Test case for AVC project – core

* For the core part of the project the robot should follow the white line to reach the finish position.
* The robot should move according to the bends.
* The robot must not collide with the laid-out objects.

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 “findWhiteError” function in the code calculates the error of the path or how centered the white line is in the image.
* 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”.

Table for test cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Speed | Turn | Turn | Turn | Turn |
| Right– 5.0  Left– 5.0 | 1.0 - follows white line, does not reach finish line, inefficient as robot moves very slow. | 5.0 – does not strictly follow white line, does not reach finish line, inefficient as robot moves very slow. | 10.0 - does not strictly follow white line, does not reach finish line, inefficient as robot moves very slow. | 15.0 – follows white line imperfectly, does not reach finish line, inefficient as robot moves very slow. |
| Right– 10.0  Left– 10.0 | 1.0 – follows white line, reaches finish line, but time taken is still high. | 5.0 - does not strictly follow white line, reaches finish line, but time taken is still high. | 10.0 - does not strictly follow white line, reaches finish line, but time taken is still high. | 15.0 - follows white line imperfectly, reaches finish line, but time taken is still high. |
| Right– 15.0  Left-15.0 | 1.0 - follows white line, reaches finish line, time taken is moderate. | 5.0 - does not strictly follow white line, reaches finish line, time taken is moderate. | 10.0 - does not strictly follow white line, reaches finish line, time taken is moderate. | 15 - follows white line imperfectly, reaches finish line, time taken is moderate. |
| Right– 20.0  Left– 20.0 | 1.0 - follows white line, reaches finish line, time taken is adequate. | 5.0 -does not strictly follow white line, reaches finish line, time taken is adequate. | 10.0 - does not strictly follow white line, reaches finish line, time taken is adequate. | 15.0 - follows white line imperfectly, reaches finish line, time taken is adequate. |
| Right– 30.0  Left– 30.0 | 1.0 – does not follow white line, reaches finish line but takes incorrect path. | 5.0 - does not follow white line, reaches finish line but takes incorrect path. | 10.0 - does not follow white line, reaches finish line but takes incorrect path. | 15.0 - does not follow white line, reaches finish line but takes incorrect path. |
| Right– 40.0  Left– 40.0 | 1.0 – does not follow white line, does not reach finish line | 5.0 - does not follow white line, does not reach finish line | 10.0 - does not follow white line, does not reach finish line | 15.0 - does not follow white line, does not reach finish line |
| Right– 50.0  Left– 50.0 | 1.0 – does not follow white line, does not reach finish line | 5.0 - does not follow white line, does not reach finish line | 10.0 - does not follow white line, does not reach finish line | 15.0 - does not follow white line, does not reach finish line |

Using 20.0 for “vRight”, 20.0 for “vLeft ” and 1.0 for “turn” value lets the robot follow white line, the robot reaches finish line and adequate amount of time is taken.