Student Name: Michael Rowland
Robot Name: IGOR
TA: Scott Jantz
Instructor: A. A. Arroyo

University of Florida
Department of electrical and Computer Engineering
EEL 5666
Intelligent Machines Design Laboratory

FINAL REPORT

Table of Contents

Abstract

Igor is a robot which was designed to enter the IMDL competition. The competition requires that a robot enter the playing field and navigate it looking for objects to pick up. These objects will be placed in three of the four corners of the playing field and one object will be placed at random. The objects are three bags of rice, three boxes of tea, and three cans of tomato paste. Once the robot has secured an object it must recognize the object and place it in its appropriate bin. Igor was designed to accomplish this task with relative speed and reliability. It has a triangular scoop which will fit snugly into corners and it breaks in the center with a servo performing the lifting action. It navigates and detects the bins by use of infrared. It detects and distinguishes objects on its scoop by using Sharp PSD sensors.

Executive Summary

The IMDL competition provided a scenario in which Igor had to complete a basic task which we humans take for granted. To locate, recognize and pick up objects and to then place these objects in their appropriate bin. To achieve this goal Igor must overcome the following list of tasks. It must:

- Avoid walls and locate objects
- Pick up the objects
- Identify the objects
- Recognize that it has procured the object
- Locate the appropriate bin
- Drop off the object

The problem as a whole required that Igor be somewhat fast since it is a competition. It also had to be sturdy and balanced enough to be able to pick up the objects. Since the arena is relatively small it would need to be able to change direction quickly to avoid walls, and its lifting mechanism needed to be practical and simple since this was the most critical part of the robot and the one which was consistently tempting to over-engineer.

I decided to make it four wheel drive for speed and control and I also felt that all its navigation could be accomplished by using IR. I came upon the idea of having Igor hinge in the middle from my T.A. Scott Jantz, and decided that it would be both practical and not too difficult to use a servo as this hinge to raise and lower the scoop on the front of the robot. I got the idea of the PSD sensors while in a bathroom in Atlanta Airport.

These are the same sensors that turn on and off the water automatically by detecting ones hands. I decided to get these for my scoop so that Igor could detect objects that were on it.

Introduction

Igor is a robot which was designed to enter the IMDL competition. The competition requires that a robot enter the playing field and navigate it looking for objects to pick up. It must secure these object and place them in their appropriate bin. This report outlines the nature of this competition and the process by which Igor was created. The executive summary explains how I arrived at my final design and the main body of this report outlines how this design was carried out and also the specifics of the design. In closing this report examines the outcome of the competition and provides the necessary documentation used in creating Igor.

MAIN BODY

Integrated System

Since Igor was being specifically designed for the purpose of entering the IMDL competition, its design was restricted by size, practicality, functionality, and the environment that it was going to be in. As stated it had to be quick at accomplishing its task, strong and stable as it would be lifting the objects, and its lifting mechanism needed to be practical and durable. I knew that the lifting mechanism would be the crucial link and that I could not afford to over engineer this.

I therefore decided to design Igor with four wheel drive which would enable it to turn and maneuver easily and also give it speed and stability. The platform would need to be wide for stability also. Since the lifting mechanism was so crucial I decided to try design it with as few moving parts as possible. The fewer operations that needed to be performed in the lifting phase would lead to a faster robot with less chance of failure. With this in mind I decided to have the robot capable of breaking in the center and pivoting about the axis provided by its wheels to pick up its object. This operation resembles the operation of a dolly. The front and back of the robot needed to be connected by a strong servo with enough torque capable of lifting a 1lb bag of rice about the front axle.

Igor's navigation is all performed by using infra-red and bump sensors. The Sharp GP2D02 sensors are used to identify and verify capture of the objects. It has a triangular shaped scoop made from light aluminum protruding from its front end that slides along the ground in front of Igor like a dustpan. The Sharp sensors are set to detect anything

that goes on the scoop. It is powered by a 12 volt rechargeable battery pack and is controlled by the Talrik MRC11 and MRSX01 electronic boards. Igor has four bi-directional motors with the right side controlled by the same port (motor 0), and the left side controlled by (motor 1). Its wheels are 4 inches in diameter for better leverage. It has a 110 ounce-inch servo attaching its front and back halves. This servo is a heavy duty metal-geared servo made by Hitec. Igor is programmed with IC to accomplish all of its tasks. A diagram of Igor's design is included below in **figure 1**.

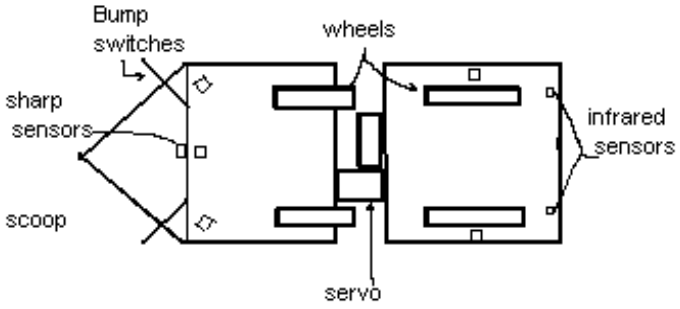

Figure1: Overhead view of Igor.

Igor's sequence of operation will be to:

(1) Initialize sensors

(2) Navigate playing field avoiding walls but searching for corners until its scoop is full

(3) When object is in scoop raise scoop slightly

(4) Navigate playing field again but now searching for bin

(5) When bin is located raise scoop to max position and then maneuver itself in front of the bin

(6) Reverse servo direction to drop off object

(7) Repeat process again

Mobile Platform

The platform design was centered around the pick-up servo, but also focussed on the problem of finding a corner and on maintaining balance when the object has been procured. It is constructed out of lightweight wood and has the electronic boards and battery pack mounted on the back above the rear axle to offset the weight of objects that are picked up. I was originally concerned about whether or not the platform would be able to turn effectively with the four wheels, but this proved to be no problem at all. The servo is mounted on the forward end, of the back half of the platform, with the servo horn toward the front and pointing to the right side of the robot. The servo horn is attached to a 1 inch block of oak which is protruding out of the back of the front half of the robot. I used oak because it is a hard wood and I could not afford to have the mount getting loose due to the consistent motion changes that the robot would be experiencing.

Actuation

Igor is driven by four servos which were hacked to act as motors. Its lifting mechanism is wholly comprised of the single servo in the robots center. The drive servos are the standard $7 servos sold in the lab which were hacked according to the procedure “Servo Hack : MECKATRONIX MS410 and MS455”, written by professor Keith L. Doty and

supplied on the IMDL web page. I had to build another motor driver circuit to handle the two extra motors as the MRSX01 board only has one motor driver chip and two motor ports. To build the motor driver circuit I just duplicated the circuit on the MRSX01 board from page 17 of the Talrik assembly manual. I pulled the motor inputs (motor 0 and motor 1) directly from pins 5 and 11 on the U5 (HC04) chip on the MRSX01 board. The lifting servo is powered directly from the servo power terminal on the MRSX01 board and it plugged directly into the servo1 port on the same board.

Sensors

The primary use of sensors on Igor are for navigation and for determining if something is on the scoop. The combination of IR emitters and Sharp IR detectors is used for navigation purposes. The emitters emit at 40kHz and their intensity can be controlled by varying the resistors in RP1 on the MRSX01 board. On Igor all these resistors are 470 ohm resistors. The detector used is the SharpGP1U58 digital IR sensor which is hacked to convert it to an analog detector. This simple hacking procedure can be found on page 32 of the Talrik Assembly Manual. I also collimated both the emitters and the detectors with black electrical tape to minimize interference. The emitters are located on Igor as shown in **figure1**. Igor also has two sensors on its scoop at 1 inch off the ground which radiate outward to detect the bins. With the angled front ones placed above two inches and the ones on the scoop below two inches Igor can distinguish between the bins.

The Sharp GP2D05 sensors, which are on the scoop to detect objects, are the same sensors which are used in restrooms to turn on and off the sinks. They are a transmitter and receiver combined together and they give a one bit output indicating if

something is in is range of detection. The detection range is easily adjusted by turning a small screw on the back of the sensor. The data sheets for this sensor have been included in Appendix A. The sensor is powered by a 5 volt DC supply and by a pulse waveform Vin. The waveform Vin is displayed in the data sheets and consists of a pulse which is on (at 3 volts) for 1ms and off for 69ms. In order to test and adjust the sensors I had to create this pulse in my software and load it onto my board. I then had to send it through a voltage divider circuit so that I could get the current within the required 160uA-270uA range. I viewed the sensors output with an oscilloscope and found that when it was detecting something that it went to zero and otherwise it was at 3 volts. When plugged into the analog ports on the MRSX01 board it will give a reading of 252 if it is seeing nothing and zero if it sees something. I used three sensors on my scoop to cover the whole scoop area of detection. I used a separate voltage divider for each sensor. This voltage divider is shown below. Out 1 is a pin on the IO header on the MRSX01.

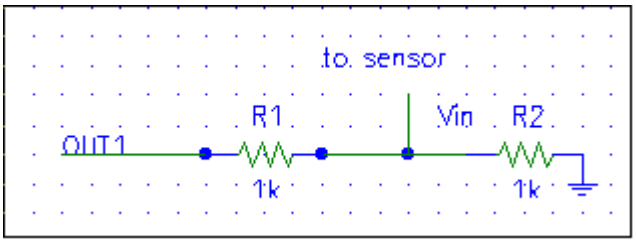

Figure 2:

Igor also makes use of two bump sensors which extend on two probes off the front of the robot at 45 degree angles which indicate if Igor is in a corner. Unfortunately I did not have time to install these before Igor's demo.

Behaviors

Igor's behavior characteristics consist of obstacle avoidance, corner finding, object pick-up, object recognition, bin finding, and object drop off. In Igor's software I made each behavior a process which was running at all times but was awaiting a particular global variable value to be effective. The sensor routine must be consistently radiating and evaluating the analog values to detect the objects when they go on the scoop. The corner finding process is operating all the time also until there is something in the scoop. Then the corner finding process stops being effective and instead triggers the bin finding process. The bin finding process then triggers the lift and dump routine. I found that my code required a lot of msleep commands because otherwise the processor was to fast for some of the sensors. Also I could not vary speed of the motors with IC because I had the servo powered at all times and IC tends cause problems when both are running.

CLOSING

Conclusion

The competition proved to be a very tough and challenging project. I unfortunately did not get all that I had hoped accomplished due to time constraints and mechanical failure. I had Igor doing trial runs on the day of demo but unfortunately the lifting servo burned out before I could demo it. I have decided that there is just a little too much torque and force exerted on the servo due to sudden turns and bumps and so I am replacing the burnt out servo with two such servos acting in unison. I also ran out of time before I could install the bump sensors which I feel would have cut down on some of the stress on the servo. I am very pleased with the scoop sensors since they worked very well and were extremely accurate. I would like very much to replace the servo with two servos and re demo next semester as I am confident that Igor will be very successful in its operation.

```
int m=0xf2;
```

```
int in_bin;
int pid, pid1, pid2, pid3, pid4;
int search;

 void forward()
  {
  m=(m|0xc0);
   poke(0xffb8,m);
  }

  void motor1bk()
  {
   m=(m&0x7f);
   poke(0xffb8,m);
  }

  void motor0bk()
  {
   m=(m&0xbf);
   poke(0xffb8,m);
  }

  void bothback()
  {
  m=(m&0x3f);
   poke(0xffb8,m);
  }

  void findcorner()
  {

  while(1)
  {
  if (in_bin==0)
  {

  if (analog(4)>100 && analog(6)>108)
      {
      forward();
        motor(0,50.0);
        motor(1,50.0);
        msleep(1000L);
        bothback();
        motor(1,100.0);
        motor(0,100.0);
        msleep(1000L);
        forward();
        motor(1,100.0);
        motor(0,100.0);
        msleep(500L);
        motor1bk();
        motor(1,100.0);
        msleep(1500L);
        forward();
        motor(1,100.0);
        motor(0,100.0);
```

```
    }
     else if (analog(4)>121 && analog(6)<123)
     {
        motor0bk();
         motor(0,100.0);
         }
     else if (analog(6)>124 && analog(4)<124)
      {
        motor1bk();
         motor(1,100.0);
         }
     else if (analog(5)>124 && analog(4)<109 && analog(6)<116)
      {
        motor1bk();
         motor(1,100.0);
         }

      else if (analog(2)>120 || analog(3)>120)
        forward();
      else  forward();
      motor(0,100.0);
      motor(1,100.0);

     }
}
}

    void sensor_1()
    {
    while(1)
    {
     m=(m|0x20);          /* set out1 1*/
     poke(0xffb8,m);
     msleep(01L);
     m=(m&0xdf);           /* set out 1 0*/
     poke(0xffb8,m);
     msleep(69L);
    }
    }



     void check_bin()
     {
     while(1)
     {
         in_bin=0;
         m=(m|0x03);                        /* selects irdt8*/
         poke(0xffb8,m);
         msleep(200L);
         if (analog(0)<=20)
             in_bin++;

         m=(m|0x24);                        /* selects itdt9*/
```

```
         m=(m&0xfc);
         poke(0xffb8,m);
         msleep(200L);
         if (analog(0)<=20)
             in_bin++;

         m=(m|0x21);
         poke(0xffb8,m);                          /*irdt10*/
         msleep(200L);
         if (analog(0)<=20)
             in_bin++;
         }
        }

         void bin_full()
         {
         while(1)
         {
         if (in_bin>0)
         {       kill_process(pid);
                 kill_process(pid1);
                 kill_process(pid3);
               /* msleep(2000L);*/
               forward();
               motor(0,100.0);
               motor(1,100.0);
               msleep(500L);
                bothback();
                motor(0,100.0);
                motor(1,100.0);
                msleep(2000L);
                servo_deg(118.0);
                search = 1;
                in_bin =0;


        }
      }
   }

    void find_bin()
    {
    m=(m&0xf0);
    m=(m|0xf2);
    poke(0xffb8,m);
    while(1)
    {
         if (search == 1)
            {
             msleep(200L);
             forward();
             if((analog(0)>123 && analog(4)<109) || (analog(7)>123 &&
analog(6)<109))
             {

               if(analog(0)>123)
               {
```

```
  servo_deg(90.0);
  motor1bk();
  motor(0,50.0);
  msleep(1000L);
  forward();
  motor(0,50.0);
  motor(1,50.0);
  msleep(1000L);
  servo_deg(110.0);
  servo_deg(90.0);
  bothback();
  motor(0,100.0);
  motor(1,100.0);
  msleep(2000L);
  servo_deg(123.0);
  pid=start_process(sensor_1());
  pid1=start_process( check_bin());
  pid3=start_process( findcorner());

  search=0;
    }
    else if(analog(7)>123)
    {
  servo_deg(90.0);
  motor1bk();
  motor(1,50.0);
  msleep(1000L);
  forward();
  motor(0,50.0);
  motor(1,50.0);
  msleep(1000L);
  servo_deg(110.0);
  servo_deg(90.0);
  bothback();
  motor(0,100.0);
  motor(1,100.0);
  msleep(2000L);
  servo_deg(123.0);
  pid=start_process(sensor_1());
  pid1=start_process( check_bin());
  pid3=start_process( findcorner());

  search=0;
    }
 }

else if (analog(4)>122 && analog(5)>125)
{
 motor0bk();
 motor(0,100.0);
 msleep(2000L);
 }
 else if (analog(6)>122 && analog(5)>125)
  {
 motor1bk();
 motor(1,100.0);
 msleep(2000L);
```

```
              }


              else if (analog(4)>122 )
              {
              motor0bk();
              motor(0,100.0);
              msleep(500L);
              }
             else if (analog(6)>122)
               {
             motor1bk();
              motor(1,100.0);
              msleep(500L);
              }
              else if (analog(5)>124)
              {
              motor1bk();
              motor(1,100.0);
              msleep(1500L);
              }

              else if (analog(2)>120 || analog(3)>120)
              forward();
              else forward();
              }
}
}

  void main()
  {
  search=0;
  servo_on();
  servo_deg(123.0);

    pid=start_process(sensor_1());
    pid1=start_process(check_bin());
    pid2=start_process(bin_full());
    pid3=start_process(findcorner());
    pid4=start_process(find_bin());
  }
```