of choice to be used so that the encoders can be attached on the outside of each wheel.

One problem that we ran into with the gyroscope is choosing what type of gyroscope to use. Analog Devices offers two types, an evaluation board and a ball-grid array. We were unsure what this meant, but after researching the two, we decided to go with the evaluation board. The ball-grid array board is basically designed with holes where the pins are. The board that you place this device on is supposed to already have pins where you just prop up the device with the pins inside the holes. The ball-grid is good if you are planning on replacing these devices on the board frequently. Since we are not going to be doing this and we want something permanent, we decided to go with the evaluation board.

We were confused by how to actually use each sensor together. At first, we were under the impression that we were supposed to run each sensor separately and then compare the results. We later realized that we have to integrate the two systems to produce a very good navigation system. We are going to integrate all the sensors together, using the accelerometer and encoders to calculate distances, gyroscope for angle and direction, and compass to verify at each point where we are.

When we ordered our counters, we were told that TI would take eight weeks to deliver. We decided on obtaining our counter by asking for one in the electrical engineering department.

The last time constraint that we ran into was time. We were not able to follow our Gantt chart, mainly due to the late arrival for our parts, but we worked on our algorithms and designed the layout of our board.

Budget

Since changes have been made to the design, the budget of the project has changed also. Most of the items in the original budget have not been changed. Here is a list of modified and additional parts not listed in the original budget.

Part	Supplier	Phase	Quantity	Price	Cost
Modified Parts					
Incremental Optical Shaft Encoder	US Digital	Construction	2	$ 66.00	$ 132.00
Encoder to Counter Interface Adapter	US Digital	Construction	2	$ 16.00	$ 32.00
New Parts					
OOPic II+ Upgrade Kit	Mark III Robot Store	Construction	1	$ 30.00	$ 30.00
Digital Compass	Mark III Robot Store	Construction	1	$ 38.00	$ 38.00
Gyroscope	Analog Devices	Construction	1	$ 33.00	$ 33.00