Name: Solomon Ayansola Student number: M00881460

DESIGN OF A CONTROLLER FOR SIX MOBILE ROBOTS

Introduction

The objective of this report is to the design of a controller for six mobile robots and simulate their trajectory using MATLAB. The code is developed in MATLAB and uses an ODE45 solver to simulate the motion of the robots over time. The code has been developed in four stages, starting from CODE1 to CODE5, with each code building on the previous one. The final code is CODE5, which represents the complete solution to the problem of controlling the motion of six robots.

Kinematic Control Formula

In order to control the motion of a mobile robot, a kinematic control formula is used. The formula relates the velocity and angular velocity of the robot to the position and orientation of the robot with respect to its desired goal. The formula used in this code is given by:

v = kp * (pt - p);

w = ka * (at - a) + kb * b;

Where v is the linear velocity of the robot, w is the angular velocity of the robot, kp is the proportional gain for the position control, ka is the proportional gain for the orientation control, kb is the gain for the orientation control, pt is the target position of the robot, p is the current position of the robot, at is the target orientation of the robot, and a is the current orientation of the robot. The term kb * b in the formula represents the control input for the orientation control, where b is the angular velocity of the robot.

Controller Parameters

The control parameters used in the code are:
kp = 3;
ka = 8;
kb = -1.5;
K = (kp, ka, kb) = (3, 8, 1.5)
These parameters have been selected based on experimentation and trial-and-error. They represent a set of values that can drive the robots to the goal by following a smooth trajectory. The value of kp determines how fast the robots will move towards the target position, while the values of ka and kb determine how fast the robots will orient themselves towards the target orientation. The value of kb has been chosen to be negative because it represents the damping of the angular velocity of the robots.

Development and Explanation of the Codes

The code has been developed in four stages, starting from CODE1 to CODE5.

CODE1: Code1 is written for a single robot, and it calculates the motion of the robot over time, starting from an initial position and orientation and moving towards a target position and orientation. The initial conditions are defined as p0, a0, and b0.

The target conditions are defined as pt, at, and bt.

The time interval for the motion is defined by T and dt, with t being the time vector for the motion.

The control parameters for the motion are defined as kp, ka, and kb.

The ODE solution is computed using the ode45 function from MATLAB, which is a numerical integration method that solves ordinary differential equations (ODEs). The input to the ode45 function is the robot_dynamics function, which defines the robot's motion. The function robot_dynamics takes two inputs: t and x, where t is the time vector and x is a vector that contains the states of the robot.

The states of the robot are defined as p, a, and b. p is the position of the robot, a is its velocity, and b is a control input.

The velocity and control input are used to compute the derivative of the states using the equations

v = kp * (pt - p) and

w = ka * (at - a) + kb * b.

The final output of the function robot_dynamics is the derivative of the states, which is defined as xdot = [v; w; 0].

Finally, the results of the motion are plotted using the plot function from MATLAB, which plots the position p against the velocity a. The x-axis of the plot is labeled as "p" and the y-axis is labeled as "a". The plot is shown in fig1 below

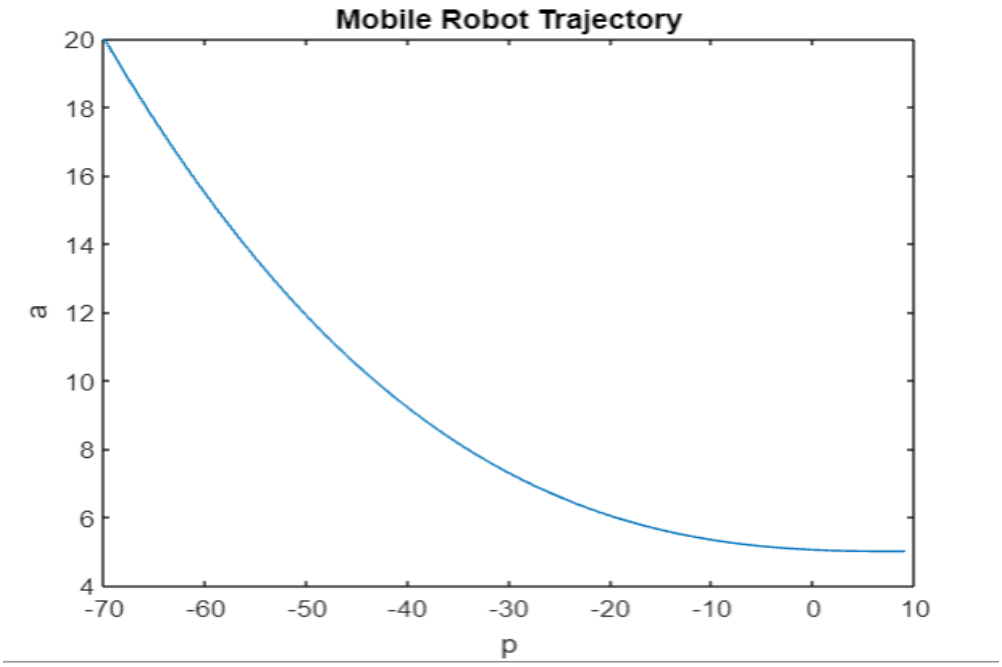

Fig1: Trajectory of the robot generated by code1

CODE2: Code2 is extended to six robots, and the same calculations are performed for each robot. The simulation results are plotted in fig2, showing the relationships between p and a for all 6 robots. A for loop is used to plot each of the 6 robots' trajectories, and a legend is added to distinguish between them.

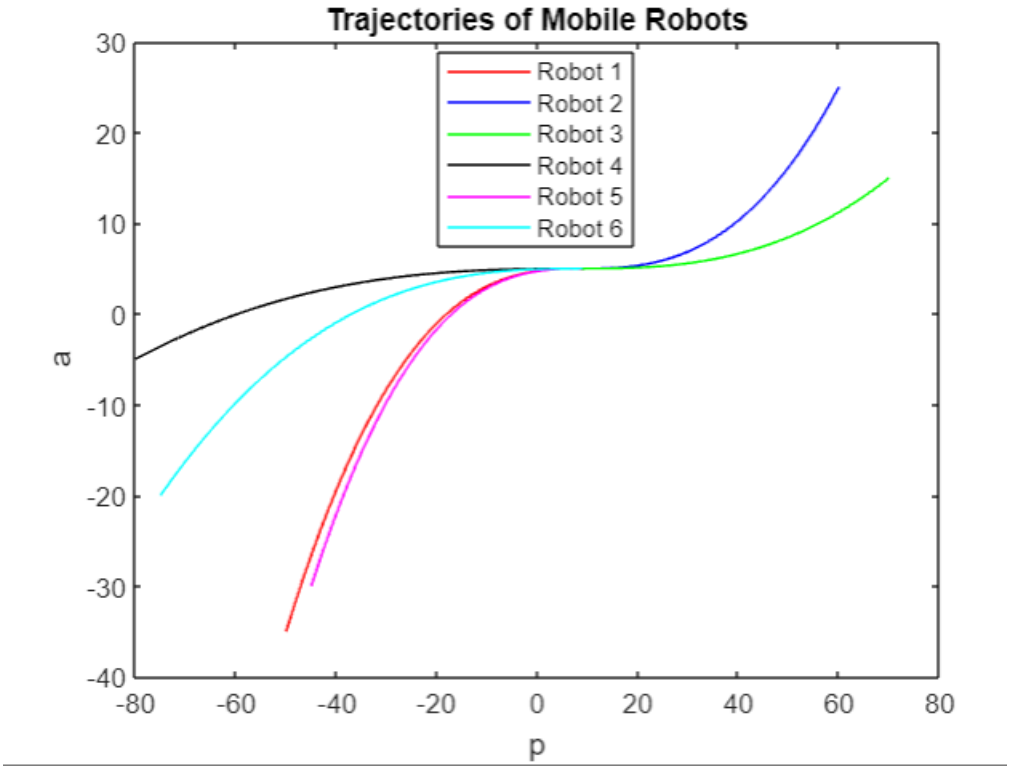

Fig2: Trajectory of the mobile robots generated by code2

CODE3: Code3 is modified to plot the motion of each robot individually, which allows for a better visualization of the motion of each robot.

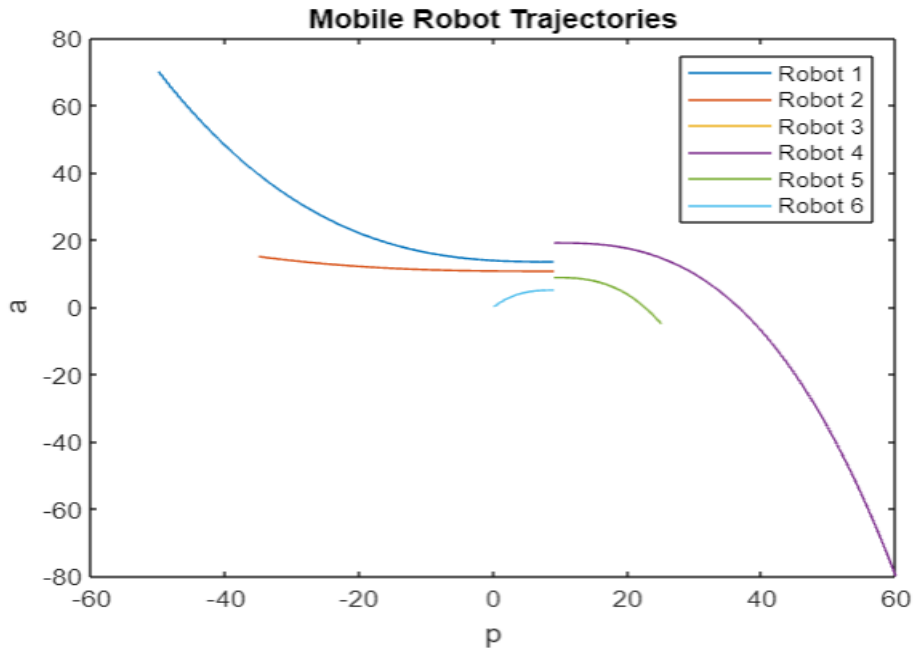

Fig3: Trajectory of the mobile robots generated by code3

CODE4: the final code, the code is written in a more structured and readable manner, making it easier to understand and modify.

The second and final target conditions are defined as pt2, at2, and bt2 such that

p0 = [-50; 60; 70; -80; -45; -75]; a0 = [-35; 25; 15; -5; -30; -20];

If the current time t is greater than T, which is the end of the first goal's interval.

CODE5: Code5, the final code is similar to code4 but which constraint on the maximum and minimum velocity. Restricting the maximum velocity to 50 give fig5.

Conclusion

In this report, the design of a controller for six mobile robots has been analyzed. The code has been developed in five stages, starting from CODE1 to CODE5, with each code building on the previous one. The final code, CODE5, represents the complete solution to the problem of controlling the motion of six robots. The kinematic control formula has been explained, and the set of controller parameters that can drive the robots to the goal by following a smooth trajectory has been found. Finally, the development of the code from CODE1 to CODE5 has been explained.

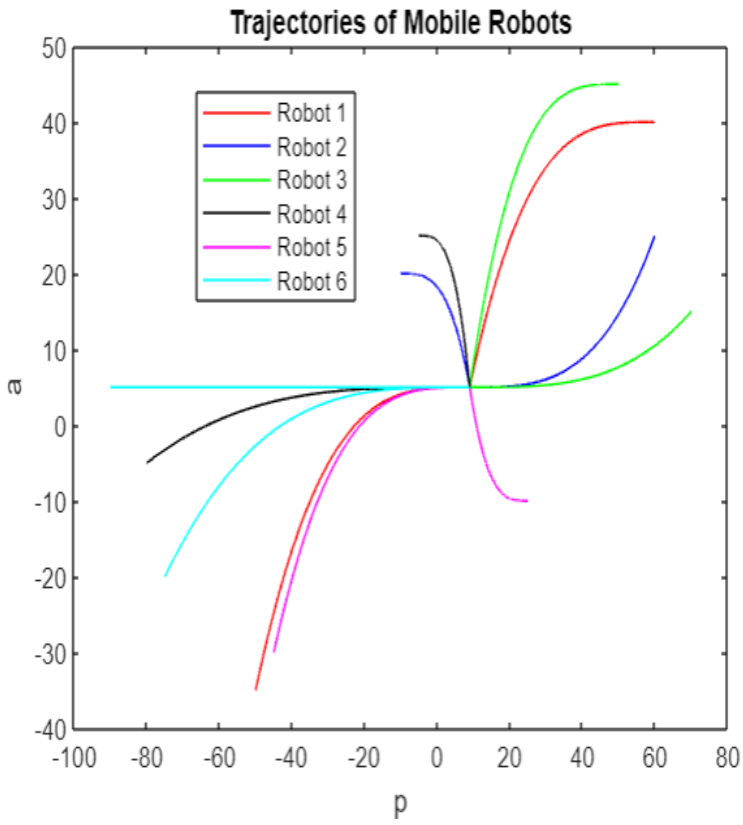

Fig4: Trajectory of the mobile robots generated by code4

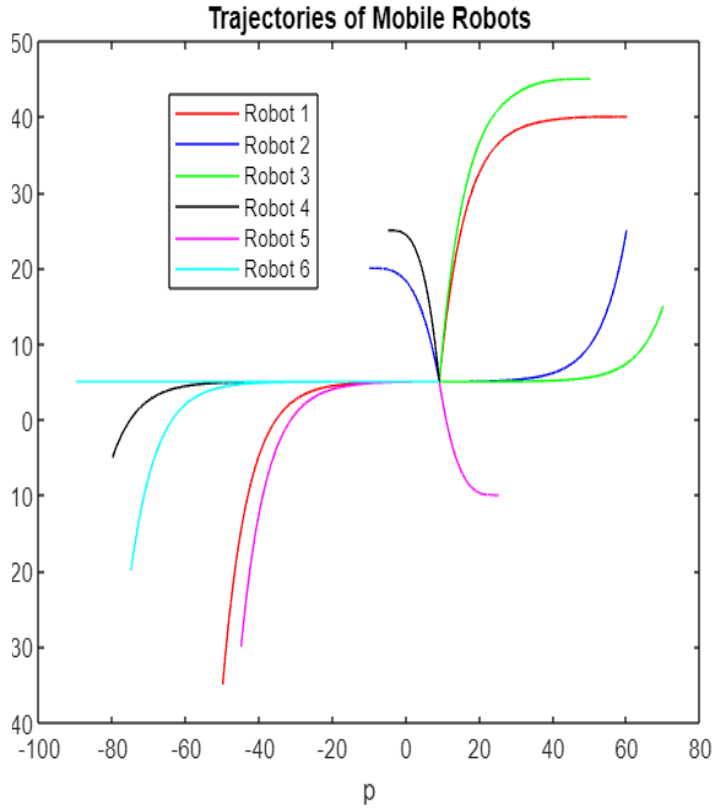

Fig5: Trajectory of the mobile robots generated by code5 with velocity constraint 50