**Intelligent Mobile Robotics – FEEG6043**

**Assignment 1: Planning and Control**

*Group 1 – Skye* McKechnie*, Connor Costa, Danae Topi, Marwan Ismail*

A white rectangular box with black text

AI-generated content may be incorrect.Task 1

Table

*The final values for the control parameters and wheel set up are as follows within Table 1:*

*To find the optimized values we swept through a range of control parameters (0.5-2.0 for) Tau and (0.1-0.6 for) L whilst keeping the wheel geometry constant. This allowed us to measure the radial error of the robots estimated path with reference to its actual location. Tables 3 and 4 outline the averages that where taken within the sweeps to determine the best values.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **T 0.5** | **T 0.75** | **T 1** | **T 1.25** | **T 1.5** | **T 1.75** | **T 2** |
| **AVERAGE** | 0.0472 | 0.0446 | 0.0446 | 0.0504 | 0.0504 | 0.0504 | 0.0504 |
| **MAX** | 0.089 | 0.080 | 0.080 | 0.092 | 0.092 | 0.092 | 0.092 |
|  |  |  |  |  |  |  |  |

A number on a white background

AI-generated content may be incorrect.

Table & 3 outlining the averages taken for the sweep.

A screenshot of a graph

AI-generated content may be incorrect.

Figure : Plot of robot parameters using control variables listed in table 1, and for waypoints at [(0,0), (1.4,0),(1.4,1.4),(0,1.4), (0,0)]

*Due to the high unbounded error of our motion model, the actual robot conforms quite poorly to the reference trajectory, although the control algorithm seems to do quite well with the estimated position. The motion model seems to be overturning compared to the real model.*

*Bellow is a summary of the ground truth position errors with respect to the reference trajectory, whilst sweeping through Taq and L.*

Figure 2 & 2: showing group truth positions errors with respect to reference trajectories

*The overall code can be split up into four sections firstly Kinematics and Motion Modelling, secondly trajectory generation, thirdly Feedback Control, and Sensor processing (using Lidar and wheel encoder data).*

*A math equations with numbers

AI-generated content may be incorrect.The motion model work by covering the wheel rates into robot twist initially using the function forward kinematics. This uses the matrix provided bellow where v and w are the linear velocity and angular rotation respectively.*

*A math equations on a white background

Description automatically generatedThese values and a small time difference value are then used to transfer the pose of the robot from the current timestamp to the next using the function rigid body kinematics which uses the equations provided bellow:*

*The main factors that effect the how the pose of the robot changes from time stamp to time stamp are the linear velocity of the robot, its angular velocity about its radius of rotation and the time taken between measurements. In light of this, effecting the wheel diameter will have no effect on the radius of rotation but only on the linear velocity of the robot and hence rotational velocity about the same radius of rotation. Changing the wheel spacing will however reduce the overall rotational speed by increasing the radius of rotation. To optimize linear error the wheel diameter should be effected and to change radial error when turning the wheel diameter should be altered.*

*The photos bellow outline how the variations in the wheel diameter and separation effect the performance of the robot. The green line outlines its true trajectory with the blue and grey being its trajectory.*

*A screen shot of a computer game

AI-generated content may be incorrect.A computer graphics on a black grid

AI-generated content may be incorrect.A computer screen shot of a graph

AI-generated content may be incorrect.*

*A screen shot of a computer game

AI-generated content may be incorrect.A screen shot of a graph

AI-generated content may be incorrect.*

Figure 9 - d= 0.162 , W = 0.015 as shown above, to robot thought it was traveling much slower than it actual was and hence had a large overshoot and got stuck in the side of the wall.

Figure 3- d = 0.03, W = 0.074, as shown above the robot thinks its wheel span is much smaller than its true value and thinks that it is rotating faster than it is and therefore undershoots its curve

Figure 6- d = 0.162, W = 0.074, the optimum values found with the lease amount of error.

Figure 9 – d = 0.3, W = 0.074 as shown above the robot thinks its wheel span is larger and hence it is rotating slower than it actually is and over shoots its turn.

Figure 3 - d = 0.162, W = 0.15 the robot thought that it was traveling much fast that its true value and hence a very large undershoot.