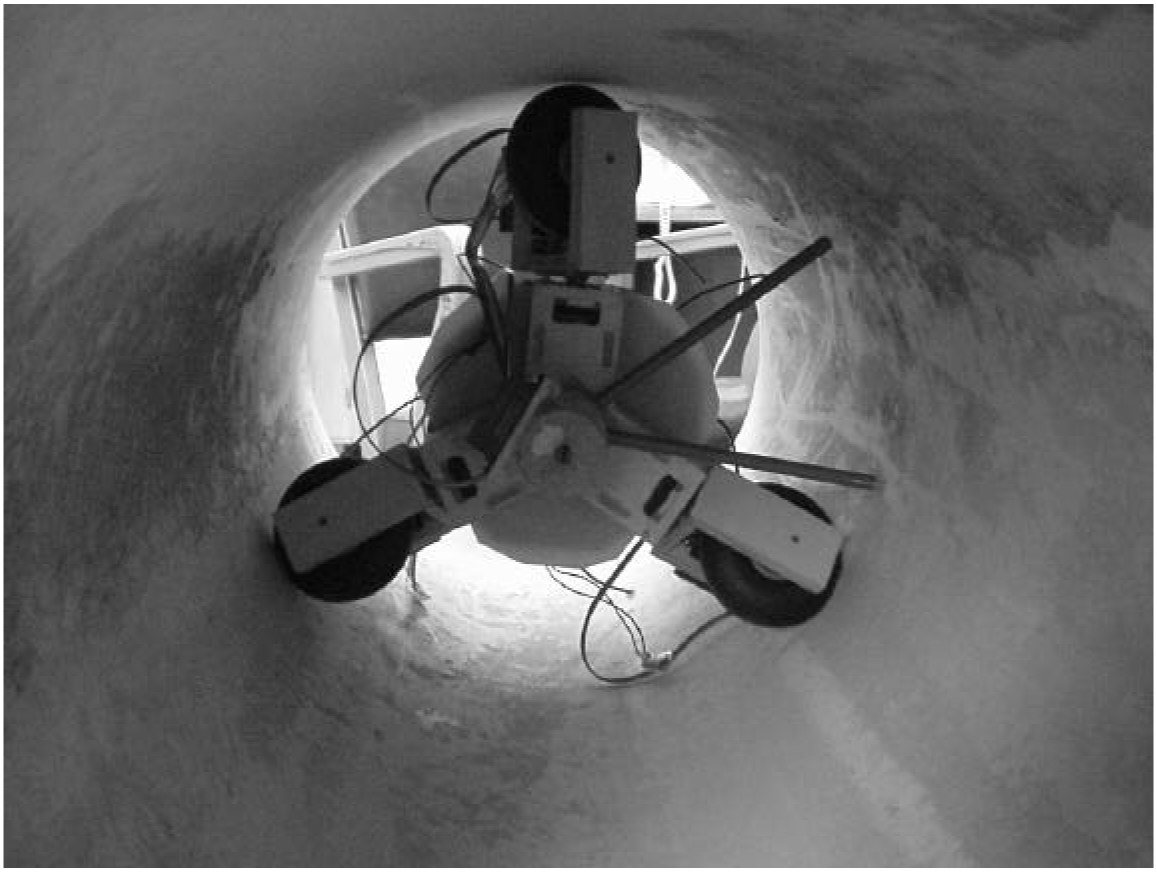
OPCR
(OH's Pipe Cleaning Robot)



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

By: Young Hoon OH

Date: April 23, 2002

Table of Contents

Abstract

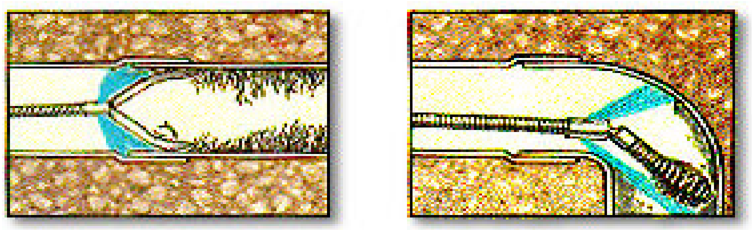
The role of OPCR (OH's Pipe-Cleaning Robot) is simple. It automatically finds the obstacle, runs back, changes the pitch of the wheel and cleans it. When it does not find any more obstacles, then change the pitch to the run mode again and run until the end of the pipe. When the robot reaches the end of the pipe, its work is done- that means it stops-. The sensors are used for finding the obstacles and sensing the end of the pipe. The test was successful and all the problems are solved.

Executive Summary

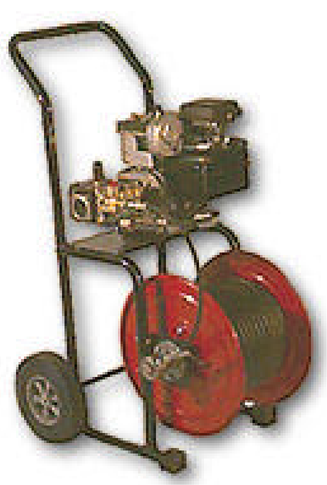
The OPCR runs with a helical motion inside the 12-inch pipe. The merit of the helical movement is that the OPCR doesn't have to have a device to operate the cleaning device such as scraper or cutter. That is, the motion of the robot can eliminate all the obstacles without any other particular operation. Another superb thing of the OPCR is that it can change the pitch of the motion. If one wants to eliminate the obstacles in detail, the OPCR can do the command with a smaller angle of the wheel. The small number change in program can decide whether the obstacle remover is needed for detailed work or not.

Introduction

Though there is a commercially available pipe-cleaning machine (i.e. Marco Scope Inc. see Figure 1 and 2), still we need to develop self-mapping and cleaning up and painting inside the long-range, big pipe regularly. For the maintenance and research purposes, the new type of pipe-cleaning robot is proposed. After this development, conventional cleaning technique can be adapted to this machine (i.e. gas emitter or painting).

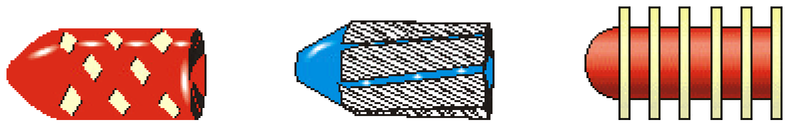


<Figure 1: Marco Set Snake Combo>



<Figure 2: MG-2200 on MC-1525 Cart Carrier >

Another example for cleaning the pipe is shown in Figure 3. Those models are called 'pipe-cleaning PIG' and can travel inside the pipe. The noticeable thing is that those were designed to travel long range. For the example shown in the Figure 3, it can travel to the range of 25 miles. One advantage of the OPCR is that it can change its pitch depends on the size of the obstacles.



<Figure 3: Ryan Process - Standard Red Series (25 miles/41 kilometers in length)>

Integrated System

OPCR composed of three parts: Head Unit, Driving Unit, and Stability Control Unit. Its head contains the sensory system that can detect the obstacles that should be cleaned up and also find the end of the pipe for the robot to stop its motion. The structure of the head unit is composed of cutter (or scraper) to cut the obstacle out, and the sensors (one for finding obstacles and one for detecting the end of the pipe). The driving unit has only two purposes, which are driving the robot properly and changing the pitch of the wheels depends on the size of the obstacles. It contains six motors to control whole body. Three motors attach to three wheels to drive and other three motors change the angle of the individual wheel. Since the robot moves forward in a helical way, the wheel of the robot must change its angle to move. It will move forward fast when the angle of the wheel increases, and for the slow motion, the OPCR needs to change to the smaller angle. The stability control unit will act passively. When the driving unit fails to control its stability of aligning the geometric center of the robot to the center of the pipe, the support control unit gave the stability to hold this robot. The stability control unit composed of four struts stretched out to the inside of the pipe.

OPCR has relatively simple control: Find the obstacle, clean it, and drive through until it gets to the end of the pipe. The flow chart is shown on Appendix A.

Mobile Platform

OPCR specification:

Head Unit

Diameter: 5.544 inches

Height: 2.25 inches

Driving Unit

Individual Strut Length

(Including wheel housing, angle control motor): 3.475 inches

Diameter: 5.544 inches

Height: 3.4 inches

Stability Control Unit

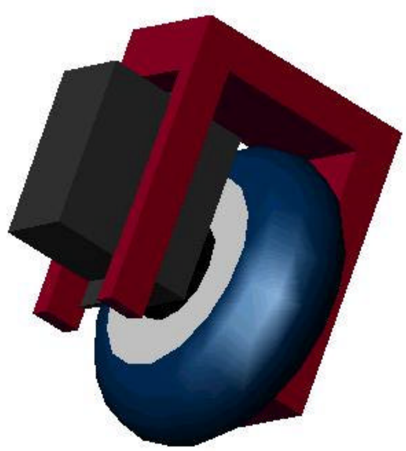
Diameter: 5.544 inches to 2.4 inches

Height: 3.4 inches

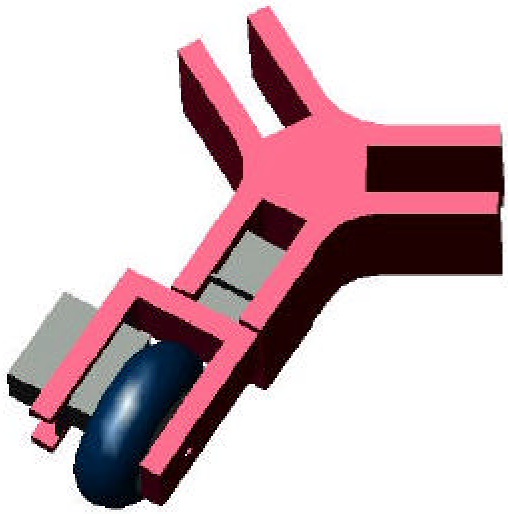
Actuation

OPCR needs six actuations as mentioned earlier. Three of them are used to drive the wheels and others are used as an angle controller of the individual wheel. Six 42 oz-in motors are chosen for this robot. Since the size of the robot is relatively small compared to the 'real' robot that can be useful in the 'real' situation, the inertia effect of this model will be neglected (Friction could be the most powerful source of driving).

Figure 4 and 5 show the position of the motor for individual strut.



<Figure 4: Wheel Housing (red), Motor (gray), and a Wheel (Blue)>



<Figure 5: Basic Structure of the Driving Unit>

Sensors

There will be used only two types of sensors: one that can sense the obstacles and other one for sensing the end of the pipe. For the first one, there are two choices: force sensor and near-infrared range sensors. For the second one, using near-infrared sensor will be applicable. The final decision was made before the test setup preparation period. IR sensor was chosen for sensing obstacles. For sensing the end of the pipe, bump sensor was used.

Behaviors

The first goal of this robot is to find and clean the obstacles while traveling inside the pipe. It will travel through the pipe with a helical motion. While moving at a fast speed, it will find the obstacles with a sensor. Once it finds the thing to remove, it will back off a little (2 inches) and change the angle of the wheel. Then it will rotate with a same speed, but will move forward slowly because the angle of the wheel is smaller. When it finds no more obstacles, then it stops and changes the angle of the wheel again to the original (run-mode) angle and runs again. The second goal of the robot is to find the end of the pipe. When it closes to the end of the pipe, then it stops or comes back to the original position depends on the program. My mission for this experiment is to stop at the end of the pipe.

One other thing that this robot can do is running through the tilted pipe. The experiment will be held on the straight pipe, but if the design of the robot can be improved, the robot will turn a little bit while moving. However the design will be improved, it still turns in a limitation.

Experimental Layout

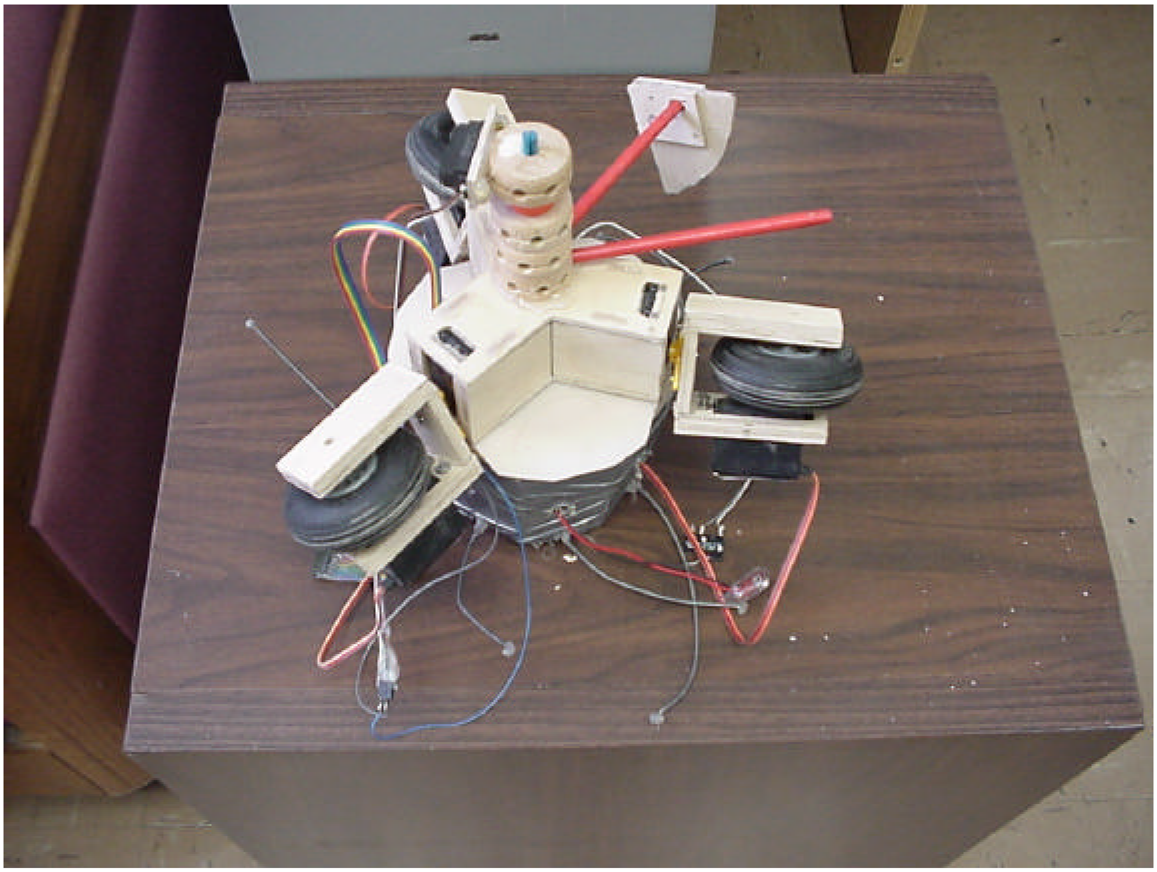
The hardest thing to do the experiment of this robot is finding a proper pipe to test it. In the light of the design, the actual size of the pipe to test this robot will be minimum diameter of 12.5 inches. The developer found and is helped from the Watson Construction Co. to get a pipe. They donated the 12-inch diameter pipe to the developer (See Figure 6).

<Figure 6: The Pipe>

The Problems

There were several problems occurred when designing and performing the OPCR. Firstly, while choosing the experimental setup, miscalculation of the pipe size was found. The pipe itself should have 12-inch diameter, however, actual inside diameter is less than 12-inch. It has a wall of .5 inches. Thus, the outer diameter of the OPCR should be resized. The problem was solved with TA (Aamir)'s help. To reduce the outer diameter of the OPCR, the inner springs were cut of .1 inch each. Also, the outer wheel housing was reduced its size of .125 inches each.

Secondly, the number of the output port for the servos is not matched with the servos to be controlled for the OPCR. As mentioned earlier, the number of the driving robot is six; three for the driving the wheel and three for the angle control. The board chosen for the OPCR was MTJPRO11A. The problem was solved with connecting three servos in one port. Connecting three servos with using only one port was decided because only one signal is needed for controlling three servos. The complete body is shown in Figure 7.



<Figure 7: The Whole Body>

Conclusion

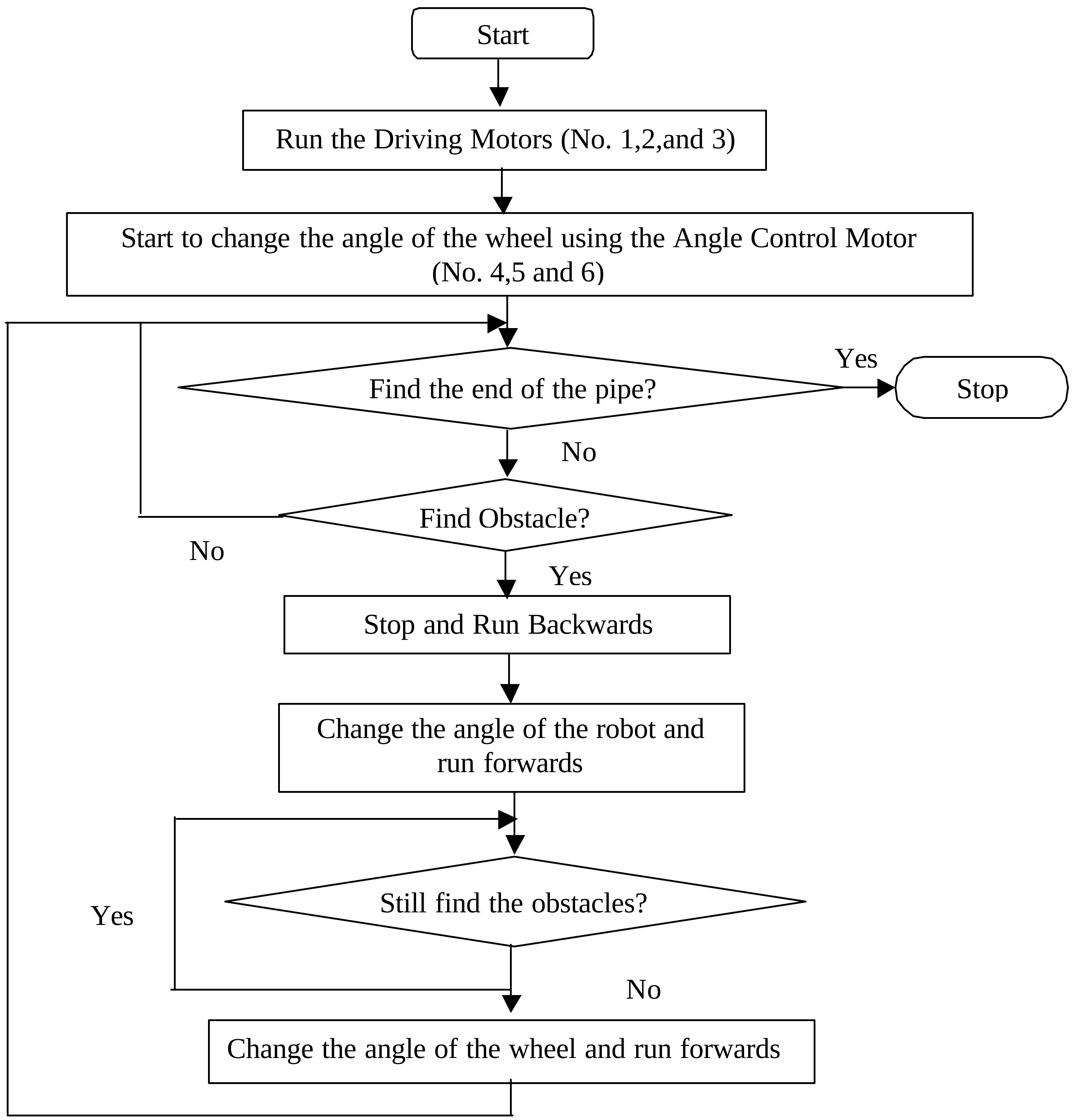
The OPCR was built and tested successfully. The pitch control was well done by changing the angle of the wheels. On the demo day, the OPCR runs with high pitch angle, however, it can also travel through the pipe with a small angle. The improvement can be made through the installment of two more IR sensors when it moves fast. Because of the high pitch angle, sometimes the one IR sensor cannot detect the obstacles. With changing the pitch angle to the lower one, the IR sensor in the OPCR can detect the obstacles well. Another thing to improve is turning ability. The OPCR cannot run with a curved pipe. If another developer wants to modify this robot, one can think that other choice of the stabilizer part and the individual angle controller could be useful. Or if one wants to develop smaller pipe-cleaning robot, in the light of this robot, using gears to control the angle of the wheel is recommendable. Using smaller wheel is strongly recommended.

Acknowledgement

The author thanks to Watson Construction co. for donating the pipe. Also thanks to Scott Shamblin who gave the initial idea of the OPCR, Aamir Qaiyumi (TA) for helping to solve all the interferences problems, and J.P. Clerc for letting me uses his parts, ideas and materials.

Appendix A

<Flow Chart for OPCR>



Appendix B (The Program)

```
/************************************************************************
* Title: OPCR.c *
* Programmer: Young Hoon OH {from code by Ivan Zapata & K. Doty & Yoh}*
* Date: April 16 2002 *
* Version: 1.2 *
* *
* Description: *
* The robot will wait for the bumper to be sensed and turn on *
* Three outer motors turn the wheel to move
* Three inner motors control the angle of the outer motors.
* When the front IR sensors find the obstacles,
* the OPCR will stop and turn to 2-3 inches back.
* The inner motors then turn to the small angle
* After turning, the outer motor start to turn forward.
* This gives OPCR to move with slow forward speed, but same rotation
* The OPCR will stop when it finds no longer obstacles.
* Then it changes to the larger angle and move forward.
* The OPCR will stop when it closes to the end of pipe.
*
* Note:
*  motormask[0] = 0x80;          Drives PA7, left motor,  0 - outer in OPCR
*  motormask[1] = 0x08;          Drives PA3, right motor, 1 - inner in OPCR
*  LEFT_MOTOR      0
*  RIGHT_MOTOR   1
*  Set PA3 (OC5) and PA7 (OC1) to output
*
*  Bumper Value : 1 --- 127
*                 2 --- 71
*                 3 --- 43
*                 4 --- 21
*
*  IR sensor value : 70  (fixed)
*            120 or over (Obstacles)
*
************************************************************************
/
/************************** Includes *******************************/
#include <analog.h>
#include <motortjp.h>
#include <clocktjp.h>
#include <isrdecl.h>
#include <vectors.h>
#include <stdio.h>
```

```
/************************* End of Includes *****************************/
/*************************** Constants ********************************/
#define OUTER_MOTOR 0
#define INNER_MOTOR 1
#define MAX_SPEED 100
#define ZERO_SPEED 0
#define BUMPER analog(0)
#define RIGHT_IR   analog(2)
#define OBSTACLE 100
#define IRE_ON  *(unsigned char *)(0x7000) = 0x07
#define IRE_OFF       *(unsigned char *)(0x7000) = 0x00
#define START         while(BUMPER<120)
#define FRONT_BUMP  (BUMPER>10)&&(BUMPER< 120)
#define BACK_BUMP  BUMPER>120



void turn(void);

void main(void)

{
int i, ir;

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

IRE_ON;
START;

        /*while(BUMPER < 120);*/
        while(1)
        {
                ir=RIGHT_IR;

                motorp(OUTER_MOTOR, MAX_SPEED);
                motorp(INNER_MOTOR, 160);

                if (ir < OBSTACLE)
                {
                        motorp(OUTER_MOTOR, ZERO_SPEED);
                        wait(400);
                        motorp(OUTER_MOTOR, -MAX_SPEED);
                        wait(1500);
```

```
                motorp(OUTER_MOTOR, ZERO_SPEED);
                wait(400);
                motorp(INNER_MOTOR, 90);
                wait(400);
                motorp(OUTER_MOTOR, MAX_SPEED);
                wait(2000);

                motorp(OUTER_MOTOR, ZERO_SPEED);
                wait(400);
                motorp(INNER_MOTOR, 160);
                wait(400);
                motorp(OUTER_MOTOR, MAX_SPEED);
            }
            if ((BUMPER > 60) && (BUMPER < 90))
            {
                motorp(OUTER_MOTOR, ZERO_SPEED);
                wait(100000);
            }
        }
}
/*************************** End of Main *****************************/
```