DESIGN DOCUMENT

Measuring 6DOF platform accuracy

System Overview:

In this project we’ll Check that the chair moves exactly in the specific direction according to commanded values. For this we will measure the dynamic orientation of chair at every 2s. We measure the translational motion of chair up to nearest 1 millimeter and angle up to ½ degree of accuracy.

Software design:

the software consist of the two systems a Sensor server and monitor module.

Sensor Server;

The sensor server responsible for retrieving data from the distance sensor server and angle sever. It sends the combine data to monitor .

The format of the data is as it folows:

H xxxx, xxxx, xxxx, xxxx, xxxx, xxxx, yyyy, yyyy, yyyy, /n

Where the xxxx represents ASCI digits for the actuator lengths and the yyyy represents floating point numbers for the angles separating by commas and ending with \n trailer.

The first thing that the code dose is to clear the strings and waits for character and if the character is a H then it read the distance data until it receive /n otherwise it waits until it receives H . once it receives /n it returns sensor data . after that it combines with angular data and send it to monitor.

After importing these values in Monitor, it will then calculate the error by comparing the values with commanded values.

Distance sensor

The Distance sensor server reads the sensor values using The Mega board . the mega reads the six sensor distance values using softwareSerial.

The format of sending the data is as it follows:

H xxxx, xxxx, xxxx, xxxx, xxxx, xxxx, \n

The format starting with H, six length values separated by comma, and ending with \n trailer and Send this to Uno on the specified port. XXXX repesants intiger values between 300mm and 1000mm. a value of zero will be sent if sensor not connected.

The first thing that the code dose is to clear the strings and it adds a H as header. Then it reads the six sensor values and send it to the angle server . . if a sensor is not connected then it will send a 0 .

Angle server

Angle server reads the 3 angular values using the Uno . In Uno we have a function called get sensors

## Hardesign:

We are measuring length of 6 actuator using Maxbotic sonar sensor MB1043 HRLV-MaxSonar EZ4. 1 sensor is attached to each actuator in such a way that each sensor and its target is attached at both end of actuator. Sensors are capable of reading 30cm to 5000cm accurately, precision up to the nearest 1mm on their serial pin. This is if the sensor is at the bottom end of its target and at the top end of the actuator.

To receive the values from the sensor, we have to use Pin 5-Serial Output: By default, the serial output is RS232 format (0 to Vcc) with a 1-mm resolution. If TTL output is desired, solder the TTL jumper pads on the back side of the PCB as shown in the photo to the right. For volume orders, the TTL option is available as no-cost factory installed jumper. The output is an ASCII capital “R”, followed by four ASCII character digits representing the range in millimeters, followed by a carriage return (ASCII 13). The maximum distance reported is 5000. The serial output is the most accurate of the range outputs. Serial data sent is 9600 baud, with 8 data bits, no parity, and one stop bit, as described from the datasheet of the sonar sensor

The reason of selecting Arduino101 is it has on board Bluetooth low energy (BLE) and gyro sensor. This also calculate the 3 degree of angle with the help of on board accelerometer and gyro meter sensors. (Yet to explain) This value is precise up to nearest ½ degree of angle. It also get 6 distance values from Atmega 2560boards via software serial. After getting all the required data it arranges in a format starting with H, six length values separated by comma, three floating point radians separating by commas and ending with \n trailer. afterword its transmits this format to the monitor via serially

Multiple HRLV MaxSonar EZ sensors can be used simultaneously in the same environment generally with little to no interference

(cross-talk). Even so, some cross-talk may still occur for users wishing to use a large number of sensors in the same environment. This interference is rare and can be up to +/-1 cm of the target’s distance. Because of this, sensor

to sensor interference must be accounted for. To avoid interference between sensors, chaining can be used to prevent cross-talk between sensors. This will be necessary when using 3+ sensors depending on mounting and environment. The chaining works in a way where it enables the first sensor and reads the data while disabling the rest of the sensors. Once it finishes reading the data the first sensor then gets disabled and then the second sensor is enabled and reads the data while the other sensors are disabled and the chain goes on like this until all the sensors read the required data.

The recommended chaining method is AN Output Commanded Loop. The first sensor will range, then trigger the next sensor to range and so on for all the sensors in the array. Once the last sensor has ranged, the array stops until the first

sensor is triggered to range again.

**Arduino Uno**

* Receives Six sensor values
* Get the angular measurements
* Arranging the data in seeding format
* Sending data To monitor

Monitor

**Arduino Mega**

* Receives the six sensor values
* Sending sensor values to R101