INTELIGENT MACHINES DESIGN LABORATORY

FINAL REPORT

ARISTARCHUS

JULIO C. CORREA

UNIVERSITY OF FLORIDA

April 2000

Abstract

This report summarizes the design process for the mobile robot Aristarchus. Both mechanical and electrical aspects are considered. Also it is explained in detail one of the most significant features of this robot that is the attention given to the details regarding to the distribution of parts and to the manufacturing of the robot.

1. INTRODUCTION

The mission of Aristarchus is to find a ball inside an arena an after that to place it in a goal. The design of a device able to perform the desired task was designed considering that any solution should fulfill the following criterions:

- Be simple
- Be compact
- Be reliable
- Good appearance

The first criterion implies that only very basic components should be used for the robot platform and for the sensors. Doing this the cost and the maintenance of the system improves, but also means that is not possible to expect many complexes behaviors due to the limitations in the capabilities of the parts.

Compactness is an attribute of any good design. It is related to the consumption of power. The small size however makes more difficult any change in the robot due to the limitation of space. The labor of the designer is increased to distribute all the parts in the reduced space.

Reliability is essential for any device. Basically means that the system is able to work at any time that it is required. This restriction requires that the machining of the mechanical parts and all the individual components be strength enough to avoid failures due to the weakness of the robot.

Finally the aesthetic in extremely important in any object that the public is going to observe. The experience shows that only people with technical training may realize the effort behind a device even if looks simple, but this is not true for people without that vision. As a result many good works are neglected for their appearance. Because of this, and to create a pleasure impression since the beginning, the design of Aristarchus considers strongly the way it looks.

The general criterions mentioned are present in any decision about any part of the robot as it can be seen in the following sections.

2. MECHANICAL PLATFORM

The body of the robot is presented in Figure 1. Details related to the gripper and bumper sensors are not illustrated there to make easier the comprehension. Instead of more refined manufacturing process involving lathes and mills, the robot was built using only the T-machine and the driller. In this way all the modifications were easy to do at the lab due to the availability and simplicity of these resources. Some parts of Lego were included and the joining between parts is accomplished by the use of screws

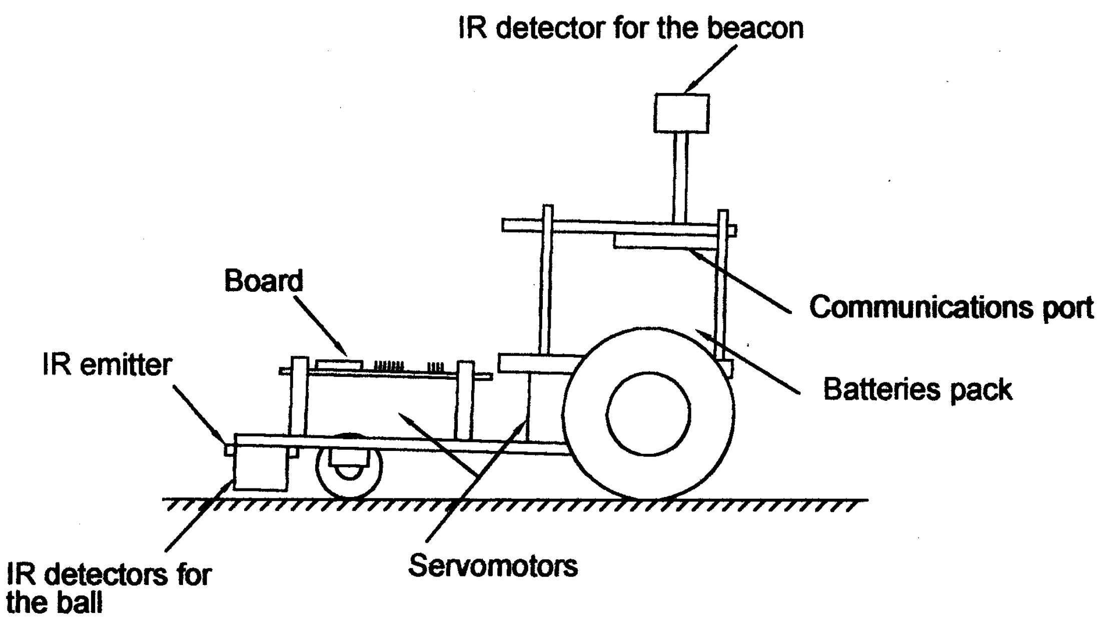

Figure 1. Layout of Aristarchus

3. BEHAVIORS

Aristarchus posses 4 behaviors to fulfill its task:

- To search for the ball: once the motion begin the first task consists on look for the ball inside the arena. This is accomplished by motions in straight line that change their directions when the robot reach a border and by the detection of an IR signal by the sensors located at the bottom of the machine. See Figure 2.

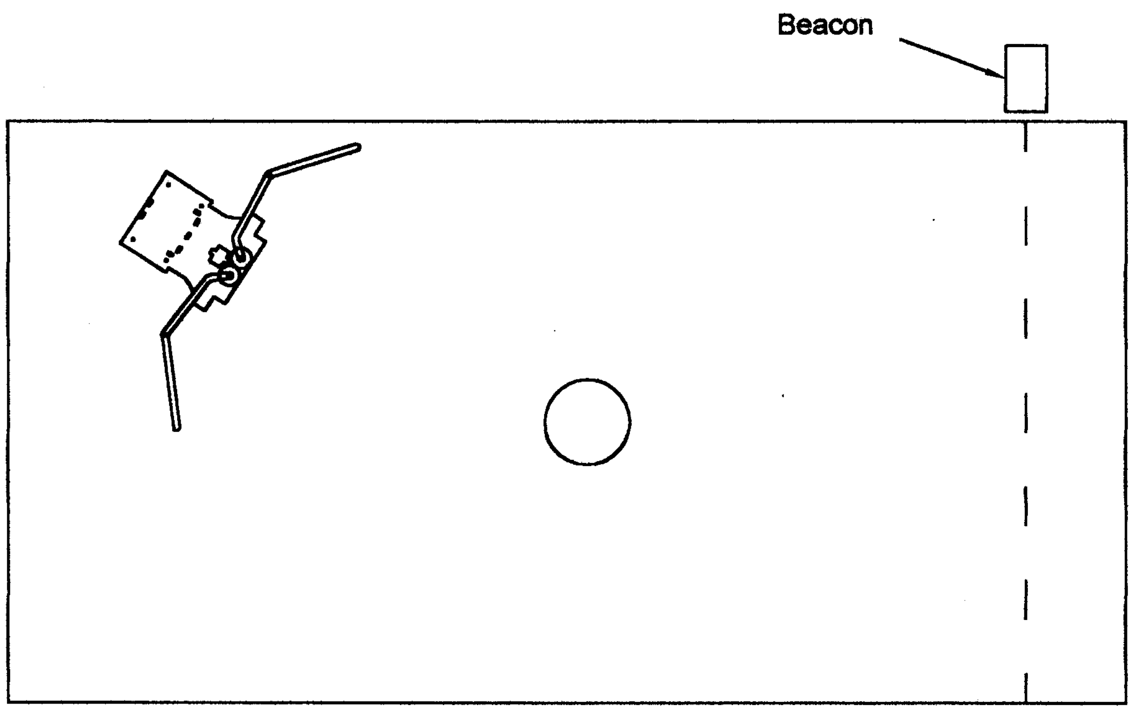

Figure 2. First stage: looking for the ball.

- To catch the ball: after the signal in the bottom IR receptors is increased over certain value, this means the ball is just in front of Aristarchus and the order to close the gripper is given. See Figure 3.

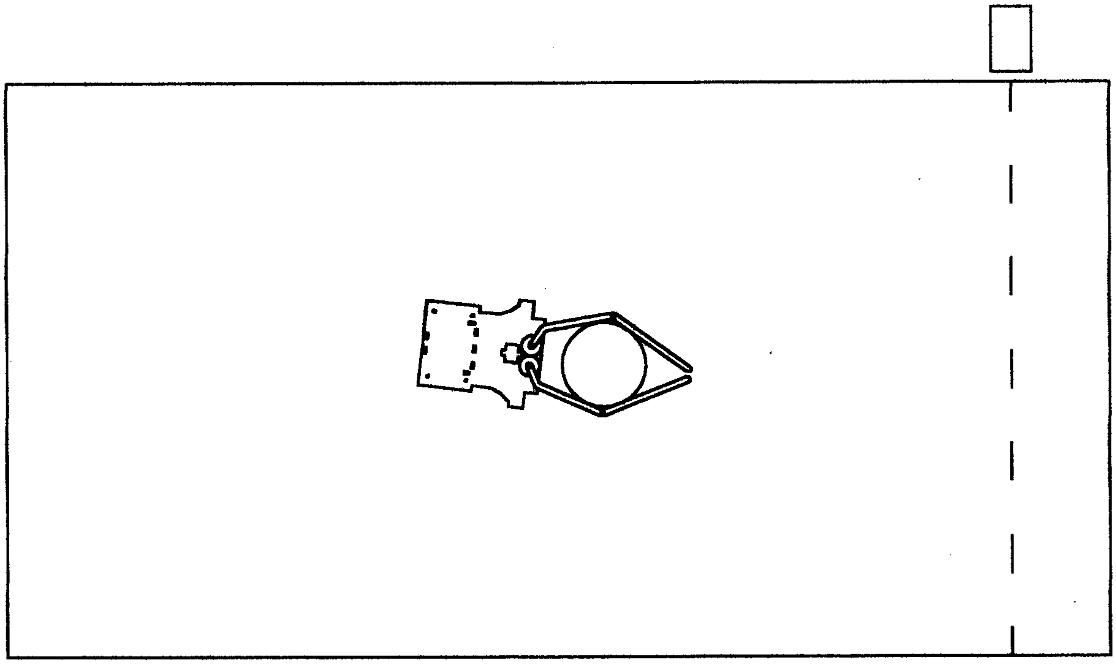

Figure 3. Second stage: catching the ball.

- To place the ball at the goal: The next step is to look for the goal. The process consists of non-programmed motions until the signal of a lateral beacon is detected by IR receptors located in the top of the platform. This information indicates to the robot that the next time that a bumper sensor is activated is because the robot reaches the goal and the task is fulfilled. See Figure 4.

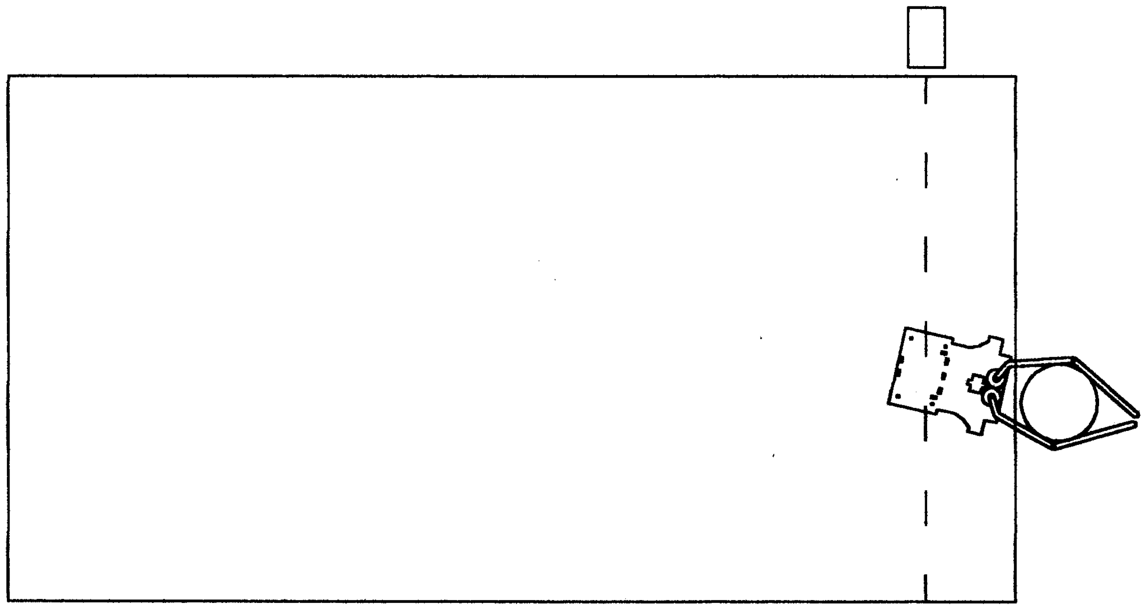

Figure 4. Stage 4: placing the ball at the goal line.

- To keep inside the limits of the arena: the robot cannot abandon the limits of the field. This is accomplished by two front bumper sensors that detect when Aristarchus touchs the borders of field. Immediately the robot reverses and rotates to avoid the obstacle.

It is apparent that the first three behaviors are in sequence, but the last one takes place at any moment the robot reach the limit of the field. This structure in the order of the behaviors is extremely important at the moment of the development of the software.

5. SENSORS

Being coherent with the criteria of simplicity the robot only posses IR emitters/receptors and bumper sensors. Outside the robot there is a beacon to locate the goal line. They are summarized in Table 1.

Table 1. Sensors for Aristarchus

Type	#	Description
Lower IR emitter/receptors	2	Locate the ball
Upper IR receptors	2	Locate the beacon
Bumper sensors	2	Avoid to go out of the field
Gripper sensor	1	Indicates the status of the gripper
Beacon	1	External to the robot. Locates the goal

The functions of the IR sensors and bumpers sensors were explained before. The gripper sensor indicates if the gripper is open or closed and in this way indicates indirectly if the ball is caught or not.

The beacon emits IR radiation. The frequency can be changed easily, but due the only IR receptor available works at 40 kHz, this was the frequency selected for the beacon. Its circuit was suggested in another reference and is shown in Figure 5.

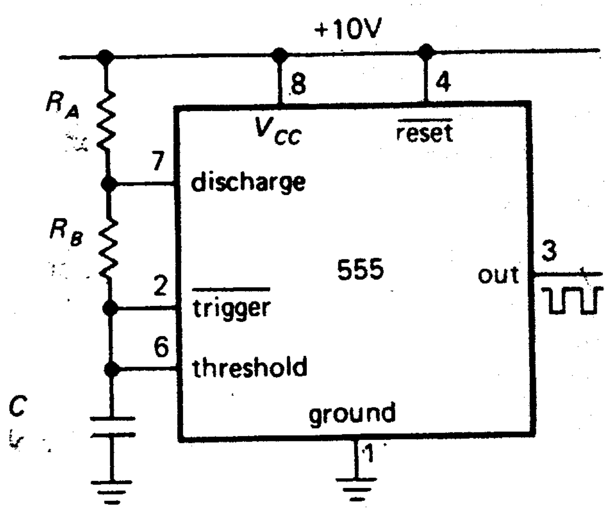

Figure 5. Circuit for the beacon.

The period of oscillation of the output follows the equation

$T = 0.693(R_A + 2R_B)C$

where T is in milliseconds, R in k and C in F. Many combinations are possible to obtain the same period. For the current circuit $R_B = 0.681$ k , $R_A = 0.419$ k and $C = 0.02$ F, hence $T = 0.0246ms$ or $f = 40kHz$.

Three IR emitters in series were placed at the output together with a resistor of 380 to limit the current in the leds to 2.5 mA. Both values were chosen experimentally in such a way that limit the intensity of the IR signal to a value adequate for the field. Other situations would require different calibrations.

6. ACTUATORS

Aristarchus only need the three actuators presented in Table 2. The servomotors for the wheels are hacked but the servomotor for the gripper conserves its original configuration to be able to open and close the gripper. Since all the actuators are non-linear they were calibrated to assure their proper operation on the field

Table 2. Actuators for Aristarchus

Type	#	Description
Servomotor	1	Drives the right wheel
Servomotor	1	Drives the left wheel
Servomotor	1	Drives the gripper

7. MECHANICAL DETAILS

Once a general idea has been conceived, there is still a big gap between the theoretical conception and its implementation in a physical device, and the difference between the success or the failure of a system is not in the general conception but on the details.

As can be inferred from the sensors, Aristarchus is extremely simple in its electronic part which is not a disadvantage to carry out its mission, but the real design effort for this mobile robot is in its mechanical parts. Each detail was considered carefully to be assure the general performance.

Problems due to the details can be classified roughly in expected and unexpected. The first ones permits to take the actions in advance but the second ones appears only in the assembly process or when the robot is running. Generally are difficult to find a good solution for them and are the responsible of the bad performance.

Due to their practical nature are difficult to describe in a technical paper, however some of them are reported here to illustrate this important but hidden step.

Platform.

Figure 6 illustrates the shape of the platform. Although its construction is simple, it includes the space enough to bear the sensors and the actuators. The holes permit to the cables to cross the platform and some of them define the exact location of the parts. In many cases the distances between components is less than 1 mm, to reduce the size of the robot as much as possible.

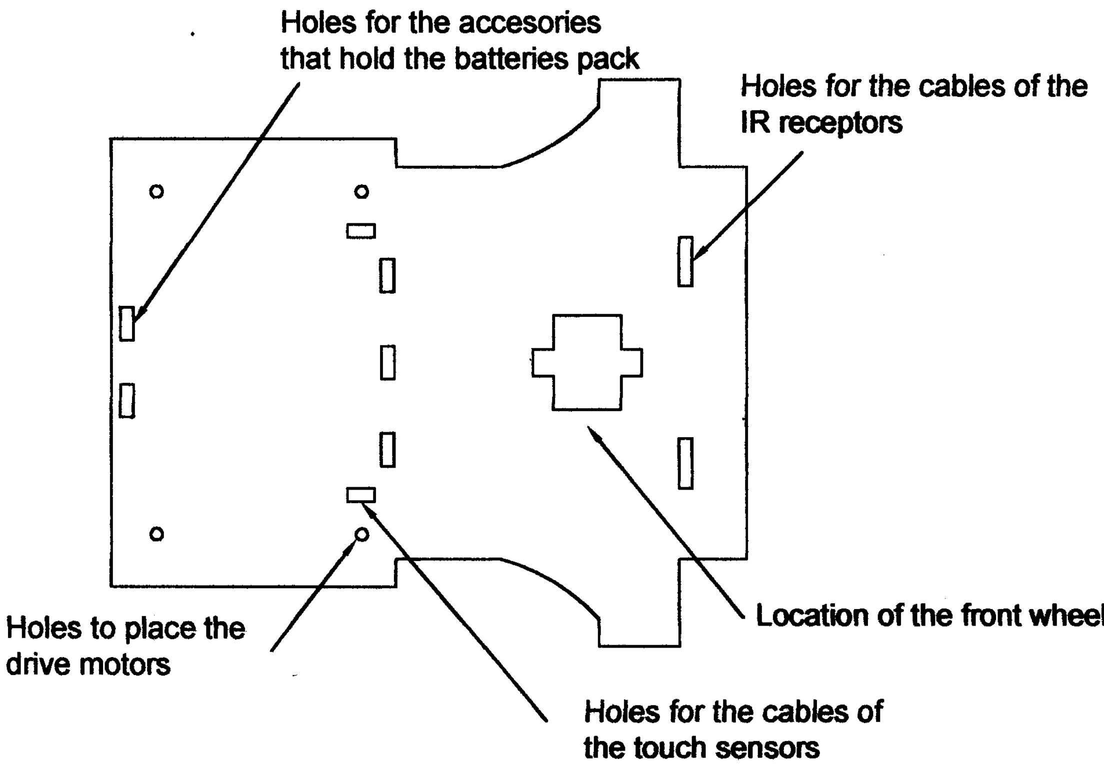

Figure 6. Details of the platform.

Bumper sensors.

They must detect the borders of the field which height is about 3 mm, but at the same time they cannot touch the ground. Hence they require a careful calibration of their heights. Also a counterweight was added to keep down the sensor at any time. See Figure 7.

Front IR receptors

For any reason these sensors should touch the borders. Also as their function is to detect the radiation reflected only for the ball, they must keep the proper height to avoid any reflection coming from the borders. The height is provided for the front wheel and is critical for the proper performance of the system. See Figure 7.

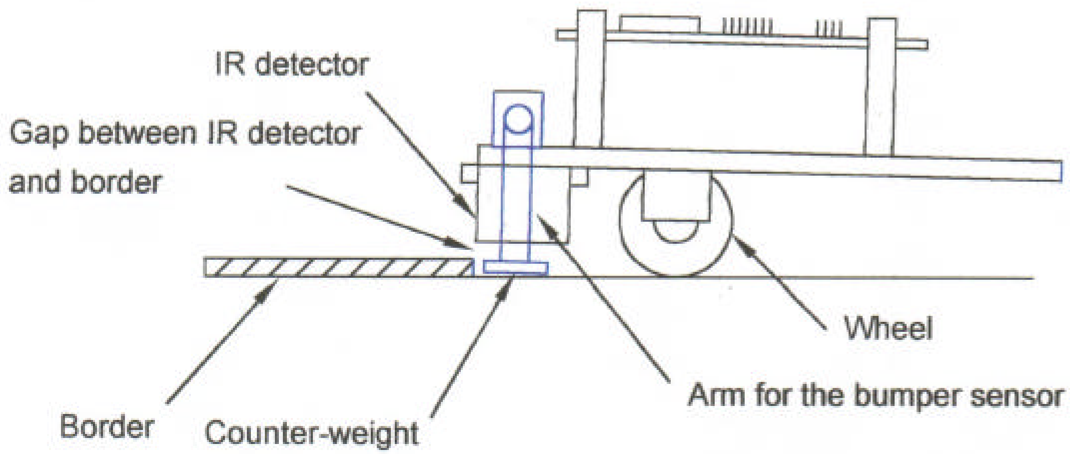

Figure 7. Detail of the bumper sensor and the lower IR detector.

Gripper.

If the arms of the gripper would be rigid, there is the risk that the gears than drive the gripper slip at the moment of catching the ball and therefore the synchronization would be loss. To avoid that risk each arm was divided in two parts with an elastic join between them. In this way, no matter the fashion the gripper catch the ball, the mechanical device absorbs the misalignment and no further problems arise. See Figure 8.

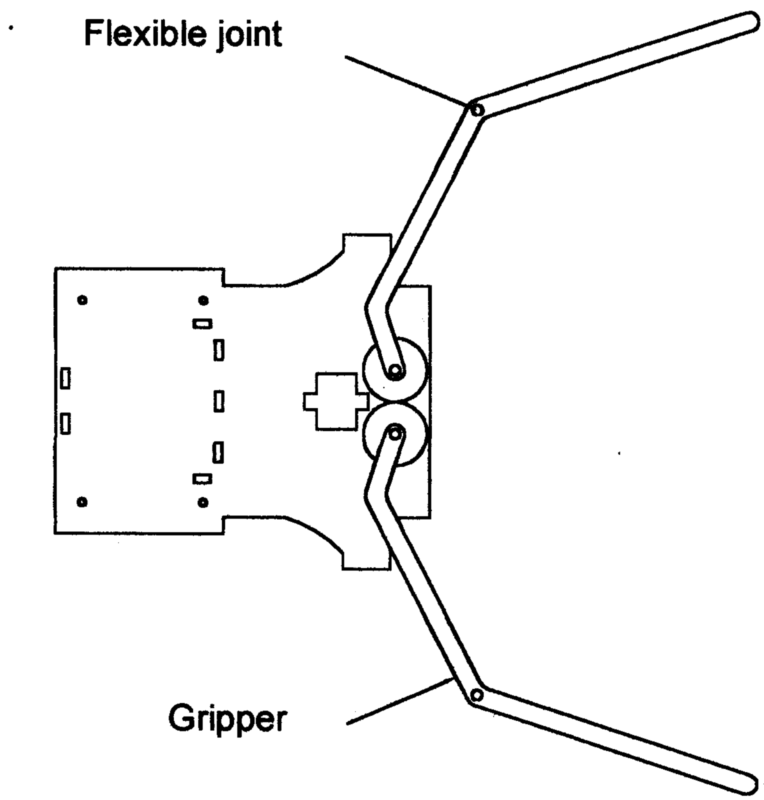

Figure 8. Detail of the gripper.

Borders

The borders of the arena are built in short slender parts which can be easily assembled using joints similar to those used by a puzzle, as it can be appreciated in Figure 9.

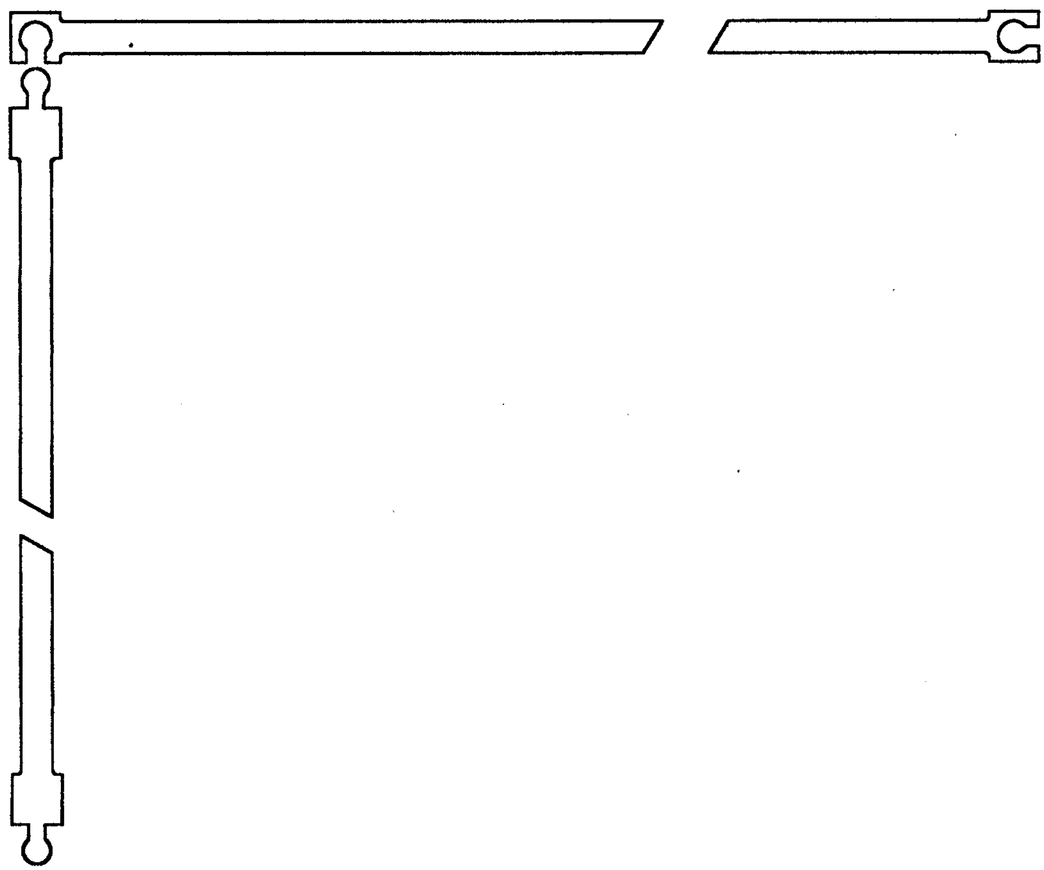

Figure 9. Detail of the border segment joints.

8. CONCLUSIONS

Aristarchus passes all the tests and its performance is satisfactory according to the initial conception. Because to search the ball is not an entirely a controlled behavior but also there is some random in this process, the time to locate the ball could be longer in some cases.

Due to its design, it is not a closed project. Other behaviors can be added just modifying the software. For example to include more strict rules about the way Aristarchus reach the goal is an interesting alternative. Also to have more than one robot in the same field competing between them would increase the attention of the spectators.

The capabilities of Aristarchus are limited to the number and simplicity of its sensors. If they could be improved at reasonable costs, the same structure could be adapted for another tasks, like to pick objects in an open field.

Robots are both mechanical and electric. A good mechanical design simplifies significantly the electronic design and enhances the performance of a robot.

Once an idea is conceived is important to pay attention to the details to guarantee the success of the design. They cannot be neglected at any moment.

All the design criteria established at the beginning of the project were satisfyed.

APPENDDDIX

SOFTWARE FOR ARISTARCHUS

```
/***************************************************************************
*                                   ARISTARHUS                    *
                     *
 ***************************************************************************/
/**************************** Includes **********************************/
#include <tjpbase.h>
#include <stdio.h>

/************************* End of Includes *****************************/

/**************************** Constants *********************************/
#define SCG 118
#define LEFT_MOTOR 0
#define RIGHT_MOTOR 1
#define CRUISE_RGHT 92
#define CRUISE_LEFT -10
#define noise 100
#define top_noise 100
#define smin_l -40
#define smin_r 61
#define smin_rb 62
#define smin_lb -35
#define MinTopSignal 128
#define MinReqBotSig 100
#define StopSignal 100
/************************* End of Constants *****************************/
void main()
/****************************** Main ***********************************/
{
  int bump, irdr, irdl, irtr, irtl, speedr, speedl, touch, DetectBeacon, Steps, Grasp;

  init_analog();
  init_motortjp();
  init_clocktjp();
  init_servotjp();

  *(unsigned char*)0x7000=0xff; /* IR turned on*/
```

```
servo(1,900); /* Initial position gripper*/
wait(500);
servo(1,2500);
wait(500);
servo(1,900);
wait(500);
servo(1,2500);
wait(500);


Steps=0;                  /* Initialization parameters*/
DetectBeacon=0;
Grasp=0;
touch=0;
speedr=CRUISE_RGHT;
speedl=CRUISE_LEFT;

while(1)


{
    irdr=analog(2);        /* Reading of sensors*/
    irdl=analog(3);
    irtr=analog(5);
    irtl=analog(4);
    bump=analog(0);

    speedr=CRUISE_RGHT;
    speedl=CRUISE_LEFT;
                               /* Commands if the robot touchs the border*/
    if((DetectBeacon==0)&&(bump>20)&&(bump<30))
        {
         speedr=100;
         speedl=-100;
         touch=1;
        }
    else if((DetectBeacon==0)&&(bump>40)&&(bump<50))
        {
         speedr=-100;
         speedl=100;
         touch=1;
        }
    else if((DetectBeacon==0)&&(bump>85)&&(bump<95))
        {
         speedr=100;
         speedl=-100;
         touch=1;
        }
    else if((DetectBeacon==0)&&(bump>100))
        {
```

```
            speedr=-100;
            speedl=100;
            touch=1;
           }
else if((DetectBeacon==1)&&(bump>10))
           {
            servo(1,2500);
            speedr=-100;
            speedl=-100;
            touch=2;
           }

                                                      /* Detection of the ball*/
else if((irdr>irdl)&&(irdr>noise)&&(bump>75))
           {
            speedr=smin_r;
            speedl=smin_r+(irdr-irdl);
            touch=0;
           }
else if((irdl>irdr)&&(irdl>noise)&&(bump>75))
           {
            speedr=smin_r+(irdl-irdr);
            speedl=smin_l;
            touch=0;
           }
else if ((Grasp==1)&&(DetectBeacon==0)&&(irtl>=128))
           {
            DetectBeacon=1;
}

                                   /* The gripper is activated*/

if((irdl>SCG)&&(irdr>SCG))
           {
            servo(1,900);
            Grasp=1;
           }
                                   /* Arbitror for the motions of the robot*/
if(touch==0)
           {
            motorp(RIGHT_MOTOR,speedr);
            motorp(LEFT_MOTOR,speedl);
            wait(10);
           }
else if(touch==1)
           {
            motorp(RIGHT_MOTOR,-100);
            motorp(LEFT_MOTOR,-100);
            wait(500);
            motorp(RIGHT_MOTOR,speedr);
            motorp(LEFT_MOTOR,speedl);
```

```
                wait(900);
                touch=0;
                }

        else if(touch==2)
                {
                        while(Steps<2)
                        {
                         motorp(RIGHT_MOTOR,speedr);
                         motorp(LEFT_MOTOR,speedl);
                         wait(700);
                         servo(1,900);
                         wait(500);
                         servo(1,2500);
                         wait(500);
                         Steps=Steps+1;
                        }
                 motorp(RIGHT_MOTOR,0);
                 motorp(LEFT_MOTOR,0);
                 wait(5000);
                 touch=6;
                }
}
}
/**************************** End of Main ******************************/
```