Date: December 5, 1995
Student: Michel H. Boyer
Instructor: Keith L. Doty

University Of Florida
Department Of Electrical Engineering

EEL 5934
Intelligent Machines Design Laboratory

Final Report

Table of Contents

Abstract

One of the new faces in the Machine Intelligence Lab this Fall semester is “Mazo” Mazo is a robot designed to apprehend any prey that moves or makes noise. With a light body and a ergonomic layout of its components (sensors, processor, battery...), Mazo is able to respond quicker and better to the detection of a prey. A part from being a natural killer, “Mazo” has some other useful behaviors, like object avoidance, edge of the world detection that make it an exceptional robot.

Executive Summary

Mazo is made out of a perforated doubled up aluminum body. This body is light in weight to carry the expanded Motorola S68HC11 and the necessary components to conduct the behaviors. Two Futoba servo motors are mounted under the round platform on an axis at each corner. There are four sharp infrared detectors under the front end, they are use together with the four emitters on top of the platform to avoid any objects. There is another emitter under, at the front end, which is used by the middle sharp sensors to detect if there is an upcoming edge, so that it does not damage its beautiful frame. This latter emitter also helps Mazo to avoid getting stuck on black walls, which is not seen by the sharp sensors. Another sharp sensor is hidden between the caster and the platform. This sensor is used to find other robots sneaking behind Mazo.

To reflect as much the characteristics of a predator, Mazo has a head with eyes and ears. The small wood platform serves to turn (0-180 degrees) a ultrasonic motion detector and two sound detectors[1]. The ultrasonic movement detector uses a 40 kHz signal to notice if something is moving . The sound detection uses itself two mic, one for each side, to detect where the sound is coming from. From these "head" sensors, Mazo can determine where the prey lurks and it will go towards the target.

All the decisions and priorities are executed by the "brain", the S68HC11 microprocessor.

[1] The sound detection was added as a senior design. For more information, refer to EEL 4914 Michel H. Boyer's senior design: Sound Detection.

Introduction

The purpose of this project is to have a predator catch a prey. This is a type of project that many people address as “cat and mouse” chase. But for this to happen, the predator must first find the prey. Since nobody in the Machine intelligence lab will sacrifice their robot for a “capture scene”, Mazo is designed to locate and follow a prey.

A predator uses his speed, agility, experience, smell, vision, touch, hearing to achieve his goal. Mazo will try to emit some of theses simple behaviors as best as a intelligent machine and its designer can.

Therefore Mazo’s objectives are:

1) To have fun moving around if there aren’t any preys lurking around.
2) To find a prey using motion detection and sound detection.
3) To follow the prey until it has considered it a capture.

Robot Platform

The new platform is made out of a sheet of perforated aluminum double up. It is stronger, cleaner, and the perforations help make switching components places very easy. The switches fit right in the holes and the robot seems more stable. And for a predator Mazo looks really like a distinguish hunter.

The platform has a circular shape with a 12in diameter. There are rectangular cuts at each corner, so as to fit the 5in gripping wheels. There is a 11/4 in caster attached by means of screws at one of the other corners. The microprocessor is located on top facing forward. This a very easy platform to work with.

Servo motors assembly

There are two Futoba (FP-S148) servo motors used to move the robot. The motors were modified to rotate 360 degrees and to be used with the another motor driver. This easily done by taking out completely the circuit that the servos come with and directly soldering the power and ground to the back of the actual motor. The stopper, located on one of the gears is also cut off. the motors horns are glued to the back of the wheels and the motors are attached by tightening the screws between the horn and the motor head gear. It is attached to the platform with the help of L-brackets.

Added to the robot is another servo motor on top of the platform to help turn a small platform, made out of plywood, which holds the motion sensor and sound detector. No modifications were made to the motor, so it only has a 0° - 180 ° rotation.

Infrared, Ultrasonic and sound sensors

Mazo uses Sharp IR sensors to avoid objects. These sensors where modified to be used with the 68HC11's A/D converter. The analog output, of the sharp sensor, comes from a wire soldered at the end of a capacitor. Mazo has 5 of those sensors.

The ultrasonic sensor uses sound to detect motion. This device sends out a sound at 40kHz a couple of times and if the signal strength changes to a certain level determined by the value given from the board's potentiometer (called 'sensitivity' control), it detects movement. This sensor, originally designed for house alarms, was too sensitive, so some modifications (mostly to the resistors) was made to cut down from its 20 feet detection to now a 4 - 5 feet. This is to assure that Mazo will detect movement from a prey that is close by, decreasing the possibility of loosing the subject.

The circuit and characteristics of this sensor can be found in Fig.1 and Table.1.

The sound detection will use the circuit of Fig.2. This sensor is made out of two parts, the left and the right side. Both will use FET Mic to detect emitting sounds. The processor will compare the intensity from the analog inputs and will direct the robot towards the area where the sound intensity is higher. One thing to keep in mind though, the circuit, from the characteristics point, are not the exactly the same (as shown in Table.2), so programming will have to take it in consideration and make the necessary adjustment.

Both the ultrasonic motion detector and the sound detector of Mazo are use to direct him towards the prey by using a movable platform. This makes it possible for Mazo to detect motion or sound by stopping (therefore not making a noise) and turning its “head” to sense its surrounding.

Experimental Layout and Results

(Refer to fig.1 and table.1)

When the ultrasonic motion detector sends out a signal at 40 kHz, the signal is received through the receiver and is amplified by two amplifiers. the first one is set to let the 40 kHz signal through. Between the two is a negative peak detector (diode and resistor of 1MΩ) which follows the envelope of the 40 kHz signal. If there is no movement, the envelope is just a straight line. the second amp time constant is much slower than the first so that it will follow this envelope. This is fed through a circuit (the circuit detects potentials which move both below and above a given range) to detect positive and negative pulses. If there is no movement the potential is half coming from the second amplifier and neither diodes will conduct.

Behaviors

The behaviors that were realized using these sensors were:

1) Object avoidance

Four detector and emitters operate to tell the processor of any incoming objects.

2) Edge of the world detection

A single emitter pointing towards the ground and the use of the two middle detectors.

3) Black wall escape

Using the same previous emitter towards the ground, with a little angle towards the front. The robot thinks that when it hits a black wall it has reached the end of the world and should turn away.

4) Motion detection

Mazo can detect any movement and go towards it.

5) Sound detection

If there isn't any movement, but there is sound. Mazo will move towards the sound.

6) Robot detector

Using a single sharp sensor on the rear, Mazo will let other robots approach, and when at a certain distance, Mazo will turn around and face the other robot.

Conclusion

Mazo is complete. All the hardware works, and there is no sign of either noise or other problems. The algorithm demonstrates the behaviors of the robot as best as it can. The algorithm, though, is not completely perfected. The robot is a little bit slow when using the behaviors of the head. This is due to the lack of "trial and error" at the programming stage. But this issue could be taken cared of with an extra week or two of work. But the robot runs very well and is pretty affective in its search for a prey. The only thing that was discouraging for Mazo is that it could not actually demonstrate, on other robots, its natural ability to be a fierce killer.

Potentiometer	Resistance	Distance	Output from A/D
Minimum setting	2 MΩ	5.7 feet	134
Maximum setting	3 MΩ	4.2 feet	133

Table.1 : Results measured from motion detector.

Analog Value when →	**Stable**	**Maximum**	**Minimum**
Right Side	156-159	237*	232*
Left Side	166-170	40	35
Distance	-	6 feet**	-

Table.2 : Characteristics of sound sensor.

* Value observed. May differ since is very fast.

** Depends on the intensity of the sound. Measurement made for normal talk sounds.

```
/*  Functions that move the robot in the wanted direction */



void go_right() {               /* if object is on left side of robot*/
   motor(left_motor,norm_speed);
   motor(right_motor,lit_speed*-1);
   wait(75);
}

void go_left()  {               /* if object is on right side of object*/
   motor(right_motor,norm_speed);
   motor(left_motor,lit_speed*-1);
   wait(75);
}

void go_forward() {             /* always goes straight ahead        */
   motor(right_motor,right_speed);
   motor(left_motor,left_speed);
}


void go_back()  {               /* If robot gets into cornor          */
   motor(right_motor,half_speed*-1);
   motor(left_motor,half_speed);
   wait(400);
            }

void go_full_back()  {              /* If robot gets into cornor          */
   motor(right_motor,back_speed*-1);
   motor(left_motor,back_speed*-1);
   wait(800);
            }




/* The following function always checks the processor time */


void  read_time()
{
while(1)  {

sys_time = seconds();

        }
}



/* The next two functions keep count of the eyes and ears */
```

```
void  read_sonic()   {
int i=0;
motion=analog(7);
   while (1)  {
   if( (i>60) || (motion != m_sensor) )
         break;
       else {
             motion = analog(7);
          wait(3);
             i= i+1;

                  }
   }
  }



void listen() {
int j=0;
left_ear = analog(5);
right_ear = analog(6);

while(1)  {
   if( (j>450) || ( (left_ear > l_noise && right_ear > rmin_noise) || (right_ear > r_noise && left_ear >
lmin_noise)))
   break;
   else{
       left_ear = analog(5);
       right_ear = analog(6);
       j = j +1;
     }

       }
}


/* This function reads the sensors */

void read_sensors()  {           /* read in sensors                        */
   while(1){
     poke(0x7000,0b00001000);
     wait(3);
     left=analog(3);
     poke(0x7000,0b00000001);
     wait(3);
     right=analog(0);
     poke(0x7000,0b00000010);
     wait(3);
     middle_r=analog(2);
     poke(0x7000,0b00000100);
     middle_l=analog(1);
     behind = analog(4);
     poke(0x7000,0b10000000);
     wait(3);
```

```
        und_l = analog(1);
        und_r = analog(2);
        defer();
        }
}
```

/* The following function allocates times for moving around
and avoiding obstacles, then after a specified time, gives
control to the head. */

```
void avoid_obstacles()  {          /* read in sensors and compare with        */
                                   /* threshold  level.                       */
   while(1){


    if (sys_time < times)  {

if (middle_r >= m_threshold  || middle_l >= m_threshold)
      {

           go_back();
       }


if (behind >= back_threshold)
        {

        go_back();
         wait(500);
         }

if(und_l < under_threshold ||  und_r < under_threshold)
{
ao();
wait(300);
go_full_back();

go_back();
wait(200);
}


if( right > threshold && left > threshold  )
{
temp = temp + 1;
  if(temp > 75)
  {
  ao();
  wait(300);
  go_full_back();
```

```
  temp = 0;
    }
}


    if (left > threshold && right <= threshold)
          go_right();                                  /* jump to right  */

  else  if (right > threshold && left <= threshold)
          go_left();                                   /* jump to left   */

     else go_forward();


    }

    else {

       ao();
       check=0;

       servo_on();
       servo_deg(90.0);
       wait(50);
       servo_off();

       place = 1;
       wait(3500);
       read_sonic();
    if (motion != m_sensor && (place == 1))
    {

       place =1;
       }
           else {
           listen();
           sound_detect();
       if( check == 0) {
            servo_on();
            servo_deg(180.0);
            wait(350);
            servo_off();
            place = 2;
            wait(3750);
            }
   }

   if(place == 2 && check == 0)
     read_sonic();


   if(motion != m_sensor && (place == 2) && (check == 0))
            {
```

```
            go_left();
            wait(750);}



           else {
           if(place != 1 && (check == 0)) {
               listen();
               sound_detect();
               if(check == 0) {
               servo_on();
               servo_deg(0.0);
               wait(400);
               servo_off();
               place = 3;
               wait(4000); }
               }
}
      if(place == 3 && check == 0)
         read_sonic();

         if(motion != m_sensor && (place == 3) && (check == 0))
                     {

                      go_right();
                      wait(650);}

           else  {
           if(place == 3 && (check == 0))
             listen();
             sound_detect();
                      }
           if(place != 1) {
             servo_on();
             servo_deg(90.0);
             wait(350);
             servo_off();
             }
         place =1;
         servo_off();
         times = sys_time + sonic_delay;
                   }
  defer();
   }
     }



/* the following function control the decision  when a sound is detected */


void sound_detect()  {
```

```
side = 0;

if(place == 1) {

 if(left_ear > l_noise && right_ear > r_noise)
 {
 l_sub = left_ear - l_noise;
 r_sub = right_ear - r_noise;
 if(l_sub > r_sub)
 side=1;
 if(r_sub > l_sub)
 side=2;
 }

 if(left_ear > l_noise && right_ear < r_noise && right_ear  > rmin_noise || (side == 1))
        {
        check = 1;
        go_left();
        wait(600);

        }
  else
   if(right_ear > r_noise && left_ear < l_noise && left_ear > lmin_noise || (side == 2))
           {

           check = 1;
           go_right();
           wait(600);

           }

     }

if (place == 2) {

 if(left_ear > l_noise && right_ear > r_noise)
 {
 l_sub = left_ear - l_noise;
 r_sub = right_ear - r_noise;
 if(l_sub > r_sub)
 side=1;
 if(r_sub > l_sub)
 side=2;
 }


 if(left_ear > l_noise && right_ear < r_noise && right_ear  > rmin_noise || (side == 1))
        {
        check = 1;
        go_back();
        wait(350);
        }
  else
   if(right_ear > r_noise && left_ear < l_noise && left_ear > lmin_noise || (side == 2))
```

```
       check = 1;
     }

if (place == 3)  {

 if(left_ear > l_noise && right_ear > r_noise)
 {
 l_sub = left_ear - l_noise;
 r_sub = right_ear - r_noise;
 if(l_sub > r_sub)
 side=1;
 if(r_sub > l_sub)
 side=2;
 }


   if(right_ear > r_noise && left_ear < l_noise && left_ear > lmin_noise || (side == 2))
           {
           check = 1;
           go_back();
           wait(350);
           }
    else
     if(left_ear > l_noise && right_ear < r_noise && right_ear  > rmin_noise || (side == 1))
     check = 1;
     }


        ao();

      }


void main()  {

 reset_system_time();
 start_process(read_time() );
 start_process(read_sensors() );
 start_process(avoid_obstacles() );






}

void wait(int milli_seconds)     {
  long timer_a;
  timer_a = mseconds() + (long) milli_seconds;
  while(timer_a > mseconds()) {
    defer();                    }}
```