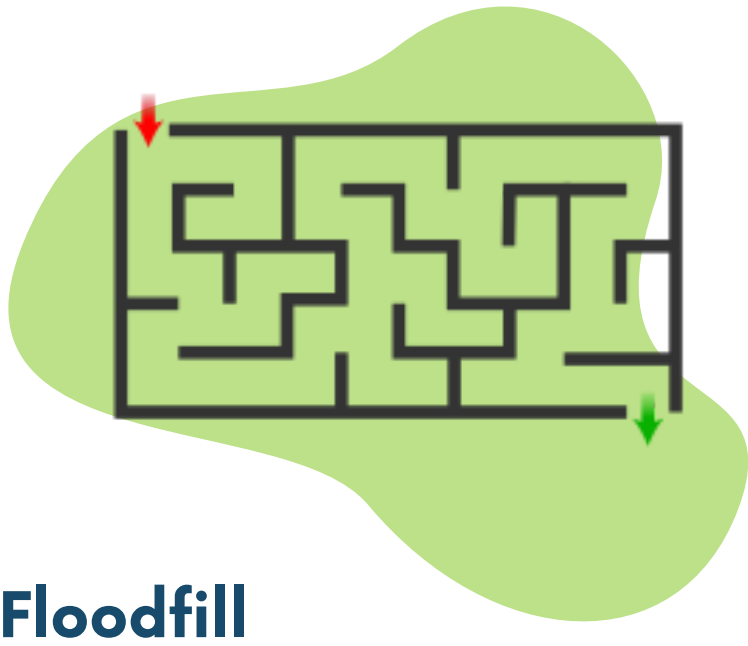
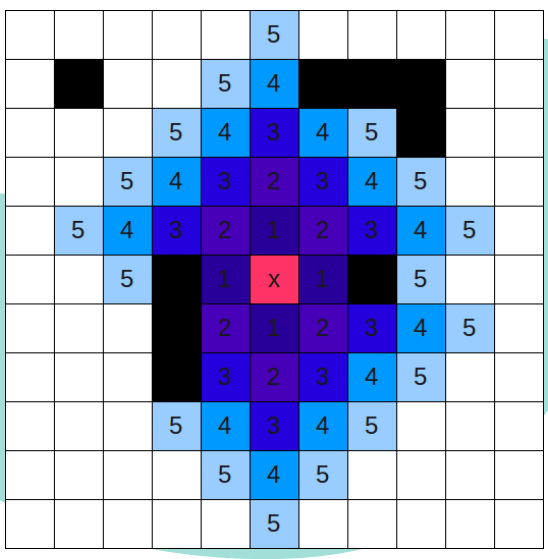
Maze solving Autonomous Mobile Robot

Operating Systems - ICC102
Luis Rodolfo Macias
Gustavo Vazquez

Maze solving

There are several maze solution algorithms: Dijkstra, A*, Tremaux, Left wall, DFS, BFS and the most popular one Floodfill



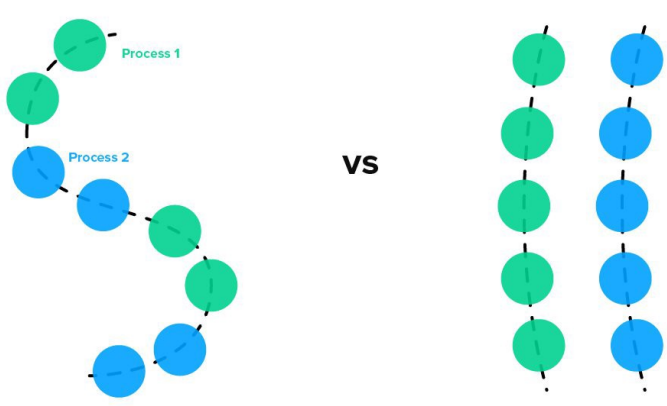


Floodfill

It's an algorithm pretty basic for computer graphics, its the representation of how the paint bucket works, but can be modified to work as a maze solving algorithm like (BFS and DFS).

Concurrency

Is the composition of independently executing processes. "Concurrency provides a way to structure a solution to solve a problem that may (but not necessarily) be parallelizable." (Wahome, 2020). (FreeRTOS was used in this implementation to perform concurrency)

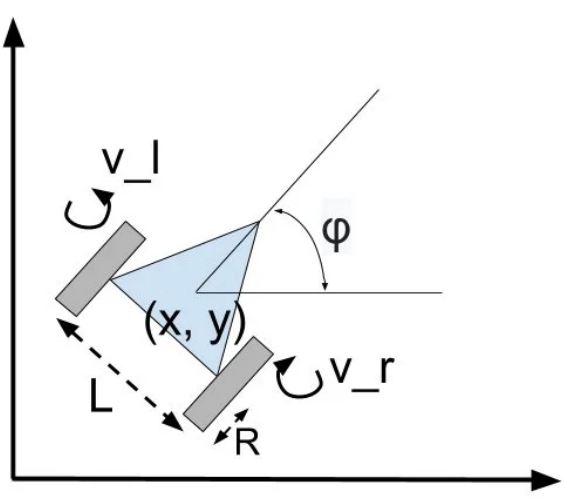


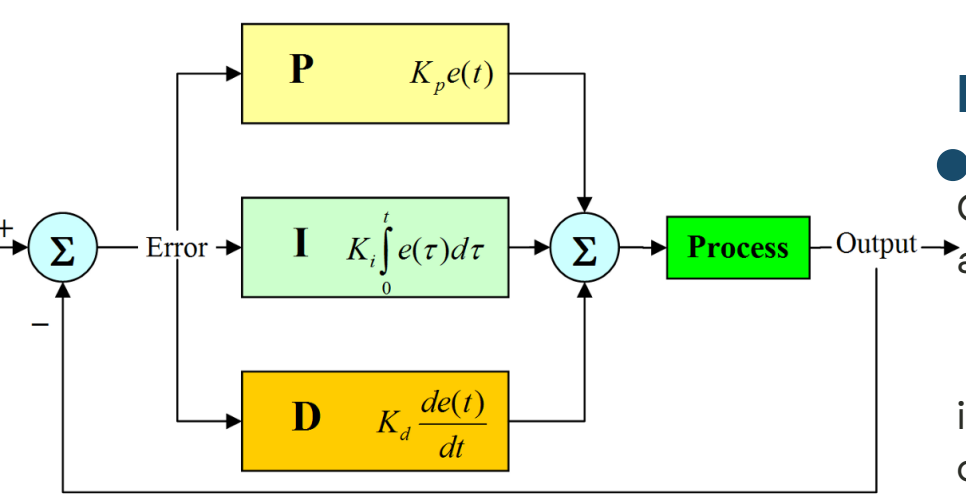
Differential drive robot

Mobile robot whose movement is based on two separately driven wheels

Odometry

Is the use of data from motion sensors to estimate change in position over time. (Encoders, IMU, etc)





PID Controller

Control for each wheel according to the position, angle and destination of the robot. In this implementation It calculates the input frequency PWM of each motor

Reference

Wahome (2020). Concurrency is not parallelism. Available at: https://kwahome.medium.com/concurrency-is-not-parallelism-a5451d1cde8d