School of Electrical, Computer & Telecommunications Engineering
University of Wollongong
Intelligent Control – ECTE441/841/941
Practical Project II

Introduction

Continuing on from the first project, this second project also provides you with an opportunity to apply the methodologies you are taught in the lectures and hence to develop a better understanding of them. The project is designed based on Matlab Fuzzy Toolbox and Simulink.

Assessment

Your project work will be assessed based on the report you submit. Your report should include:

- The procedures and methodologies applied
- The results produced including the diagrams and critical analysis of them
- A soft copy of your Project Report should be submitted online via the "Submission of Project Report 2" on the Moodle site. A soft copy of your simulation model is also required for validation and assessment of you developed programs and this needs to be firstly incorporated into a single folder of the several files including Matlab and Simulink fuzzy controller files. This folder should be then compressed as a single **ZIP** file (**DONOT use RAR**) and submitted online via the "Submission of Software for Project 2 Zip file" on the Moodle site.

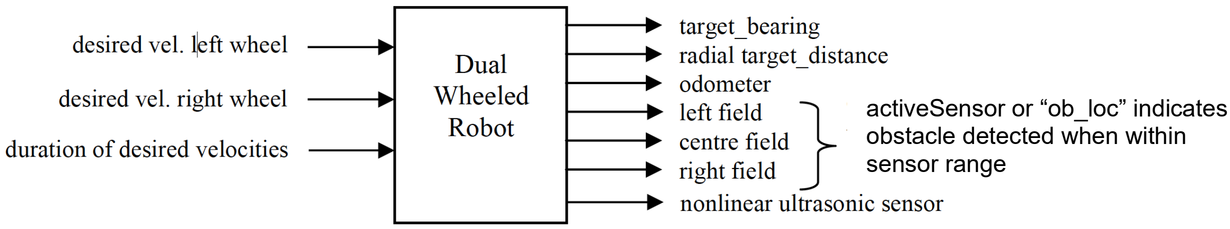
In spite of commonality of results, your report should not resemble any other report. In case of identical reports, your mark will be the mark of one report divided by the number of identical reports.

PLANT and OBJECTIVE

The plant in this project is a mobile robot, which is required to move in a two-dimensional space from a set starting location *(X = 3.4, Y = 5.8)* on the 14-obstacle arena provided (*14OA-obstacle_data.p*), until it reaches the perceived (or only estimated) target goal location. In doing so it must also avoid certain unsafe obstacles areas (irregular polygons) that are suspected mine fields. If the robot enters one of these areas it may detonate a mine (Game Over). A simple 5-obstacle arena (*5OA-obstacle_data.p*) is also provided to aid the development of your controller – simply copy and rename either arena file to become: *obstacle_data.p*

Initially the target location is only approximately estimated, and as the robot moves closer towards this, clearer signals will be received by the robot sensors, with higher signal to noise ratio. With these stronger signals, more accurate determinations of the true target location become possible – hence the final (or actual) target location may be different compared to its initial estimated location. Although path planning is one approach to addressing this problem, you are required to develop an Intelligent Controller to achieve this without pre-planning – that is you should **NOT** utilise any coordinate positions in your submitted robot controller otherwise marks will deducted.

Matlab code for the kinematics of the robot has been already written in order to simulate its dynamic response in a range of contexts. Importantly, the motion and precise direction, or pose of the robot, is achieved through independent control of both left and right wheels. The robot model also includes a number of sensor mechanisms, which your designed controller can utilise, to guide progress towards the goal, as well as avoid any nearby obstacles. This has subsequently been incorporated within the Simulink environment, in order to facilitate the development of various intelligent control solutions.
Thus the robot model in this case represents a multi-input multi-output, or MIMO system.



TASK 1 – FUZZY CONTROL SIMULATION

Apart from developing a general Fuzzy controller to drive the robot to its destination, you will also need to capture data of your controller's behaviour and responses. Several sections of data may need to be captured, and saved as training data for the alternative approaches of Task-2, such as custom developed neural network controllers, or alternatively ANFIS optimisations.

You should provide plots of the complete trajectory of the robot as it travels to the target, as well as all input and output signals (on the same graph if possible), as well as the rule surface in each case. You should also report the odometer value, the total distance travelled by your robot in reaching its target.

Learning Objectives

On successful completion on this part you should be able to carry out
- Complex simulation and modelling of control systems using Simulink
- Evaluate control system performance

Concepts
- Control system simulation
- Modelling & Building of Fuzzy Inference System (Mamdani or TSK)
- Nonlinear systems & process dynamics

TASK 2 – FUZZY ANFIS CONTROL

In this task you are required to develop Neural Controller, OR alternately an optimised Adaptive Neural Fuzzy Inference System (ANFIS) controller for the same robot in Task-1. The second alternative will require you to convert your initial fuzzy controller of Task-1 to an ANFIS equivalent.

This will require you to compare all membership functions in the initial ANFIS system with those of the former (Mamdani or TSK). Using the training data of Task-1 (and maybe now Task-2) you should adapt and optimise the various membership functions using the "ANFISEDIT" tool. Once again compare the MF's, and once again, provide plots of all input and output signals (on the same graph if possible) as well as the rule surface in each case.

Learning Objectives

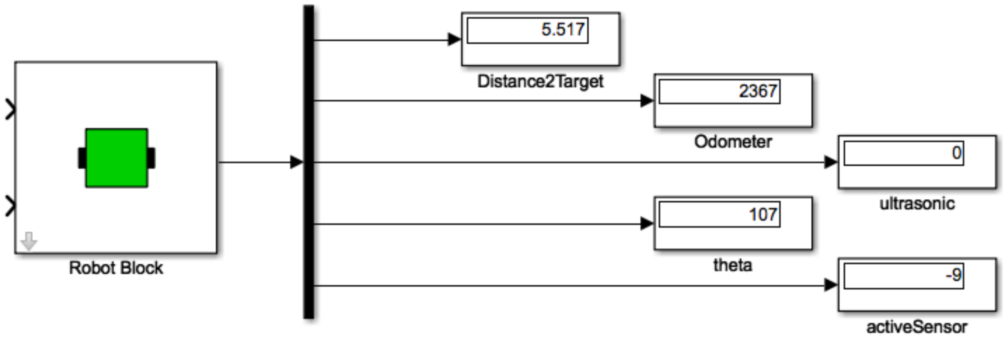
On successful completion on this part you should be able to implement and optimise a controller developed as an Adaptive Neural Fuzzy Information System (ANFIS).

Concepts
- ANFIS fuzzy system control
- Intelligent controller optimisation

ROBOT TECHNICAL DATA:

Robot: two parallel wheels drive the robot; the speed of each wheel can be set separately, this is given as a Simulink model file, "***RoboBlock.slx***". Apart from this there are two associated but compiled files, which are required for the simulation to work. If the robot collides with an obstacle, OR, consumes all of its available energy resource, the simulation terminates, this may often happen as you develop your fuzzy control system(s). The complete suite of files are contained in a zipped folder "project-2.zip" available on the Moodle site. Without a designed controller the simulated robot will move randomly and most likely enter a minefield.

The multiplexed output of the Robot Block model in Simulink, has been conveniently de-multiplexed with the following infrastructure, exposing the five sensory signals.



Target: is a circular Roundel type symbol = 

Obstacles: obstacles here are represented as polygons with varying radii;

Block inputs:

vl : velocity of left wheel

vr : velocity of right wheel

range of vl, vr: [-2, 2]

Block outputs:

dis*:* is the estimated line-of-sight or radial distance between robot and target

activeSensor (ob_loc)*: the range of obstacle detection sensor is active within 0.5 meters*

-9 = No obstacle detected

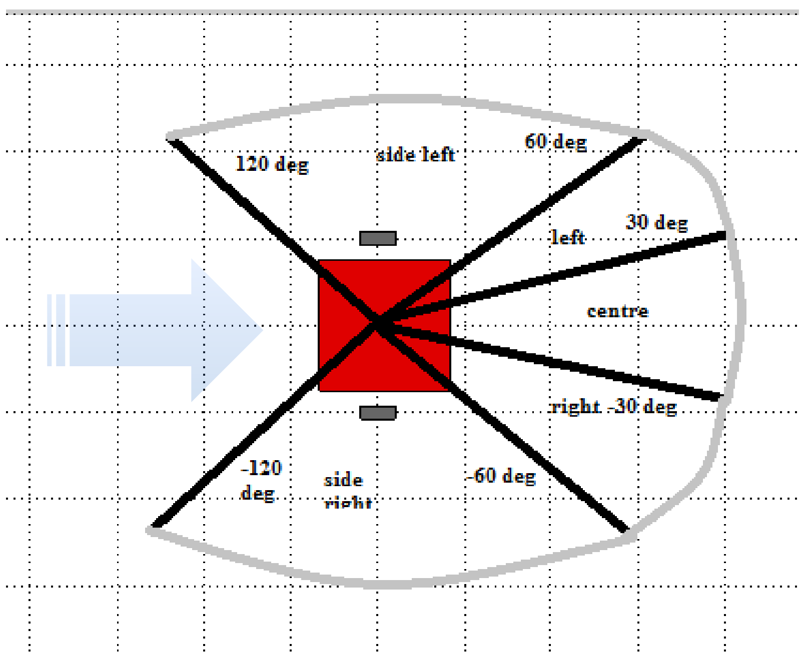
-5 = obstacle detected on the left side

-2.5 = obstacle detected on the left

0 = obstacle detected in the middle

2.5 = obstacle detected on the right

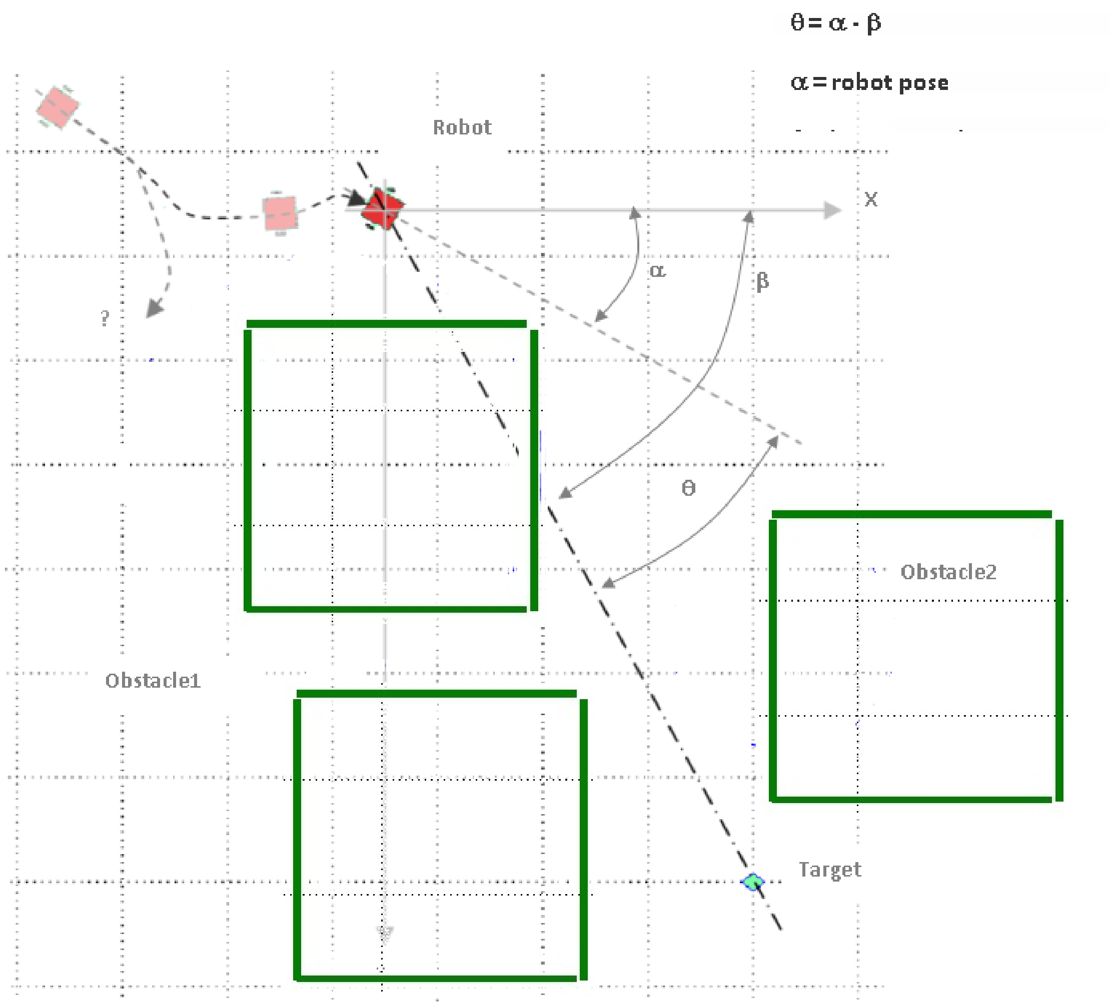
5 = obstacle detected on the right side



***ultrasonic**:* sensor output in non linear units, value increases from 0 to 25 as the distance decreases when within 0.5 meters of an obstacle that is ONLY directly in front of the robot's poised forward direction.

odometer: a record of the accumulated distance travelled by the controlled robot

***Theta**:* is given as the angle between robot pose (where it is facing) and target bearing angle.



An example simulation view explaining bearing and robot pose angle.

Note: In previous sessions the project-2 task involved various square or circular objects to avoid, however since 2015 the robot should negotiate around (unsafe) irregular polygons of unmapped land mine areas.