**Test Cases for AVC Project - Initial Program**

* The robot must run the course with two motors each controlling the wheels on either side, but one motor is more powerful than the other
* The robot must run the course that has turns but the robot has difficulty when trying to turn
* The robot must run the course with bends and turns while moving at a very fast speed
* The robot must run the course without knowing how the course is laid out
* The robot must run the course without knowing where the finish will be
* The robot must run the course while trying to dodge and manoeuvre around many obstacles

**Test Cases for AVC Project - Challenge**

* For the challenge section of the project the robot should travel through a maze to reach the finish position.
* The robot should move according to turns and gaps in the walls
* The robot must not collide into walls

The code-

* The “loadImageToMatrix” function in the code first loads the image into matrix that the robot can view through its camera.
* The “compressImage” compresses the image into 4 colors. The only colors that the image contains are White, Green, Yellow and Red, hence it is compressed to these colors.
* For each individual pixel, the ratio of red to green and ratio of red to blue are taken.
* The function “pathBlocked” inspects if the path is being blocked by any obstacle or wall.
* The function “turnControl” determines which way the robot will turn depending upon the white line that it should follow.
* The “main” function in the code determines the speed and turns the robot depending on the calculated “error”.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Speed | Turn | Turn | Turn | Turn |
| Right– 5.0  Left– 5.0 | 1.0 – robot begins the maze, does not reach the finish because it cannot recognize a gap in the wall. | 5.0 – robot begins the maze, does not reach the finish because it cannot recognize a gap in the wall. | 10.0 - robot begins the maze, does not reach the finish because it cannot recognize a gap in the wall. | 15.0 – robot begins the maze, does not reach the finish because it cannot recognize a gap in the wall. |
| Right– 10.0  Left– 10.0 | 1.0 – runs the maze, reaches the finish, but time taken is still high. | 5.0 - does not run the maze and does not reach the finish. | 10.0 - does not run the maze and does not reach the finish. | 15.0 – runs the maze, reaches the finish, but time taken is still high. |
| Right– 15.0  Left-15.0 | 1.0 – runs the maze, reaches the finish, time taken is moderate. | 5.0 – runs the maze, reaches the finish, time taken is moderate. | 10.0 - runs the maze, reaches the finish, time taken is moderate. | 15 - runs the maze, reaches the finish, time taken is moderate. |
| Right– 20.0  Left– 20.0 | 1.0 – runs the maze, reaches the finish, time taken is adequate. | 5.0 - does not make any turns at all, does not reach the finish | 10.0 - does not make any turns at all, does not reach the finish | 15.0 – does not turn accurately, reaches the finish, time taken is adequate. |
| Right– 30.0  Left– 30.0 | 1.0 – runs the maze but does not reach the finish. | 5.0 - runs the maze but does not reach the finish. | 10.0 - runs the maze but does not reach the finish. | 15.0 - runs the maze but does not reach the finish. |
| Right– 40.0  Left– 40.0 | 1.0 – runs the maze but does not reach the finish. | 5.0 - runs the maze but does not reach the finish. | 10.0 - runs the maze but does not reach the finish. | 15.0 - runs the maze but does not reach the finish. |
| Right– 50.0  Left– 50.0 | 1.0 – runs the maze but does not reach the finish. | 5.0 - runs the maze but does not reach the finish. | 10.0 - runs the maze but does not reach the finish. | 15.0 - runs the maze but does not reach the finish. |

Using 20.0 for “vRight”, 20.0 for “vLeft ” and 1.0 for “turn” value lets the robot run the maze, reach the finish and adequate amount of time is taken.