Center of Gravity Device: Additional Design Enhancements and Embedded Software

July 29, 2014

Author
Alex Lerikos
alerikos@gmail.com

I have made two additions to the design of my center of gravity device. These additions address a few of the improvements I discussed in my initial report. I have added a adjustable mount to the center of gravity device that allows for more secure mounting and allow for CubeSats of multiple widths to be compatible with my device. I have also written a sample program for the ATMega8 microcontroller using the Arduino framework. This program performs the calculations and measurements of the position of the center of gravity on all three principal axes and utilizes system interrupts to allow the operator time to position the satellite.

Mechanical Design Addition: Adjustable Mount

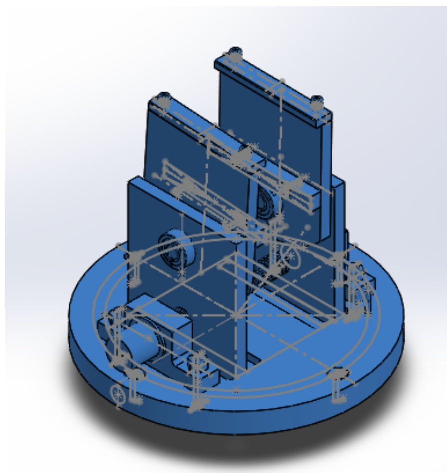

Figure 1: Satellite Mount Assembly

A potential issue with my initial design was the lack of mount clamp adjustability. Previously, only CubeSats with identical widths to the 3U CubeSat could be mounted into my device could be measured. Additionally, the width of my previous mount clamp was fixed and therefore could not account for slight variations in the width of each satellite. With my new mount redesign, both of these issues are addressed.

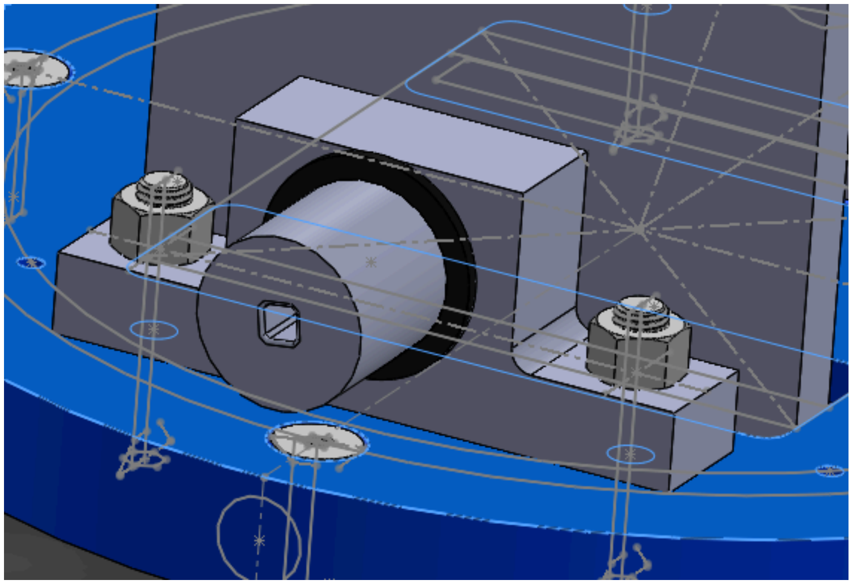

Figure 2: Shaft and Shaft Mount

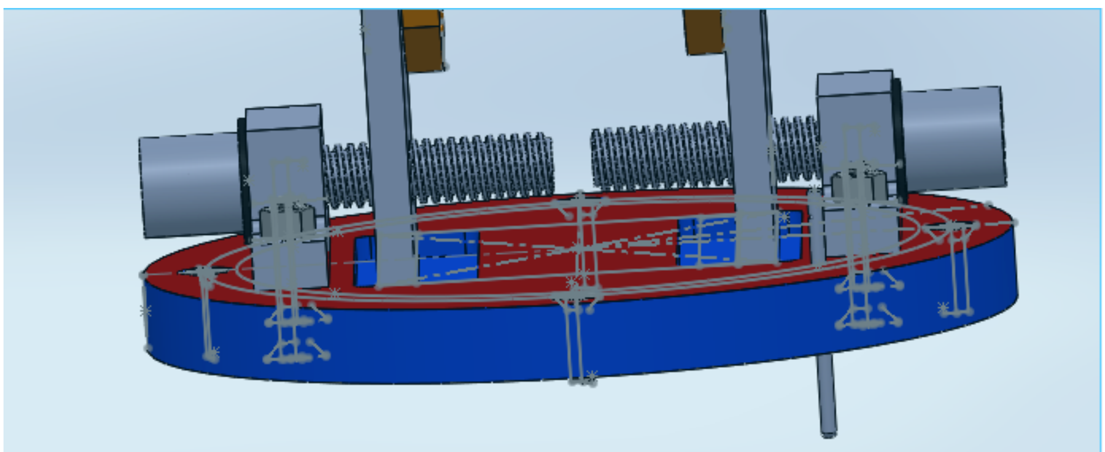

Figure 3: Side Clamping Plates and Shaft orientation

My redesign implements a vise-based clamping system. As shown in Figure 2 and 3, two shaft mounts are attached to the base straddling the geometric centerline of the bottom plate (red) using two mounts. Each shaft is threaded with an ACME trapezoidal thread form with a nominal diameter of 1 inch. Both shaft mounts are not threaded but the Side Clamping plates are. Each plate is threaded such that when the each shaft is turned clockwise, the plates will move toward each other. The plates will move away from each other when each shaft is turned counterclockwise.

Two channels with equal distance to the centerline of the bottom plate have been cut into the plate to insure the alignment of the Side Clamping Plates and to assist in installation of the shafts. During the shaft installation, each plate is oriented such that their inside face is against the inner wall of the channel. Each plate is therefore guaranteed to be of equal distance to the centerline as the shafts are threaded. The bottom of each channel will need to be lubricated to insure smooth moment.

A 3/16-inch square chuck wrench can be used to adjust the mount. To insure that the Side Mounting Plates are both equal distant from the center, the device operator will need to measure the distance from the bottom plate's centerline to the inside face of each Side Clamping plate and adjust the plates accordingly. This is only critical however for the measurement of one of the center of gravity distance measurements and can be accounted for if the operator notes the offset that the Side mounting Plates are from the centerline of the bottom plate.

Software Addition: Interpreting the Torque Sensor Output

I have included a sample program to be used with an ATMega8 microcontroller using the Arduino framework. My program takes the voltage output of the torque sensor as input and performs the calculations and measurements of the distances to the center of gravity on all three principal axes. The calculations and order of operation execution that this program implements to determine the distances are described in my previously submitted design document and revision. This program assumes that the voltage output of the torque sensor is amplified. Please see the "**Additional Torque Sensor Analysis"** section at the end of this report for a description of the importance of input voltage amplification.

My program utilizes a single interrupt that is triggered with the rising edge of a discrete voltage change at the interrupt pin. This interrupt allows the operator time to move and position the satellite as well as record a tare-value for the force sensor output without the program continuing to execute. This interrupt could be in the form of a switch or button.

The program calculates the torque using voltage calibration values that must be previously determined. These values include two known torques and their recorded torque sensor voltage output. Using these two output voltages and the known torques, a linear relationship can be defined to interpolate torque readings for additional sensor output voltages.

The program makes three measurements, each with two stages. When the operator starts the program, it waits for the operator to signal the first interrupt. It then records the tare force sensor value and waits again for the next interrupt. The operator then positions the satellite and signals the interrupt again. The program then takes five torque sensor output samples and averages them. It then takes the average and performs the calculations needed to determine the position of the center of gravity with respect the axis being measured. It then waits for the operator to signal the interrupt again before repeating the process on the next axis to be measured.

This program outputs information to the Serial Consol. However, it can be easily adapted to output to a display depending on the type of display used in the device. It also currently is designed to handle interrupts for strictly the ATMega8 microcontroller but it can be adapted for other microcontrollers by changing the handling of the interrupts in the setup function and changing the interrupt pin macros.

Additional Torque Sensor Analysis

My program assumes the torque sensor voltage output is amplified. Amplification is important, as the unamplified voltage signal is usually too small to be accurately read.

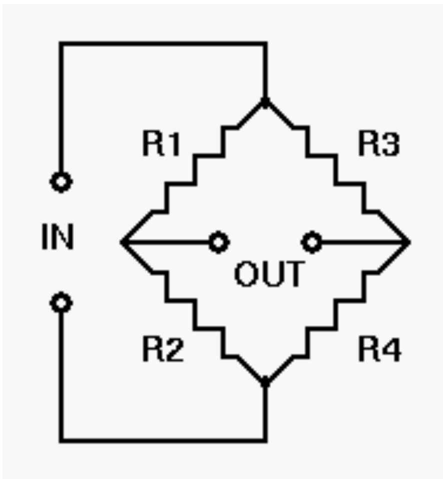

Figure 4: Wheatstone Bridge[1]

Static torque sensors based on strain gauges and resistors. The strain gauges and resistors are laid in a Wheatstone Bridge configuration (see figure 4). One or more of these resistors are strain gauges that change resistance as they are laterally elongated. Because this change in resistance is typically too small to measure, a Wheatstone Bridge is used to measure the difference in voltage across the bridge as the strain gauges are strained.

The voltage difference across the bridge is typically very small and must be amplified to insure an accurate reading. Instrumentation amplifiers (such as the Texas Instruments INA126 amplifier) are most commonly used for to amplify the signal. These amplifiers are superior to operational amplifiers as their gain can be adjusted using an externally added resistor.

[1] Reference: http://www.play-hookey.com/dc_theory/resistors/wheatstone_bridge.html

Only after the voltage output signal is amplified can it be used to measure the torque. Some torque sensors have built in amplification systems, which is a feature to consider when selecting a torque sensor.

References

[1] Arduino Reference Library, http://arduino.cc/en/Reference/HomePage: July 30,2014.

[2] http://edg.uchicago.edu/tutorials/load_cell/: University of Chicago, July 30,2014.

[3] Handling external Interrupts with Arduino http://gonium.net/md/2006/12/20/handling-external-interrupts-with-arduino/: gonium, July 30,2014.

[4] Acme - The Transversal Motion Thread, https://www.atimetals.com/products/Documents/TC_AcmeThreadforms.pdf: ATI Metals, July 30,2014.

[5] Valeria La Saponara, Walt Harris, Dave Horsley , Lecture 5: strain gauges and Wheatstone bridges , https://smartsite.ucdavis.edu/access/content/group/41ff5981-aa91-4bcb-921e-81fde35c2964/EME107BS14_Lecture5.pdf: University of California-Davis, July 30,2014.