

**Dead Reckoning Group
Bi-weekly Report 2**

**Lee Fithian
Ajay Joseph
Steven Parkinson
Saba Rizvi**

Accomplishments

- Assembled Controller Board
 - Met with Josh to brush up on soldering
 - Soldered components to board
 - Booted up board – received the ‘green light’
- Servos
 - Modified for continuous rotation
 - Calibrated with software
 - Tested rotating servos (different speeds and directions)
- Assembled Chassis
 - Designed platform
 - Built platform
 - Used Plexiglas frame
 - Used metal chassis provided
 - Cut holes for wheels and encoders
 - Attached glider for rear support
 - Provided space for easy access for battery switches
- Compass Integration
 - Asked Marco for wrapping wire and tool
 - Connected to OOPic I/O pins
 - Used I2C connection
 - Tested and calibrated with sample software (did need calibration)
- Encoder Mounting – progress
 - Modified mounting brackets
 - Bought L-braces to attach to platform
 - Assembled encoder
 - Base Mount, Hub/Disk, Encoder Module, Cover
 - Attached encoder to wheel (See Problems)
- Parts Ordered
 - Ordered counters, muxes, and PCB from DigiKey
 - Ordered Parallax accelerometer evaluation board
 - Uses ADXL202 from Analog Devices
 - Attached to PCB
 - We provide the capacitors to determine T_1 and T_2
- Software
 - Downloaded and installed OOPic Multi-language 5.01 compiler on lab machine
 - Successfully compiled sample and team generated code
 - Successfully setup software to send data to OOPic via 9pin serial cable
- Data Integration
 - See attached paper

Problems / Solutions

- Ordered wrong encoder-counter interface

- This was because we didn't communicate about which output the design was being based upon.
- Lack of soldering experience
 - This kept us from jumping into the work and getting it done quickly.
- Lack of drill
 - The drill was/is essential to modifying the components that we have/will purchase.
- No software installed
 - We talked with Marco and put the OOPic software on a lab machine.
- Parts backordered
 - Accelerometer EB and OOPic sensor board were both backordered for several months. We looked into a socket for the accelerometer, but found no quick or easy solutions. We eventually were recommended to look at the Parallax accelerometer. This proved to be a solution. We ordered this part.
- Unstable software
 - The OOPic compiler is somewhat unstable. We often must tell the compiler to compile two times to get the code to compile.
 - Still learning VB. None of us has much experience with the language, so we have to look up syntax as we go. Luckily VB is relatively easy.

Goals

- Attach PCB
- Integrate accelerometer and gyroscope
- Final mount for encoders
- Write basic navigation algorithm for odometry and inertial to provide a testing basis
- Setup tests to develop values for data fusion algorithm

Data Integration

Goal: To show how we will integrate the data of our three sensors, the gyroscope, accelerometer, and encoders.

We will calculate our integrated data using variances of the error that each sensor produces. We will show this through an example.

Whichever sensor we are using, we want three values to be calculated from it. These three values are the x-distance traveled from last position, y-distance traveled from last position, and the angle relative to last position.

Finding the x distances and y distances:

To calculate the variance error, all three sensor will be tested first by making each sensor run, for example, two feet in one direction. Many trials will be done, and the x distances and y distances will be recorded each time. An example is below:

Lets say, the x distances of each sensor are as recorded:

Accelerometer: 1.5, 1.6, 1.7, 1.9, 1.5

Encoder: 1.6, 1.7, 1.8, 1.4, 1.9

The formula in summation notation for the variance in a population is

$$\sigma^2 = \frac{\sum(X - \mu)^2}{N}$$

Where σ^2 is the variance, μ is the mean, X is the individual data point, and N is the number of data points.

$$\text{Variance}_{\text{accel}} = ((.5 - .36)^2 + (.4 - .36)^2 + (.3 - .36)^2 + (.1 - .36)^2 + (.5 - .36)^2) / 5 = .0224$$

$$\text{Variance}_{\text{encod}} = ((.4 - .32)^2 + (.3 - .32)^2 + (.2 - .32)^2 + (.6 - .32)^2 + (.1 - .32)^2) / 5 = .0296$$

These numbers will be used to calculate weights in order to lessen error in our recorded results in the actual testing.

The final equation is:

$$\bar{X} \text{ position} = ((1 - (V_a)^{1/2})r_a + (1 - (V_e)^{1/2})r_e) / 2$$

Where the V's are the variances for the respective sensors, and the r's represent the recorded results of the three sensors after they have traveled for a certain time slot.

In our example, this equation would become:

$$\bar{X} \text{ position} = ((1 - (.0224)^{1/2})r_a + (1 - (.0296)^{1/2})r_e) / 2$$

The same method will be used to calculate our y distances.

The heading will be calculated with this simple equation:

$$\bar{\theta} = (\text{Angle}_{\text{accel}} + \text{Angle}_{\text{encoder}} + \text{Angle}_{\text{gyroscope}}) / 3$$

Where the Angle represents the angle recorded by each sensor.