University of Bristol
Faculty of Engineering
Robotic Systems PG

Romi report

NISHANT RAMAKURU

Supervisor:
Paul O'Dowd

First version

Bristol, May, 2020

1 System Architecture

Briefly describe the architecture of your system for the line-following task. Identify which elements of the system have the greatest influence on the level of performance of your Romi to complete the line following task.

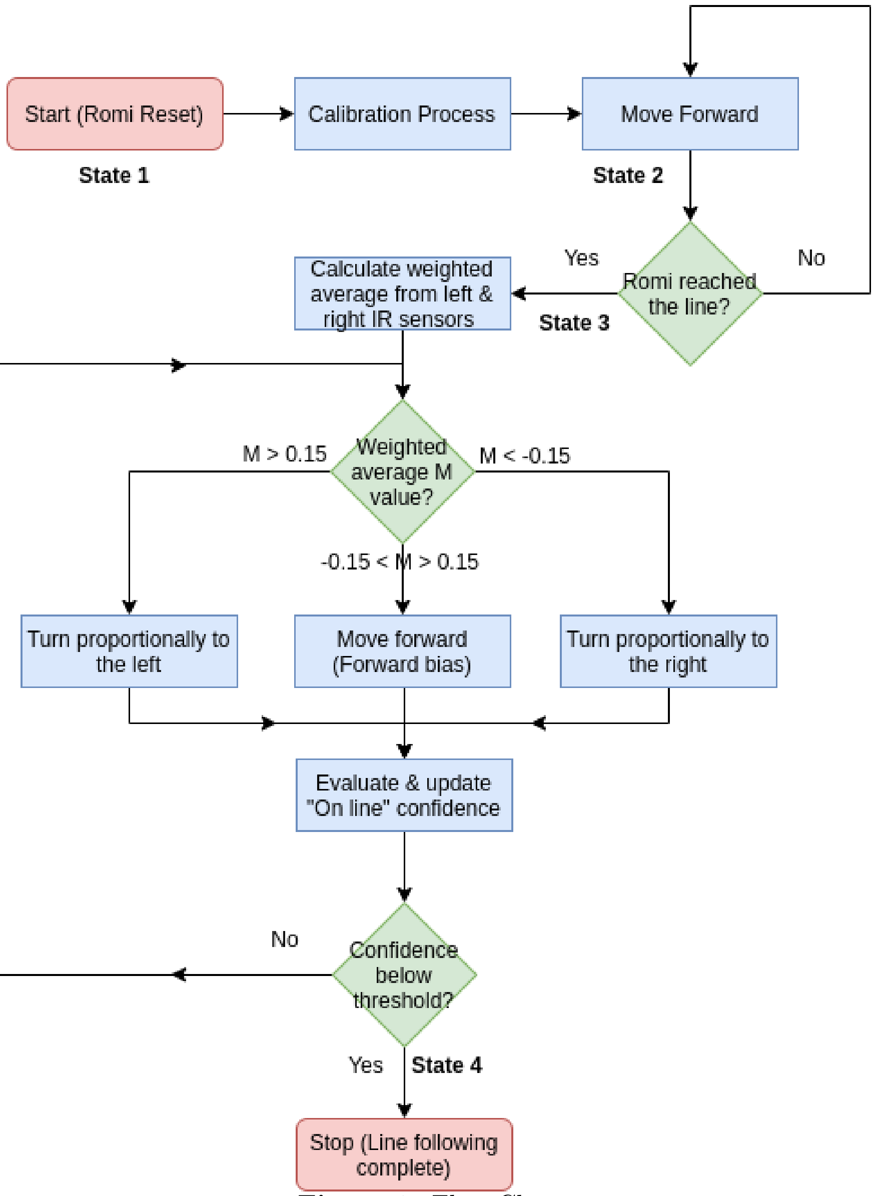

Figure 1: Flow Chart

Provide a flow chart, diagram or pseudo-code to help describe your robotic system in overview. Describe why you decomposed the line-following task into your solution the way you did.

The flow chart provided in Figure 1 acts as a visual representation of the decision procedure which was designed and then implemented in code, allowing the Romi to complete the line following task. The procedure of the system can be easily represented by the various ordered states of operation that the Romi can be in (State 1 through State 4). This is an example of a Finite State Machine (FSM) and it's implementation will be explained further towards the end of this section. On boot the Romi starts up and proceeds into State 1. This results in the Romi undergoing a reset procedure, displaying "RESET" to the user in the Serial monitor and initialises variables. After completion the Romi the begins the Calibration procedure. This is important as it allows for a baseline background colour (IR intensity value) to be obtained which can be used to subtract from the read in raw value of the sensors. The hope of this is that the white background of the track will now be re scaled such that it reads an approximate value of 0 across the whole background of the track and can act as the baseline from which calculations relating to the line can be made. Additionally, the calibration procedure was completed whilst the Romi was moving forward and was calculated by sampling the IR intensity 50 times during this period and taking the average value of the result. By completing the calibration procedure in this manner the calibration value obtained would be more representative of the white background workspace and was found to be more reliable than a single calibration value, which would be more likely be effected by changes in light and would have a greater chance of creating a systematic error which would impact the system.

From this part of the system onwards all operations were completed inside the main loop() of the Arduino script. Because of this greater structure was needed and the FSM became more crucial. State 2 of the system is straight forward. In this state the Romi is instructed to move forward and then to evaluate whether it has yet reached the line. It does this at each iteration of the loop. If the evaluation of the line results in a positive confirmation that it has reached the line, then the Romi proceeds to State 3. If evaluation determines that the line has not yet been reached the line, the Romi proceeds to move forward the following iteration.

State 3 is the state which contains the largest number of operations and is where the operations of the main crux of the line following occurs. The role of this state is for the Romi to stay on the line, turning appropriately when required and evaluating whether the end of the line has been reached. The core of this state revolves around the use of a weighted controller. The weighted controller in this case is in effect a proportional controller, adjusting the speed at which the Romi should turn depending on the extremity of the difference between the left and right sensor readings. Further to ensure that the Romi did not get stuck constantly turning in alternating clockwise and anticlockwise directions a forward bias clause was added. The aim of this was to enable to Romi to continue in a forward motion if the relative difference between the left and right IR sensor reading was small and would only begin turning again when the difference between the two values had exceeded a particular threshold. This can be seen by can be seen illustrated in Figure 1.1 above with 3 different motion outcomes depending on the Weighted average value (M). Within Stage 3 once motion had been applied, the "On line" confidence value is updated by evaluating a conditional statement regarding the values of the left, centre and right IR sensors. If the result of the conditional statement is "on the line" then +1 is added to confidence value, if the result is "off the line" the confidence value is halved. At the end of each iteration within state 3 this is evaluated, if the confidence value is now below the threshold value then the operation mode of the Romi will now be state 4, if not then the Romi will recalculate the weighted average M value and repeat the previous steps, as can be seen in Figure 1.

Stage 4 is the completion stage. Once the Romi has reached this stage it will have reached the end of the line if the operation of the Romi has worked correctly. once it has reached this stage, the Romi stops and completes an on off series of buzzer noises to demonstrate that completion has occurred before outputting a "Task Complete" string to the user.

Which element of your system was the most successful, and why?
One really successful element of the architecture of the system was the employed FSM. Despite it's simplicity in nature and only having a relatively small number of states (only 4 states), the FSM allowed the main (iterating) loop of the software to be organised more effectively, increasing the readability of the system being implemented. This seems like a trivial and insignificant change, however it was found that implementing a logical system of this type, whereby the Romi was in a fixed discrete state at any one time, resulted in far fewer mistakes being made than before this system architecture was implemented. This is because as the level of complexity of the program developed as the number of conditional decision statements being made in the iterating loop function of the program increased and became continually less manageable as greater functionality was added to the system. An additional element of the system which always worked very reliably was the confidence value method in order to determine whether the end of the line had been reached. The method allowed end of the line to be determined in a very short time after the end of the line was reached and yet never caused the Romi to stop incorrectly at the midpoint of the line following. was chosen to be evaluated with these specific criteria such that the rate at which the romi would converge upon the decision that it was off the line would be much faster than if the rate at which the confidence value would increase for not being on the line.

Which element received the most development time, and why?
Despite not being explicitly mentioned in Figure 1 above an unduly amount of time was taken tuning a previous version of the line following algorithm; a Bang Bang controller. In hindsight mistakes were made on investing so much time into this area in the early stages of this project. The Bang Bang controller (or on-off controller) worked by obtaining values for each of the IR sensors and then manually deciding how to threshold these values such that specific conditional statements could be executed which would result in the Romi moving in the desired direction at each timestep. After early success with manually experimenting with these values the Romi was successfully able to complete the initial stages of the line following task, however as the turns became sharper as the track progressed, the Bang Bang controller solution implemented struggled with being able to turn effectively; Often missing a turn completely and being unable to regain its position on the line. Instead of then quickly realising the limitations of the controller that had been implemented, time was spent pursing the Bang Bang controller further, trying to refine the sensor values corresponding to specific motor conditional statements, in order to complete the line following section; This was to no avail. Further once a substantial amount of time had been spent on this solution, the researcher was even less inclined to "give up" and purse a different solution. Eventually the researcher was able to effectively implement a weighted controller, in effect acting as a proportional controller, which, through the use of on the spot rotation and forward bias, was able to effectively navigate to the end of the line. The researcher has learnt from this experience and in the future will try to analyse various solutions before focusing in on a specific one.

What working practices would you recommend to others engineering a robotic solution for the same task, and why?
A working practice the researcher would recommend to others completing this engineering tasks would be to give consideration to the overall structure of the code of the system before embarking on the project. Consideration should be given to the problem as whole, including evaluation of the task, software, hardware and the environment, however within the software specifically thought should be given to the design of the underlying structure of the code and what specific methods/systems are going to be used to underpin this structure. Allowing the code to be effortlessly scaled up as the task progresses. This was found to be a mistake made by the researcher who only implemented a finite state machine after the mid point of work completed on this project. Further the researcher had not considered the structure of how a software solution could be implemented and instead continually added greater functionality as the task progressed without giving too much thought too the overall structure. The result of this was that logical mistakes were made as the code became continually more difficult to follow and a greater time was taken to then transfer all the code to acting in the FSM structure. In the future it would be advisable to implement a FSM or other similar system from the outset and use this then as a structural platform from which the rest of the functionality of the Romi can be written.

Break down of task when troubleshooting, helps understanding and ability to find errors [extra point if space]

Notes: Remove Evaluate from on line confidence?

2 Q2

Describe in detail a specific challenge or success relating to a sensor, motor control or a sub-system (e.g. behaviour) for your Romi robot.

[You are encouraged to use small extracts of your source code where it is relevant, e.g. to communicate a problem, your solution, or how you captured debugging information.]
Choose an element of technology (hardware or algorithms) which was interesting or challenging for you. What were its advantages and disadvantages?

An element of technology which was challenging to the researcher was the PID controller **Why did you use the solution you did? What else did you consider?**

3 Further Investigation:

Identify your chosen problem in terms of the key elements of a robotic system: task, software, hardware and environment.

The problem selected to be considered in this section of the task was that of the Romi navigating around a, previously unseen, maze. Moving from the start point to the end point. An illustration of this task can be seen in Figure 2 below. This problem takes advantage of proximity sensors being readily available and easily connectable to the Romi, and is the main hardware addition from the Romi components used in the previous line following task. Real world applications of a adaptable navigation achieved on successful completion of this task could be applied to autonomous hoovers or lawnmowers, which would be required to sense and take action against objects in the way of it's path.

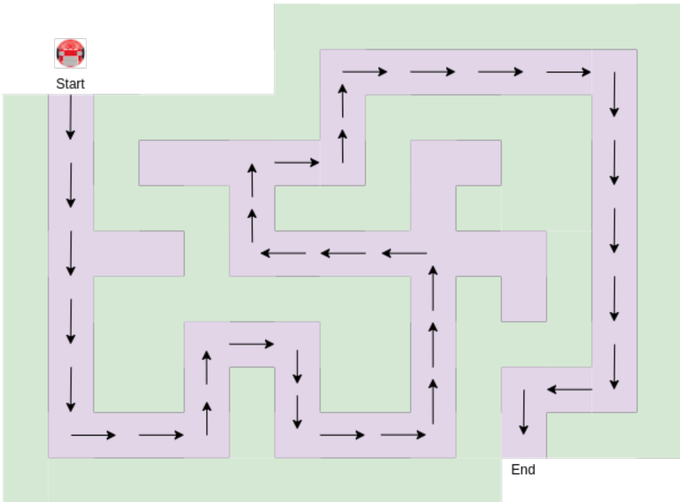

Figure 2: Maze Task

Task	To move from the start point to the end point of the maze, in as small of a duration as is possible, without colliding into the maze walls.
Software	FSM States - State 1: Calibration. State 2: Line following algorithm. State 3: Maze End Line following algorithm: Move forward(), if proximity above certain value = stop, rotate 90 degrees and evaluate distance, once clear path determined, move forward. Kinematics calculated and evaluated such that direction preference is one which is most likely to move it away from the start point. I.e. path choice is based off probabilistic robotics and is not random.
Hardware	Proximity Sensor (i.e. 3V Pololu Carrier w/ Sharp GP2Y0A60SZ IR Range Sensor - 10 to 150cm); To allow for the distance in front of the Romi to be calculated Romi Chassis Kit; Standard Romi chassis, motors and wheels used in the previous line following task.
Environment	Unknown maps configuration. Constant wall height and track thickness. Performance tested on multiple mazes.

[maybe extra above^^] [Romi will rotate on spot]

Discuss how you would design a study and how you would evaluate performance of a solution.
A study will be devised, as per Figure 2 above, which will investigate the Romi's ability to successfully navigate around a previously unseen map. The performance will be evaluated by using a series of 5 mazes, with increasing complexity of turns and patterns making maze 5, in theory, the the most challenging and maze 1 the most basic. The main performance metric of this task will be the number

of unseen mazes that the Romi can successfully complete. With completion of the majority of mazes (4 out of 5 or higher), each within a reasonable time frame, implying that the Romi would be able to generalise well to a significant number of unseen maps containing similar features. And so completion of 4 mazes or more will be the performance benchmark of this study. Additional metrics could include evaluating the average maze completion time and the average number of accidental collisions with walls made. Due to the potential performance variance of completing each maze the experiment will be repeated multiple times and an average taken for each maze, increasing reliability and validity of the results as a whole. Various configurations of mazes will be used, however the overall test set of mazes, used to evaluate the performance of the Romi, will be kept the same. This will be done so that a fair test can be maintained. If this task was wanted to be extended further the same Romi could be used with different path navigating algorithms uploaded to it. Performance of these various navigation algorithms could then be determined and compared. From experience, using a different Romi for testing, despite being of the same manufacturer specifications, does not mean that the different Romi's will have the exact same performance characteristics. Because of this the same Romi would be used in this extension task such that the only independent variable of the experiment would be the navigation algorithm.

Even though the Romi is a relatively simple robot, you have experienced some of the hidden complexity in using it to solve the line-following task. Discuss how your experience with the Romi influences your approach to your chosen robotics challenge.

From experience with the Romi it is clear that the theoretical world does not fully align with the real world in the way you would expect. With often unexpected behaviour arising from factors in the real world which would not have been considered from a theoretical standpoint. As a result of this previous practical experience the researcher would now ensure that practical testing was completed in the early stages of the development procedure, with the hope that more factors of the robots environment, that will affect its performance when completing the task, can be determined. In this experiment that would be: Test calibration, test on various corner types, test FSM, test proximity values (where on the map do unexpected readings occur) From these initial tests problems in the environment can be identified and negated as much as possible in the development of the overall solution. [Additional point]