Final Report
Grasshopper

Nicolai Hoffman

EEL 5666
Intelligent Machines Design Laboratory
Instructor: Keith L. Doty
12 December 1997

Table of Contents

ABSTRACT

My robot is a six-legged walker. Its purpose is to search for food, pick it up and then drop it off at home. The base of the robot is a RoboBug which was modified by adding a head. The head serves the purpose of picking up, holding, and dropping small metallic ball bearings which represent food. The head is a five bar mechanism with one degree of freedom. The head does all of these functions with just one motor turning in one direction. The robot uses IR emitters and detectors to find both the food and its home.

EXECUTIVE SUMMARY

The overall purpose of the class was to build an autonomous robot that preforms a practical task. It needed to have four different type of sensors and four behaviors. Using this criteria I decided to build an insect that could gather food.

Since I am a mechanical engineering student, I wanted to make something new in that area. At first I wanted a to make an effective six-legged walker but that had been done previously (RoboBug). I decided to add on to that design by adding a head. The head serves the purpose of picking up 'food', holding the 'food', and dropping the 'food' at its home. The 'food' is represented by metallic balls. Kinematically the head is a five bar mechanism with one degree of freedom.

The food is placed around a 32KHz IR emitter (beacon). The head has three 32KHz sensors, the left ear, right ear, and the nose. The left and right ear home in on the beacon until the nose can see it. When the nose can see the beacon the head is right above the food. The bug drops to the ground and activate the jaw. The jaw lowers to the ground and picks up the food with the aid of magnets. The jaw closes and if it has the food will close a connection stopping the jaw and then the bug will begin to look for home.

The home is a 40KHz beacon. The left and right eyes home in on this beacon just like the ears homed in on the 32KHz beacon. Once the sensors see that they are close enough to home, the jaw rotates just a little dropping the ball. The cycle then starts again by the bug looking for the food beacon.

INTRODUCTION

For my robot I decided to build an insect that could gather food. Since I am a mechanical engineering student, I wanted to make something new in that area. At first I wanted a to make an effective six-legged walker but that had been done previously. David Novick designed the RoboBug platform and Jennifer Laine had developed a good walking program for the Robobug. Instead of just repeating work that had already been done, I decided to add on to that design by adding a head.

The head serves the purpose of picking up 'food', holding the 'food', and dropping the 'food' at its home. The 'food' is represented by metallic balls. Kinematically the head is a five bar mechanism with one degree of freedom. The robot uses IR emitters and detectors to find both the food and its home.

FUNCTION

The main function of my robot is to be a gatherer. It was intended that Kevin Walchko's bug will search for the metal 'food' and mark it with a beacon (leave a trail to the food like ants do). However since Kevin dropped the class the food is simply be placed around a beacon. My robot will search for the beacon using IR sensors. The left and right ear home in on the beacon until the nose sees it. The bug then drops to the ground and tries to pick up the 'food' around the beacon until it gets something in its mouth (which closes a switch). It will then return to its home, using IR sensors in the left and right eye and another beacon, and drop off the 'food'.

The head has the function of finding and picking up a metal ball. It cannot get in the way of the walking of the bug. It must be simple to control and be flexible to a variety of conditions. For example it must be able to pick up the ball without knowing exactly where it is. As long as it knows that it is close (within a square inch or two) it should be able to pick it up. Also it should not need to know exactly how high the head is off of the ground. If the legs are in the up position the head should be able to pick up the bearing. The head must know if it did indeed pick up the ball and then hold it. Finally the head must be able to drop the ball.

HEAD DESIGN

The head is shown in Figure 1.

1 - body of the insect (RoboBug)
2 - head
3 - motor
4 - arm connected to the motor
5 - lower jaw that will pick up the ball
6 - magnet that will hold the ball
7 - bar connecting the head to the jaw
8 - ‘nose’ (32KHz IR sensor)
9 - ‘ears’ (32KHz IR sensor)
10 - ‘eyes’ (40KHz IR sensor)
11 - rubber bumper
12 - bar connected rigidly to the body and connected with a revolute joint to the head
13 - damper (not necessary but will reduce head vibration while walking)
14 - tooth (registers if food has been picked up and also wedges to ball out when dropping the food)
15 -metal ball bearing

Figure 1 is the head in relation to the bug's body with the legs in the up position (Bug as close to the ground as possible).

The bug uses his ears to find the beacon by the food. When the nose sees the beacon right below it, the bug stops and starts the jaw motor. The jaw hits the ground and as the jaw continues to open, the head pivots on the body so the jaw can continue to stay on the ground as the jaw moves forward. The jaw continues to move forward. When the magnet gets close enough to the ball bearing, the ball bearing is pulled into the mouth. The magnet holds the ball as the jaws pick it up. When the ball makes contact with the top tooth, an electrical circuit is completed. The bug knows that it has food and stops the motor. The bug walks back home by looking for the home beacon. When the bug is close enough to the home beacon, the jaw motor is turned back on. The top tooth wedges the ball from the magnet and the pushes the ball off of the lower jaw.

The graph on Figure 1 shows the x and y position of the tip of the jaw with respect to time. The graph shows that the jaw starts by coming almost straight down (y decreases while x remains almost the same) until it hits the ground. The tip is on the ground when the y graph is at its lowest position. The tip stays on the ground for a long time while the jaw moves forward (x increases while y stays the same). The tip raises off of the ground and goes back to the tooth. At the tooth the jaw goes in the negative x direction to push off the ball when needed.

The head was designed using the software Working Model 2.0. It was adjusted in working model until the head performed as I wanted. Using the dimensions found in working model I drew the head in AutoCAD 13. The head was cut out of wood and assembled. The actual head works very much like Working Model predicted.

SENSORS

The bug uses two boards. The little board controls the walking of the robot (Jennifer's Program). The larger board (MRC11) is responsible for reading the sensors, making a decision based on those readings and then telling the little board how to walk (i.e. go left, slower, stop...) There are six sensors that the MRC11 have to read (left ear, right ear, nose, left eye, right eye and tooth).

The food is placed around a 32KHz IR emitter (beacon). The head has three 32KHz sensors, the left ear, right ear, and the nose. The left and right ear home in on the beacon until the nose can see it. When the nose can see the beacon the head is right above the food. The bug drops to the ground and activate the jaw. The jaw lowers to the ground and picks up the food with the aid of magnets. The jaw closes and if it has the food will close a connection stopping the jaw. The bug will then begin to look for home.

The home is a 40KHz beacon. The left and right eyes home in on this beacon just like the ears homed in on the 32KHz beacon. Once the sensors see that they are close enough to home, the jaw rotates just a little dropping the ball. The cycle then starts again by the bug looking for the food beacon.

PROBLEMS

No project like this is without many problems. Here are a couple of the ones I encountered.
The first problem I encountered was that adding a large head to the RoboBug put the bug off balance. This was easy to fix by moving the batteries all to the back in the form of a tail.

Another problem was that my nose could see the beacon way before the head was above it. After trying various covers over the sensor I found that just a small black tube worked best to ensure that the beacon could not be read unless it was right under the nose.

At first my RoboBug had obstacle avoidance. However, when I started using a 40 KHz beacon I could not get the obstacle avoidance to work effectively. This could be done in software by turning off the emitters on the bug, take a reading, turn on the bugs emitters and using the difference for obstacle avoidance. However I did not have enough time to get this working.

Finally I had a problem with the 40 KHz sensors reading the 32 KHz beacon and vice-versa. Although it read the wrong beacon at a much lower level, if it was right on top of one beacon and looking for the other beacon the bug would get confused. I do not have a solution to this problem yet.

CONCLUSION

The bug works effectively. It can find beacons and pick up ball bearings. I want to thank David Novick for the design of RoboBug and his help, Melissa Jones for all of her help in the ICC11 programming and getting the bug running (I could not have done it without her programming help), and Jennifer Laine for all her help building the bug and making a walking program that I could use.

I still need to add an effective obstacle avoidance and fix the problem with the beacons, but I am very satisfied with what I accomplished this semester. The only thing that I would change is to work harder during the middle of the semester so the end would not have been so time consuming. It would have been nice to have a class period or two for the non electrical/computer engineers so that I could have learned some basics. For example I spent many hours trying to get my two boards to communicate. I did not know that when connecting the two board through the SCI ports, the transmit and receive lines on of the connectors had to be switched. This was probably obvious to most, but I had no idea to do this.
Overall this has been the most interesting and my favorite class of college. I am glad that I took it and hope more mechanical engineering student have the same opportunity.

APPENDIX

ICC11 Program Code

```
#include <mil.h>
#include <hc11.h>
#include <serial.h>
#include <analog.h>

#define TOC2_ISR jaw_hand
#pragma interrupt_handler jaw_hand

void sensor_module();
void read_sensors();
void init_jaw();
void find_food();
void pick_up();
void find_home();

int nose_sensor, turning_on, left_eye, right_eye;
int tooth_sensor;
int left_ear, right_ear, tooth, nose;
int bored, food_det, go_home;
char last_char, las_char;

void main()
{
   init_jaw();
   init_analog();
   init_serial();
   last_char = 'Z';
   las_char = 'Q';
   nose = 0;
   tooth =0;
   food_det = 0;
   go_home = 0;
   bored = 0;
   while(1)
   {
      read_sensors();
      sensor_module();
      if(nose)
         pick_up();
      else if(go_home && tooth)
         find_home();
      else if(!go_home && !tooth)
         find_food();
      ++bored;
```

```
    }
}

void init_jaw()
{
    INTR_OFF();
    TOC2 = 0000;
    TCTL1 = 0xC0;
    OC1M = 0x00;
    TMSK1 = 0x40;
    turning_on = 1;
}

void read_sensors()
{
    left_eye = analog(6);
    right_eye = analog(7);
    left_ear = analog(2);
    right_ear = analog(0);
    nose_sensor = analog(1);
    tooth_sensor = analog(3);

}

void sensor_module()
{
    if(tooth_sensor < 120)
    {
        tooth = 1;
        go_home = 1;
        INTR_OFF();
    }
    else
        tooth = 0;


    if((nose_sensor > 102) && (tooth == 0))
    {
         nose = 1;
         TCTL1 = 0xC0;
         INTR_ON();
    }
    else if(nose_sensor <= 102)
         nose = 0;

}

void jaw_hand()
{
```

```
     if(turning_on)
     {
       TOC2 += 2500;
       TCTL1 = 0x80;
       turning_on = 0;
     }
     else
     {
       TOC2 += 37500;
       TCTL1 = 0xC0;
       turning_on = 1;
     }

     CLEAR_FLAG(TFLG1, 0x40);

}

void find_food()
{
   if ((left_ear > 95) && (right_ear > 95))
   {
      if(last_char != 'F')
            {
            write("F");
            last_char = 'F';
            }
   }
   else if((right_ear > 100) && (left_ear < 100))
   {
      if(last_char != 'R')
            {
            write("R");
            last_char = 'R';
            }
   }
   else if((left_ear > 100) && (right_ear < 100))
   {
      if(last_char != 'L')
            {
            write("L");
            last_char = 'L';
            }
   }
   else
   {
       if(bored > 15000)
       {
         if(last_char != 'F')
```

```
            {
            write("F");
            last_char = 'F';
            }
         if(bored > 40000)
           bored = 0;
        }
        else
          if(last_char != 'R')
            {
            write("R");
            last_char = 'R';
            }
   }
}

void pick_up()
{
   write("S");
   while(tooth == 0)
   {
      read_sensors();
      sensor_module();
   }


}
void find_home()
{
   int i;
      if(las_char != 'P')
            {
            write("S");
            las_char = 'P';
            last_char = 'Z';
            }
   if((right_eye > 104) && (left_eye > 104))
   {
       TCTL1 = 0xC0;
       INTR_ON();
       for(i=0; i<15000; ++i);
       INTR_OFF();
       nose = 0;
       go_home = 0;
       tooth = 0;
       las_char = 'Q';\
   }
   else if((right_eye > 90) && (left_eye > 90))
   {
```

```
    if(last_char != 'F')
        {
        write("F");
        last_char = 'F';
        }
  }
  else if((left_eye > 100) && (right_eye < 100))
  {
    if(last_char != 'L')
        {
        write("L");
        last_char = 'L';
        }
  }
  else if((right_eye > 100) && (left_eye < 100))
  {
    if(last_char != 'R')
        {
        write("R");
        last_char = 'R';
        }
  }
  else
  {
    if(last_char != 'R')
        {
        write("R");
        last_char = 'R';
        }
  }

}
```