MAZE SOLVING USING X80 ROBOT

Advisor: Dr. Ahmad Masoud
Mr. Muhannad El-Hassan

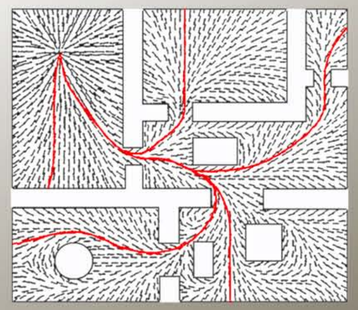
Ahmad Ballas Maher Abdulrahman Abdulrahman Al-Ghamdi Ahmad Al-Tenbakji Hamza Zeidan

Objectives

- To study the robot mobility and how to utilize the Harmonic Potential Field (HPF) to solve a maze.
- To program a MATLAB code that solves both known and unknown mazes of any size.
- To physically implement the approach on the X80 mobile robot platform.

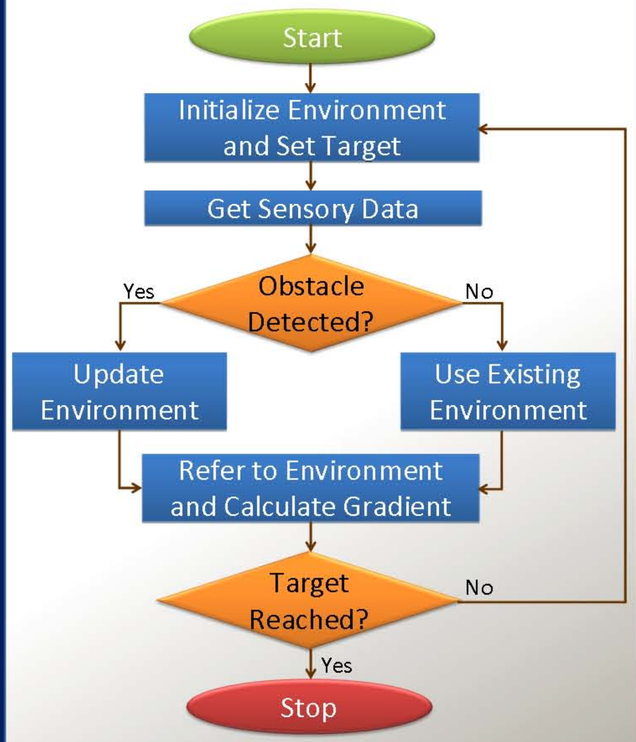
HPF Approach

- Motion planning is needed to guide the robot to the target.
- The Harmonic Potential Field planner is an effective way to convert the environment into guidance signals.
- This method mainly uses the principle of electric field flow from high to low potential.
- The environment is set as a uniform conducting sheet.
- Boundaries and obstacles are set at high potential, while the target is set at zero potential.
- The potential gradient vectors create continuous paths (electric field lines) from any point to the target.
- The robot follows one path depending on its initial position.

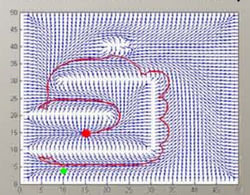


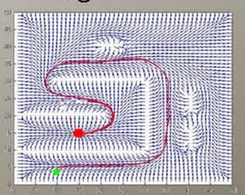
Unknown Environment

- The environment is initialized as if there are only the boundaries and the target, and the robot has to discover it through sensing at each step it takes.
- If an obstacle is detected, the environment is updated and the new path is regenerated.



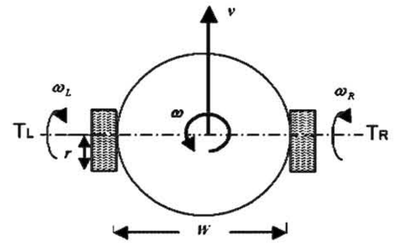
- By storing the environment at each trial, the robot improves its trajectory until the optimum path is found.
- MATLAB is used to simulate the HPF planner.
- Note: Green = Start, Red = Target.


Trial #1


Trial #2

Navigation Control

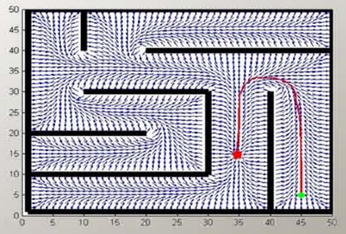
- Now the robot knows its path, but cannot go along it without actuation.
- So, the guidance signal obtained from HPF is translated into speeds of wheels.

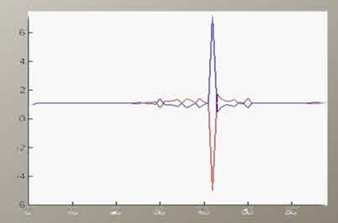


$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{W} & \frac{-r}{W} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

- Using the equations above, the robot can be modeled in MATLAB to simulate the navigation control process.
- An example of navigation can be seen below. Then, the speed signals sent to the robot wheels are shown.


Green= Start, Red= Target


Blue= Right wheel, Red =Left wheel

Experimental Results

C# programming language is used to implement the maze solving algorithm and to communicate with the robot. Real examples of the maze solving process are shown below.