**EA-467 ATTITUDE ESTIMATION LAB** (rev-b) Fall 2017

In the middle of 2016, Ire-wrote the sun sensor to have separate coefficients for +/-X and +/- Y. Instead of the overall magnitude and a center variable. This was BAD, since the Thruster lab still used the former. We should go back to that, or re-write the Thruster SunCal routine to match this one. Probably best…

**Introduction:**

The first step to controlling the attitude of your spacecraft, is estimating what it is. Several sensors are used for estimating the attitude of a spacecraft including Earth and Sun sensors, magnetometers, star sensors and gyroscopes. In this lab you will develop the means to read your spacecraft’s magnetometer and Sun sensor in order to estimate the attitude angle and rate around the Z axis. In the next control lab this attitude angle estimation will be used to implement a control algorithm to despin your spacecraft and to point towards the Sun. Your flight code for the next lab will be simple, something like this:

**INITIALIZE:** Declare and initialize all variables

**SunCal** Light green LED. Find Maximum sun value for all 6 sensors and calibrate

**DeSpin** Light orange LED. Add a Bdot or Sdot algorithm to reduce rates to near zero

**SunAlign** Light red LED. Point spacecraft to Sun. Flash LED when aligned.

**LABsat Coordinate Frame:**

The LABsat coordinate frame is shown to the right (z axis is up). Put your compass rose on your right workstation with the back edge parallel with the edge of the table and make sure your table is parallel with the lines of the room. Check the Nautical Chart at the front of the class to determine the azimuth of the front of the room using the parallel ruler to the seawall and thus, to the building. Then point the compass rose to 131° straight ahead at the blackboard.

**Part D: Magnetometer – Measuring the Earth’s Magnetic Field:**

A spacecraft uses a magnetometer to measure the Earth’s magnetic field vector. The magnetometer outputs a raw sensor count of the Earth’s magnetic field in X,Y, and Z. Each axis is a different individual sensor and so each one has to be separately calibrated for consistency and then the result scaled to meaningful values. We can use all three axes though hanging on a string or using the compass rose will only use the X and Y axes.

The Magnetometer talks to the CPU over a binary serial (SPI) bus with send/receive on pins 1-6 (pins 13 to 8 on the Arduino). There are two pins for send/receive, a serial clock, a device select pin, a reset pin, and a ready pin. Two other pins on the other side provide +5V and ground.

**Software**: Download the **ADCS\_Estimation\_Code\_Part\_D** program on the workstation from Blackboard. Load it into the Arduino editor and **resave immediately on your own drive**, and notice that the program initializes the variables and then in Setup, initializes the serial port and sets the pinModes. Then the main loop simply reads each magnetometer axis and then Serial.prints the result in raw XYZ results. Upload into the Arduino and observe the raw readings. Next, you will calibrate your magnetometer.

**SPI Bus:** The Magnetometer uses an SPI serial interface to the Arduino. An SPI bus allows multiple ***slave*** devices to share a common bus with the ***master*** CPU. The common bus has a Clock, and separate Transmit (MOSI) and Receive (MISO) lines as well as a chip-select pin (SSNOT). The Magnetometer also has a device Ready flag bit when it is ready to provide data. You should understand how an SPI bus operates. Another common such serial interface to other chips is the I-squared-C bus called I2C.

**Labsat Setup:** Set your compass-rose base on your right workstation and be sure to align the base with the edge of the table so that the +X direction of the LABsat (and pointer) is pointing at 311 degrees when +X is pointing to the back of the room (or 131 if pointed at the front of the room). This orients your compass rose with true North. A piece of tape marked CAL holds the compass rose in place. But remember the actual magnetic pole is down about 70 degrees below the horizontal in the North direction.

**Main Loop:** The main loop simply reads each of the X, Y and Z axis values from the Magnetometer and then displays the result.. It does this with three consecutive calls to the ***readaxis()*** function.

**Magnetometer Period Adjust:** The magnetometer output count is proportional to the period of an internal oscillator as affected by the magnetic field. We must set a period (prd) variable (1-7) to scale this count to the desired order of magnitude. To set this coarse calibration, in the code you will notice the prd for each axis is passed to the ***readAxis(axis,prd)*** Function for each axis call, but we default that to 6. Each increment in the **prd** variable adjusts the magnetometer output scale by a factor of two. Start by rotating your spacecraft in 3D to confirm the maximum values on each axis are within a factor of two of |1000| (between |500| to |1500| or so). If not, adjust the **prd** variable up to 7 or down to 5 to get within a factor of two of |1000|. In the next section you will adjust the fine cal factors to make these maximums very close to |1000| in each axis to give us tenths of a percent accuracy.

**Run the program.**  Pickup your labsat carefully so you can rotate in 3 dimensions to get your results. Remember, here in Annapolis, the actual magnetic north pole is as much Down as it is North… Orient each axis pointing down and North to find the maximum value of each axis, including negative axes.

**Calibration**: To do a finer calibration than is possible with **prd**, notice the code declares and initializes floating point **xcal**, **ycal** and **zcal** factors for each axis to 1.0. You will adjust these to obtain final scale readings closer to -1000 to +1000. Run the program, holding the satellite at maximums on each axis and adjusting each xcal, ycal and zcal factor each time until you get a maximum count between -1000 and +1000. Hopefully the + and – maxes are reasonably symmetrical so we will not plan to account for them separately. Note: record and keep the exact values of all of the calibration coefficients and prd values and your labsat number for future labs.

**Field Measurement:** For better experimental precision we will use the compass rose, but now the magnetometer is constrained in the local horizontal plane and will not see these 3D maximums. Place the satellite on the compass rose so that the X+ arrow is aligned with the pointer. Rotate the LABsat to maximize +X (and minimize Y). ***Note the angle on the compass rose. This is pointing to magnetic north.\_\_\_\_\_\_.*** Notice how the +X maximum is broad but the Y minimum is critical to find. Next, rotate the LABsat to align the X direction with geographic **True** north (0) on the compass rose. This will place the Y direction West, and the Z direction up. The X,Y and Z values in this orientation should represent the Earth’s relative magnetic field vector at that location. ***Record the Earth’s magnetic field vector observed by your satellite in this orientation.\_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_.***

The XYZ magnetic field vector (in Tesla) in Annapolis from a model of the Earth’s magnetic field is shown at right. The Y value would be zero if Magnetic North and True North were the same, so Y is a small value representative of the magnetic variation between North and magnetic North in this area and being a small number near zero, will vary significantly in the presence of metals, cabinets, and desks and your location in the room.

What is your Vector? \_\_\_\_\_, \_\_\_\_\_, and \_\_\_\_\_. Compare the magnitude of your X and Z values for comparison to the model’s. Since your values are relative and not calibrated in absolute Teslas, simply scale your Z value to -47e-6 and apply the same scale factor to your X and then see how they compare to the model’s value. What is your scaled vector? \_\_\_\_, \_\_\_\_, and \_\_\_\_\_. Compare your reading with the other magnetometer LABsat readings from the other teams.

**Instrument Comparison:** Now borrow the lab’s magnetometer instrument, align it with true north and compare its readings. \_\_\_\_, \_\_\_\_, and \_\_\_\_. Comment on differences.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Nearby Effects:**  Borrow a tool, a NiCd cell, a radio and a magnet from the instructor and see its effect on the readings. (KEEP THE MAGNET AT LEAST A FOOT AWAY so as to not saturate the sensor Also, these rare-earth magnets are very brittle. DO not let them CLICK HARD to hard metals). How far away does each have to be to eliminate any effect on the readings? \_\_\_\_, \_\_\_\_, \_\_\_\_, \_\_\_\_. What does this imply about satellite design?

**Linearization Test:** To see how accurately your magnetic vector estimates the spacecraft attitude, there is a line of code to convert the X and Y coordinates to an angle in the X-Y plane using the ARCTAN2 function. Un-comment the code. The syntax is

 double AZ = atan2(y,x) and then we convert to degrees by multiplying by 180/pi

The next line converts the angle from +/-180 to a convenient nautical 0-360 degrees. Once you have this working and displaying the angle A, rotate the spacecraft in a full circle and confirm that A goes from 0 to 359 and back to 0.

Now run a comparison test on the compass rose. Align the compass disk 0 to the spacecraft pointer at the AZ=000 reading (magnetic North). Rotate it in 10° increments and write down the reported AZ, and the physical angle from the compass rose. Post lab, plot AZ against the physical compass rose angle and also a line showing the perfect result. Next plot the difference between the AZ reading and the expected perfect result to better show the errors and non uniformity of the field as measured in your system.

**Part E: Sun Sensing Attitude Estimation:**

Your coarse sun sensor uses the same photoresistors from the telemetry lab. Five sensors are assembled on a 10 pin module plugged into the Analog inputs on the Arduino board so it can see in all directions (not down). The figure shows how we can combine sensors from the opposite sides as shown here for Y to give a single +/- output per axis.

Connect your LABsat USB port to your PC for observing the telemetry. Download the program **SunSenseArduino\_Part\_E2017** from Blackboard *and* ***resave to a new filename on your drive*.** To reference your compass rose to the sun and not true North, lift the “cal” tape and rotate the compass-rose disk to align 270 degrees to the base red calibration arrow. This puts 000 Azimuth directly at the lamp. The lamp should be 30 inches away.

**Raw Sun Sensors**: As in the telemetry lab, the **analog.read** statement is used to read the A/D counts of the 5 axis sensors (-Z faces down and is assumed always in the dark (0)). The analog input is proportional to the sun magnitude and we can multiply it by a calibration factor to give a scaled and approximately proportional count for each coordinate direction. To verify your 5 sun sensors, place your Labsat on the compass rose about 30 inches from your lamp. Take care that there are no other sources of light or reflections near your satellite including your hands and body (midshipman in whites) and that the room lights are off and shades down. The first block of code in the loop function reads the counts on each axis and displays it. With the cal factors all at 1.0, turn on your lamp and rotate the LABsat to see the raw count maximums on each sensor +X\_\_\_\_, -X\_\_\_\_, +Y\_\_\_\_, -Y\_\_\_\_, +Z\_\_\_\_.

**Eclipse:** Turn off the lamp. Cup your hands over the sensors and see the smallest values, from \_\_\_\_ minimum to \_\_\_\_. This is analogous to free space. Now remove your hands so that ambient room light is on the sensors and you can see that there is a strong albedo ranging from a \_\_\_\_ minimum, to a \_\_\_\_\_ maximum. This albedo will have an effect on your insitu results when you combine the sensors.

**Calibrate:** As with the magnetometer, align each axis towards the lamp and adjust your scale factors so that you get maximums near 1000 on each. This is affected by nearby reflections, so be consistent with the surroundings in your measurements. Record your calibration factors for use in future parts of the ADCS lab series with this particular LABsat: LABSAT# \_\_\_\_\_, Xpcal \_\_\_\_, Ypcal \_\_\_\_\_, Zpcal \_\_\_\_\_, Xncal \_\_\_\_\_, Yncal\_\_\_\_\_. (Note: you will need these in furure labs for this LABsat)

Example code for applying the cal factor to each sensor: Xs = Xpcal \*Xp

**Combine Into Vector Format:** Now ***comment-out*** the first block of **Serial.prints** portion of the original code and ***un-comment*** the next block which will print the algebraically combined sensors on each axis to give single Xs, Ys and Zs Sun vector components. This is done by subtracting the negative sensor magnitude from the positive sensor magnitude for each axis. As a result, you will get a positive component when the sun is on one side and a negative count when the sun is on the opposite side.

 Example code for each axis: Xs = Xp - Xn

**Angle Determination:** Next, we will measure the sun sensor vector performance using the compass rose and ARCTAN2 function to compute the rotation angle (AZimuth) from the X and Y components. In the example figure shown here, the +X panel is mostly illuminated head-on and a smaller amount of sun is on the +Y panel. Since the sun sensor output approximately relates to the magnitude of the sun on each panel, the angle  in each axis is simply the arctangent of the ratio of the +X and +Y values.

The ATAN function outputs in radians in only the first quadrant. The function called ATAN2 outputs in radians for signed values of X and Y and gives correctly signed values from 0 to +/- 180 in radians. We correct this value to give the angle AZ in 0-360 degrees as in Part E to match the compass rose.

**SunSensor Performance Testing:** As with the magnetometer, assess the performance of the sun sensor by comparing the output angle (A) to the actual angle (compass rose). Align your LABsat so the +X face is pointed at the sun lamp and your compass rose pointer is exactly on 0. Now rotate your compass rose and model through the 360 degrees in 10 degree increments clockwise, pausing on each angle where your partner will enter the actual compass angle and the LABSat-reported azimuth into Excel (or Matlab).

**Post lab**: Generate a scatter plot to show the reported AZ angle against the actual compass angles for both the magnetometer and coarse Sun sensor. Also plot the compass angle against itself which will be a straight line showing the output of a perfect sensor. How much your sensor output deviates from this is the error in the system. To help show the errors, add an error column showing the percent error and choose an appropriate scaling factor so that it shows up well on the chart too. Does your system estimate the angle of the spacecraft? What is the magnitude of the errors and where do they occur? Can you completely resolve your inertial attitude with only one vector source? Why or why not?

**SAVE all your CAL factors and your LABSAT number!**