University of Florida

Department of Electrical Engineering

EEL5666

Intelligent Machine Design Laboratory

Doc Bloc

Larry Brock

April 21, 1999

IMDL Spring 1999

Instructor: Dr. Arroyo

Table of Contents

Abstract

I am designing an autonomous robot that is going to move randomly, avoid obstacles, search for foam blocks of a specified size, and then use them to build a wall. Upon finding the block the robot will pick it up using pinchers. The robot will then follow a beacon to a specified position and build a two-story wall of the foam blocks.

Executive Summary

Doc Bloc is intended to be a wall building robot. The robot searches around until it finds a block. It then turns toward the object until the front IR sensor gets an acceptable reading. Then using the IR sensors on the ends of the pincher arms, the robot determines if the object it saw was a block and if so it then centers the block in the pinchers. The robot slowly moves forward until the block trips the IR break beam that crosses the pinchers. Finally, the robot will close the pinchers and grab the block and lift it up. At this point the robot looks for another block and repeats the process. Doc Bloc's platform is a slightly altered semicircle shape with the pincher system mounted on the front of the robot. There are two motors to control two wheels for the robot's motion. The pinchers use two servos and gears to open and close the pinchers and raise and lower them. There is a four IR system with two submini lever switches to find and grab blocks and make sure they are in the correct position. There are four other submini switches for obstacle avoidance and eight bump sensors also for obstacle avoidance. The robot is controlled by a 68HC11 board with an ME11 board and powered by eight AA batteries. The robot is programmed using IC.

Introduction

The goal of my robot is to move around randomly while avoiding obstacles and searching for foam blocks. The long term goal of the robot is for it to be able to construct a wall using the blocks. I wanted to do build a robot that had the ability to grab items. In looking at the previous couple of semesters robots I did not notice a robot that grabbed blocks and stacked them, or a robot that built a wall. This is one of the reasons I chose to build this type of robot. The toughest parts of building this robot proved to be centering the block in the pinchers, constructing the claw, and stacking the blocks. Although, I encountered these difficulties, I was able to get this robot to find the block, center it in the pinchers and pick it up and then find a new block. This paper will discuss the construction of the robot and also the successes and failures in the process of constructing the robot.

Integrated System

The robot is going to be powered by eight AA batteries. It will use two 43oz/in hacked servo motors to run the two tires. Two other 43oz/in servo motors will be used to control the pinchers located on the front center of the robot. One of the servos will be used to open and close the pinchers while the other will be used to raise and lower the pinchers. These two servos account for the two degrees of freedom of the claw.

The brains of the robot is going to be a 68HC11 along with an ME11 board. The ME11 board is used to expand the RAM memory of the 68HC11 to 32K. The board also provides a voltage regulator, some input and output addresses, and produces a desired frequency for the Sharp IR detectors.

I intend to use three IRs to locate the desired foam block. I will also use one IR located on the claw to determine when the block is in a position to be grabbed. I am also going to use about four IR sensors and eight bump sensors to handle obstacle avoidance. There will also be four submini lever switches to help with obstacle avoidance.

Mobile Platform

I am planning on using an altered semicircle platform for my robot. There will be three bump sensors on the front of the robot and five on the back to handle obstacle avoidance. The pinchers I will be using will be mounted on to the front of the platform. The pinchers are going to be able to rotate approximately 110° perpendicular to the platform enabling it to pick items up off the ground. The platform will be approximately 10 inches wide and about 12 inches long. The platform will be adjusted to account for the two tires and also for the pinchers. I will be using three inch wheels so the robot will sit slightly over an inch off the ground. These wheels are located approximately a half inch from either edge of the robot and about an inch from the rear of the robot. There is a castor wheel located on the front center of the robot and an inch from the front.

The only problems I encountered with the construction of the platform was with the construction of the pinchers. My first design of the pinchers were too short, they did not close flush on the block and they could not be lowered enough to grab the blocks. The third design of the pinchers corrected all of these problems. I designed the pinchers into three parts. The first part connected to the gear system. The second part is simply four circular pieces of wood. This lowers the third part of the pinchers to a level acceptable for grabbing the blocks. The last part of the pinchers is the section that grabs the blocks.

All three sections are connected using a bolt and nut. This allows me to adjust the angle of the last section of the pinchers to be flush with the blocks. Thus, After three different designs I had one that worked well.

Actuation

I used two 43oz/in servo to operate the two wheels of the robot. These servos are hacked into and thus operate as normal DC motors. These motors are going to be responsible for all movement of the robot. Some of the movements the robot will perform are: all forward, all reverse, left motors forward, right motors forward, and spin. This will account for the motion of the robot including turning the robot. Since I do not anticipate using the robot on any rough terrain, the 43oz/in servos give me plenty of power to operate the robot.

For the motion of the claw I am going to use two 43oz/in servos. One to control the grabbing of objects. This motor will open and close the claw to grab and release objects. I will be using gears so that I can control both arms with only one servo. The second servo will be used to rotate the claw perpendicular to the robot platform over 110°. The robot is only going to be lifting foam blocks so the 43oz/in servos are capable of handling this operation. They were tested using a block of wood cut out of the TTech machine and were successful in lifting this, so the foam blocks are no problem. Thus, by using the two motors for the claw, it will have two degrees of freedom.

Sensors

Obstacle Avoidance

One of the sensors that will be used for Doc Bloc's obstacle avoidance is the Sharp IR Sensor. There will be five IR sensors on the front face of the robot. These sensors will not only be used for obstacle avoidance but will also be used to detect the blocks. Then an IR sensor will be placed on either side of the robot about an inch in front of either wheel and a half inch from the edge of the robot. These two sensors will be at an angle of 30°, from straight ahead, away from the robot.

The IR detectors that are being used for obstacle avoidance have been hacked. The hack alters the detectors from being digital detectors to analog detectors with a range of approximately 89 to 130.

The IR emitters run off the 40kHz pins located on the ME11 board. Meanwhile, the IR detectors are connected to analog port E. The two IR sensors on either side of the robot are connected to pins one and two of the expanded PE7. The five IR sensors located on the front of the robot are connected to pins three through seven of the expanded PE7.

Bump Sensors will be used to enhance the Sharp IR Sensors in obstacle avoidance. The bump sensors will be located on the front and back of the robot. There will be a rigid bumper directly around the front bumpers and one around the back bumpers. There will be three sensors on the front wired together to detect contact on the front of the robot. This set of sensors will be connected to pin one of the expanded PE6. There will be five bump sensors wired together on the back of the robot. These sensors will be wired to pin two of the expanded PE6.

Block Detection

For block detection there will be five sensors mounted across the front of the robot. They will be evenly spaced apart from each other. The robot will travel randomly until three adjacent sensors or two on the either end get the appropriate reading from the analog IR detectors. With either scenario the robot will turn to line up with the object until the three middle sensors have an acceptable reading and the outside two have a much higher reading. At this point the robot will move directly toward the object until a beam across the pinchers of the robot is broken. At this point the robot will determine if the block is an acceptable position. As stated in a previous section these IR detectors are connected to pins three through seven of the expanded PE7.

Special Sensor System

The special sensor system is designed to determine if the block is in an acceptable position to be picked up by the pinchers. This system consists of one Sharp IR sensor and two submini lever switches. The IR emitter and detector are located across from each other on the arms of the pinchers. They are positioned at the bend in the arm (See Diagram 1). The IR detector is wired into pin three of the expanded PE6. When a block breaks this beam the robot stops its forward motion. At this point the robot will begin to close its pinchers on the block.

When the pinchers have moved to the closed position, the submini lever switches are then checked. These switches are located at the end of both arms of the pinchers (See Diagram 1). These switches are wired into pins four and five of expanded PE6. If either of the switches(active-low) or both of the switches are return a value of zero, the block is in an acceptable position to be picked up and stacked. If , however, neither switch has been tripped the block is not in an acceptable position (See Diagram 2). If it is the case where the block is not in an acceptable position, the robot will open the pinchers and back off the block. It will then reapproach the block from a different angle.

Block Stacking

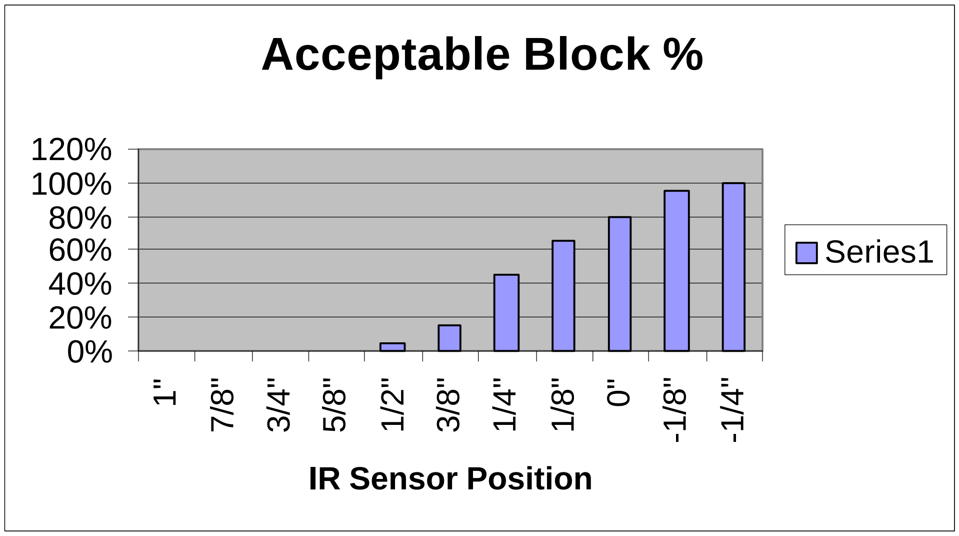
The sensors for this feature have not been determined yet. One possibility is have a spring bump sensor hanging beneath each arm of the pinchers. When the robot is carrying a block and locates another block, the robot will move toward the block until the spring bump sensor is tripped. At this point the robot would stop and drop the block. This is a tentative idea until I get to this point in the construction of the robot.

Experimentation

I performed a couple of experiments on my special sensor system. The position of the IR sensor on the arm had to be determined to give about 35% to 40% acceptable block positions. This was done by moving the IR emitters and detectors from 1/4” in front of the bend in the arms to 1” beyond the bend. The sensor was moved in 1/8” increments. The blocks were placed at different angles to the robot until an unacceptable case is found. At this point I can determine what percentage of block positions is acceptable based on the first unacceptable block position. Thus far in my experiments the best percentage of acceptable block positions is approximately 45%. This is higher than I would like, so I will continue to run experiments until I can find an acceptable result. Here are the results of this experiment:

Acceptable Block Table and Graph

1"	0%
7/8"	0%
3/4"	0%
5/8"	0%
1/2"	5%
3/8"	15%
1/4"	45%
1/8"	65%
0"	80%
-1/8"	95%
-1/4"	100%



Vendor information on the sensors is located in Appendix A. Circuit diagrams for the bump sensors and the submini lever switches to determine if the block is in a good position are located in Appendix B.

Behaviors

The original behaviors of my robot were going to be, grabbing items with the claw, lifting items with the claw, and building the wall. The first behavior of the robot will be to locate blocks. The robot will move around and upon detecting an object will turn to face the object. At this point the robot will lower the pinchers to center the block using the two IR sensors located on the end of the pinchers. If both sensors on the pinchers give high IR readings the robot will decide that the object is not a block and it will then turn around and go in another direction. Grabbing items is the next behavior that my robot will perform. It will grab the blocks once the IR break beam across the pinchers in tripped. Using a servo and some gears the robot will be able to open and close its pinchers to grab objects and release objects. To move the arms of the pinchers I am sending various PWM signals to the servo, by using the servo commands in IC, to get the desired motion of the pinchers. Once the robot has a block in its pinchers, it will look for another. Upon finding another block the robot will stop momentarily. At this point I wanted the robot to center the new block and stack the block it already has on top of it. However, I could not get this to work so instead the robot will turn drop its block and grab the new one.

Conclusion

In building this robot I discovered a lot. The first thing was the time involved in building and getting a robot to work. This is why I decided to focus on getting my robot to find and grab blocks and then to stack them. I had little trouble getting the robot to move around with the motions I wanted. The robot would go forward and back, turn left and right, and spin to whatever position I desired. Centering the block between the two arms of the pinchers proved to be a tough task. Initially I wanted to mount IR sensors on the robot platform to center the block. However, even using collimators the IR between the three centering IR sensors interfered with each other too much. For this reason I had to mount the sensors on the pinchers of the robot to center. This worked very well for centering the block. However, when I began working on the stacking behavior I found that I could no longer use this method to center because the pinchers were now closed and would give virtually the same reading all the time. This is why I was unable to stack the blocks. To be able to stack the blocks I am going to need to figure out a new design for the IR sensors and also use a Sharp IR distance detector. Thus, while having numerous problems with the construction of the robot, I did have several successes and was able to locate and grab blocks.

Appendix A

Vendor Information

• **Bump Switches**	Mekatronics (purchased from Scott Janz) $0.75 each
• **LED Emitter**	Mekatronics (purchased from Scott Janz) $0.75 each
• **LED Detector (GPIU58Y)**	Mekatronics (purchased from Scott Janz) $3.00 each
• **Submini Lever Switches**	Radio Shack (352)375-2426 **Part # 275-016A** $1.89 each
• **1 ¼" Gears(2)**	The Hobby Shop (352)3771-1128 $3.99 each
• **Servo Motors(2)**	Mekatronics (purchased from Scott Janz) $13.00 each
• **Motors(2)**	Mekatronics (purchased from Scott Janz) $13.00 each
• **ME11 Expansion Board**	Mekatronics (purchased from Scott Janz) $20.00 each
• **68HC11 EVBU Board**	University Technology Hub (352)392-0194 $68.00

Appendix C

```
/****************************************************/
/*** Larry Brock ***/
/*** Doc Bloc    ***/
/*** 04.15.99    ***/
/****************************************************/

int blocInFront;
int blocCentered;
int numStacs;
int havBloc;
int blocStacabl;
int frontBS=6;
int rearBS=5;
int rightBS=4;
int leftBS=3;
int irbb=0;
int pinchBS=2;
int IR0;
int IR1;
int IR2;
int IR3;
int IR5;
int IR6;
int IR7;

/***********************end of global variables*************************/
/*20*/

void stay(){
  motor(0,0.0);
  motor(1,0.0);
}

void goStraight(){
  motor(0,100.0);
  motor(1,100.0);
}

void goBack(){
  int i=0;
  motor(0,-100.0);
  motor(1,-100.0);
 while(i<2000){i=i+1;}
```

```
   stay();
}

void moveRight(){
   int i=0;
   int j=0;
   motor(0,100.0);
   motor(1,-100.0);
 while(i<1000){i=i+1;}
   stay();
 while(j<2000){j=j+1;}
}

void moveLeft(){
   int i=0;
   int j=0;
   motor(1,100.0);
   motor(0,-100.0);
 while(i<1000){i=i+1;}
   stay();
 while(j<2000){j=j+1;}
}

void turnRight(){
   int i=0;
   motor(0,100.0);
   motor(1,-100.0);
 while(i<1750){i=i+1;}
   stay();
}

void turnLeft(){
   int i=0;
   motor(0,-100.0);
   motor(1,100.0);
 while(i<1750){i=i+1;}
   stay();
}

void Left(){
   motor(0,-100.0);
   motor(1,100.0);
}

void Right(){
   motor(1,-100.0);
   motor(0,100.0);
}

void goLeft(){
   int i=0;
   motor(0,0.0);
   motor(1,100.0);
 while(i<3000){i=i+1;}
   goStraight();
}

void goRight(){
   int i=0;
   motor(1,0.0);
   motor(0,100.0);
 while(i<3000){i=i+1;}
   goStraight();
}

void spin(){
   int i=0;
   motor(0,-100.0);
   motor(1,100.0);
 while(i<8000){i=i+1;}
   stay();
}
/*100*/

void turnaround(){
   int i=0;
   motor(0,-100.0);
   motor(1,100.0);
 while(i<5000){i=i+1;}
   stay();
}

void goBac(){
   motor(0,-100.0);
   motor(1,-100.0);
}

/*****************end of motion routines***************/
/*116*/

void bow(){
   int i=0;
   int j=0;
   int k=0;
   int m=0;
   int n=0;
   int p=0;
   int q=0;
   int r=0;
   int s=0;
   int t=0;
   int u=0;
   int v=0;
   int w=0;
   int x=0;
   int a=0;
   int b=0;
   int c=0;
   servo_deg1(150.0);
   goStraight();
   while(i<2500){i=i+1;}
   stay();
   while(j<1500){j=j+1;}
   Right();
   while(k<2500){k=k+1;}
   servo_deg1(50.0);
   while(m<1500){m=m+1;}
   servo_deg1(150.0);
   while(u<1500){u=u+1;}

   Right();
   while(n<2500){n=n+1;}
   servo_deg1(50.0);
   while(p<1500){p=p+1;}
   servo_deg1(150.0);
   while(a<1500){a=a+1;}

   Right();
   while(q<2500){q=q+1;}
   servo_deg1(50.0);
   while(r<1500){r=r+1;}
   servo_deg1(150.0);
   while(b<1500){b=b+1;}

   Right();
   while(v<2500){v=v+1;}
   servo_deg1(50.0);
   while(w<1500){w=w+1;}
```

```
   stay();
   servo_deg1(150.0);
   while(x<1500){x=x+1;}
   numStacs=5;
}

void grabBloc(){
   int i=0;
   int j=0;
   int k=0;
   numStacs=numStacs+1;
   servo_deg2(170.0);
 while(i<1000){i=i+1;}
     havBloc=1;
     blocCentered=0;
     blocInFront=0;
     servo_deg1(150.0);
     while(k<2000){k=k+1;}
     stay();
}

void centerBloc2(){
   int i=0;
   int j=0;
   int k=0;
   int m=0;
   int n=0;
   Right();
   while(i<2000){i=i+1;}
   stay();
   servo_deg2(170.0);
   servo_deg1(70.0);
   while(j<2000){j=j+1;}
   servo_deg2(150.0);
   while(k<2000){k=k+1;}
   servo_deg2(170.0);
   servo_deg1(150.0);
   while(n<2000){n=n+1;}
   Left();
   while(m<2000){m=m+1;}
   stay();
   havBloc=0;
   blocInFront=1;
}

void centerBloc(){
  int i=0;
  int d=0;
  int j=0;
  int k=0;
  int m=0;
  int n=0;
  int p=0;
  int q=0;
  int r=0;
  int xx=0;
  int exit=0;
  stay();
while(i<1000){i=i+1;}
  servo_deg1(45.0);
  servo_deg2(150.0);
while(m<1000){m=m+1;}
  poke(0x7000,0x70);
while((analog(irbb)>105) && (exit==0) && (blocInFront==1)){
  poke(0x6000,0x50);
  IR2=analog(7);
  poke(0x6000,0xa0);
  if((analog(7)>105) && (IR2>105)){
    poke(0x7000,0xff);
    turnaround();
    goStraight();
    blocInFront=0;
    exit=1;
    servo_deg2(170.0);
    servo_deg1(150.0);
    while(xx<2000){xx=xx+1;}
  }
  else{
      poke(0x6000,0x50);
      IR2=analog(7);
      poke(0x6000,0xa0);
      if((analog(7)<105) && (IR2<105) && (blocInFront==1)){
        goStraight();
        while(j<100){j=j+1;}
        stay();
        while(n<500){n=n+1;}
        j=0;
        n=0;
            if(analog(irbb)<105){
              blocCentered=1;
              blocInFront=0;
              poke(0x7000,0xff);
```

```
             }
         }
         else{
               poke(0x6000,0x50);
               IR2=analog(7);
               poke(0x6000,0xa0);
               if((analog(7)>=105) && (IR2<=105) && (blocInFront==1)){
              motor(0,0.0);
                 motor(1,-100.0);
              while(k<100){k=k+1;}
                 stay();
                 while(p<500){p=p+1;}
                 k=0;
                 p=0;
               }
               else{
                   poke(0x6000,0xa0);
                   IR5=analog(7);
                   poke(0x6000,0x50);
                   if((analog(7)>=105) && (IR5<=105) && (blocInFront==1)){
                 motor(0,-100.0);
                     motor(1,0.0);
                 while(r<100){r=r+1;}
                     stay();
                         while(q<500){q=q+1;}
                         r=0;
                         q=0;
                       }
                       else{
                           turnaround();
                           exit=1;
                           blocInFront=0;
                   }}}}}
     stay();
     blocInFront=0;
  }

  void objectLeft(){
     int j=0;
     int k=0;
     int m=0;
     poke(0x6000,0xc0);
     while((analog(7)<=100) && (j<12000)){
          motor(1,100.0);
          motor(0,-100.0);
          while(k<400){k=k+1;}
```

```
             stay();
        while(m<2000){m=m+1;}
            k=0;
            m=0;
   }
   IR2=0;
   IR3=0;
}

void objectRight(){
   int j=0;
   int k=0;
   int m=0;
   poke(0x6000,0xc0);
   while((analog(7)<=100) && (j<12000)){
        motor(0,100.0);
        motor(1,-100.0);
        while(k<400){k=k+1;}
             stay();
        while(m<1000){m=m+1;}
            k=0;
            m=0;
        j=j+1;
   }
   IR3=0;
   IR5=0;
}

void notFront(){
   int i=0;
   int k=0;
   poke(0x6000,0x00);
   IR0=analog(7);
   poke(0x6000,0x90);
   if((analog(7)>105) || (IR0>105)){
     stay();
     while(i<350){i=i+1;}
     objectLeft();
   }
   else{
       poke(0x6000,0x60);
       IR7=analog(7);
       poke(0x6000,0x70);
       if((analog(7)>105) || (IR6>105)){
         stay();
         while(k<350){k=k+1;}
```

```
      objectRight();
     }
  }
}

void checIR(){
  int i=0;
  int j=0;
  poke(0x6000,0xc0);
  poke(0x7000,0xff);
  if(analog(7)>=100){
    blocInFront=1;
    goBac();
    while(j<1000){j=j+1;}
    stay();
  while(i<500){i=i+1;}
  }
  else{notFront();}
}

void checBS(){
  if(analog(frontBS)>100){
    goBack();
    turnLeft();
    goStraight();
  }
  if(analog(rightBS)<200){
    goLeft();
  }
  if(analog(leftBS)<200){
    goRight();
  }
  if(analog(rearBS)>100){
    int i=0;
    stay();
    while(i<2000){i=i+1;}
    goStraight();
  }
  else{goStraight();}
}

void startUp(){
  blocInFront=0;
  blocCentered=0;
  numStacs=0;
  havBloc=0;
```

```
   blocStacabl=0;
   servo_deg1(150.0);
   servo_deg2(170.0);
   servo_on();
   stay();
   poke(0x7000,0xff);
}

void main(){
   int go=0;
   startUp();
   while(go<4000){go=go+1;}
   goStraight();
 while(1){
   while((blocInFront==0) && (numStacs<4)){
            checBS();
        checIR();}
   stay();
   while((blocInFront==1) && (havBloc==0) && (numStacs<4)){
        centerBloc();}
   while((blocCentered==1) && (numStacs<4)){
     grabBloc();
     stay();}
   while((blocInFront==0) && (numStacs<4)){
            checBS();
        checIR();}
   stay();
   while((blocInFront==1) && (havBloc==1) && (numStacs<4)){
        centerBloc2();}
   stay();
   while(numStacs==4){
        bow();}
   stay();
   while(numStacs==5){stay();}
/*   while(blocStacabl==1){
            stacBloc();}*/
 }
}
```