LAB #3 SENSOR DATA CONVERSION AND CALIBRATION

OBJECTIVE

- Learn how to collect data from the SMART robot sensor interface board and display them with engineering units;
- Get familiar with range and inertial sensors;
- Learn how to perform linear sensor calibrations.

INTRODUCTION

I. SMART Sensor Interface Board

The Sensor Interface Board (SIB) is custom designed at WVU to expand the sensors onboard the SMART robot. The SIB is designed around a NetBurner MOD5213® embedded microprocessor which performs functions of data acquisition, data communication, time measurement, and if needed, control command generation. The additional sensors provided by the SIB include an Analog Devices ADIS16405BMLZ® Inertial Measurement Unit (IMU), six Maxbotix LV-EZ2 ultrasonic range finders, and two toggle switches. The IMU is a digital sensor pack that includes 3-axis accelerometers, 3-axis rate gyroscopes, 3-axis magnetometers, and an internal temperature sensor. The IMU communicates with the MOD5213 processor through a Serial Peripheral Interface (SPI). The ultrasonic range-finders are synchronized using a common trigger wire that is connected to a digital output pin on the MOD5213. The microprocessor triggers all rangefinders simultaneously every 1/20 second to minimize the cross interference among sensors. The range-finder data are collected with Analog to Digital (A/D) converters on the MOD5213. The toggle switches are connected to the digital input pins on the MOD5213 and their outputs can be used to trigger different custom software functions (e.g. stop or pause the MATLAB code from execution).

The SIB also powers three LED indicators. The red LED is used for power indication and the blue and green LEDs are controlled by the MOD5213 digital output pins.

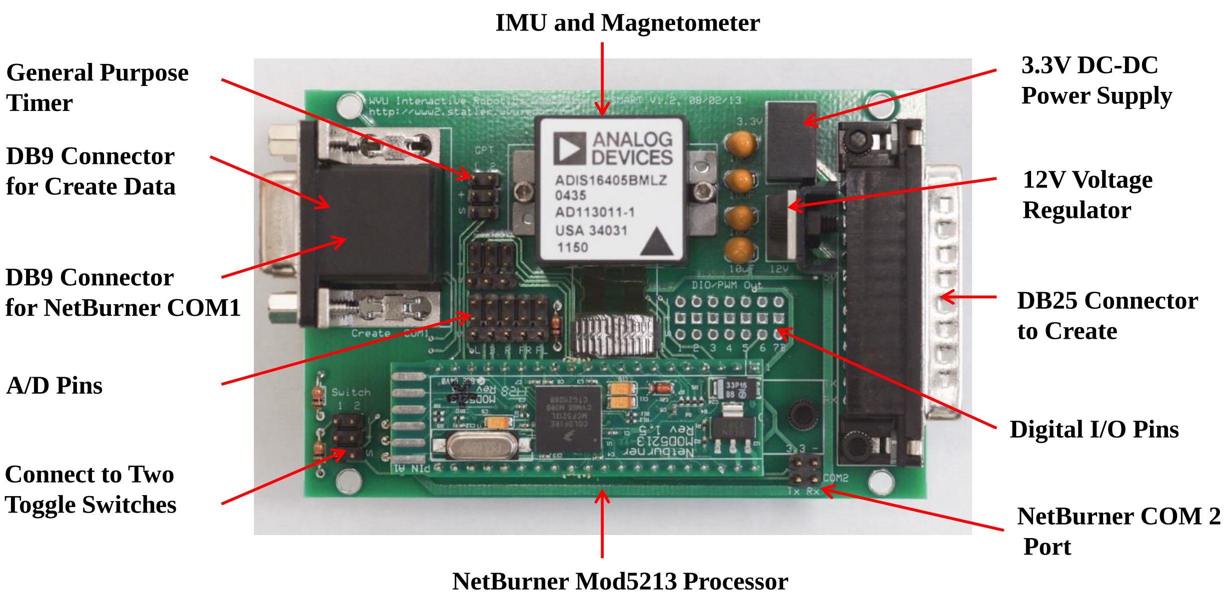

Fig.1. Layout of the SMART Sensor Interface Board

The SIB sends out data packets at a rate of 50 Hz with a baud rate of 115,200 bps. Each packet is 50 bytes long which include two synchronization header bytes, a 8-bit counter, a checksum byte, and 46 data bytes. The format of the SIB communication packet is shown in Table #1.

Byte	Content
0	Header synchronization byte #1, (65_{10}, ASCII "A")
1	Header synchronization byte #2, (90_{10}, ASCII "Z")
2	8-bit Counter (0-255)
3, 4	IMU Acceleration x-axis
5, 6	IMU Acceleration y-axis
7, 8	IMU Acceleration z-axis
9, 10	IMU Angular Rate, p, (roll rate, rotation around the x axis)
11, 12	IMU Angular Rate, q, (pitch rate, rotation around the y axis)
13, 14	IMU Angular Rate, r, (raw rate, rotation around the z axis)
15, 16	IMU Magnetic x-axis
17, 18	IMU Magnetic y-axis
19, 20	IMU Magnetic z-axis
21, 22	IMU Temperature
23, 24	Reserved A/D channel #1
25, 26	Reserved A/D channel #2
27, 28	Front rangefinder
29, 30	Front-Left rangefinder
31, 32	Left rangefinder
33, 34	Back rangefinder
35, 36	Right rangefinder
37, 38	Front-Right rangefinder
39	LSB for switch #1, LSB+1 for switch #2
40- 48	Reserved. Current values are 67-75 (ASCII 'C'-'K')
49	Checksum

Table.1. SMART SIB Communication Packet

For a 16-bit data channel, the upper byte is transmitted first, then the lower byte: Int16_Value = Upper_Byte *256 + Lower_Byte.

II. Convert Raw Data to Engineering Units

The values coming out of the SIB are raw sensor data, which need to be converted to engineering units before they can be used for other purposes. All inertial sensor outputs are 14 bits (a total of 16385 values) in length and are in two's complement format. These values needs to be converted to 'double precision' formats with a range of values between ±18g for accelerometers, ±300°/s for gyroscopes, and ±2.5 gauss for magnetometers. From the sensor spec sheet, we can find that the typical sensitivity for the accelerometer is 3.33mg/LSB. LSB here stands for the least significant bit, which is corresponding to the smallest change that an A/D can detect. The typical sensitivity for the gyroscope is 0.05°/s/LSB and the typical sensitivity for the magnetometer is 0.5mgauss/LSB. From these values, you should be able to come up with equations that can convert raw inertial data to meaningful measurements. A quick way to verify if you converted the data correctly is to check the accelerometer outputs. If the robot is in static condition, the total acceleration measurement (combining all three axes) should be near 1g.

For the range finders, the situation is a little different. There are two steps involved in the conversion process. First, the sensor output is an analog signal in voltage, which is then digitized with a 12 bit (4096 steps, 0-4095) A/D on the MOD5213. These steps are corresponding to a voltage range of 0-3.3V. So in this first step of the conversion process you will be converting the digital codes back to voltage values. From the sensor specification sheet, we can find that at 3.3 V voltage input (which is the one we use), the sensitivity of the sensor is 6.4mV/in. So in the second step, you will then convert the voltage value to distances. Keep in mind that there is no negative range measurement in this case. We will be verifying this conversion result with a series of calibration steps later.

III. Calibration of Range Sensors

Please see the class handout for detailed steps for two-point and multi-point linear calibrations.

IV. Warnings

1. Do not keep the Create powered on unattended. The robot may drive itself off the table. Only turn on the robot power when you are ready for a test and turn it off right away when you are done with the test.
2. Flip the laptop screen up when you are driving the robot. Do not use the laptop as a bumper!
3. If your laptop needs an update when restarting, let it finish. Do not turn off the laptop in the middle of an update.

PROCEDURE

I. Collect Data from the SMART Sensor Interface Board

1. Write a code that can collect the raw SIB sensor data from 'COM2' for 1000 steps. Keep in mind that SIB sends out data at a fast rate (50Hz). If you are collecting data at a lower rate (e.g. 10Hz), some data will be lost in the buffer. To prevent getting really old data, you can use the 'flushinput' function to clean up the serial buffer every once for a while;
2. Plot each channel of the raw data against time; Check if these values are responding to your inputs (i.e. movement of the robot);
3. Check if you can also read the two toggle switches;
4. Now, convert the raw inertial and range data to engineering units (use Metric units) and plot them again;
5. Verify if the conversion is correct by checking the total acceleration measurement (it should be around 1g when there is no robot movement);
6. Show the instructor your results so far.

II. Calibrate the Range Sensors

1. Pick a range finder on the robot to calibrate and use the floor tile as your reference (each floor tile in the G22 lab is approximately 1×1 foot and one foot is approximately 305mm); Use a hard-covered book or some other flat surface to reflect the ultrasound;
2. Collect 500 points of data at every two feet up to 10 feet in distance;
3. Visually inspect each data set, remove outliers, and calculate the average of the remaining measurements; Use the average values in the following calibration steps;
4. Perform a two-point calibration by hand use the measurement at 2 feet and 10 feet;
5. Perform a multi-point calibration use the MATLAB 'polyfit' function;
6. Implement the calibration equation in your data conversion code;

7. Verify the performance of your calibration by performing another set of measurements and log the errors;
8. Repeat the calibration process with another range finder and compare the calibration parameters;
7. Show the instructor your results.

DELIVERABLE

An abbreviated lab report of your experiments and answer the following questions:

1. From the data you collected, which directions are the robot x, y, and z axis?
2. What's the total strength of the magnetic field in the G22 lab? Does it match the value provided by NOAA? http://www.ngdc.noaa.gov/geomag/magfield.shtml
3. Is the output of the range finder reasonably linear?
4. What's the difference in results between the two-point calibration and multi-point calibration, and why?
5. What is the maximum error you experienced using the linear calibration method? How did you come up with this number?
6. Are the two range finders similar in terms of calibration parameters?