Robotics

Final Project

Pietro Rodlfo Masera, Tommaso Verzegnassi

Index

I) Introduction:

The project goal is to explore the ground sensor functionality of the Thymio and see what can be done with it in conjunction with odometry. The Coppelia Scene were it takes place is the following:

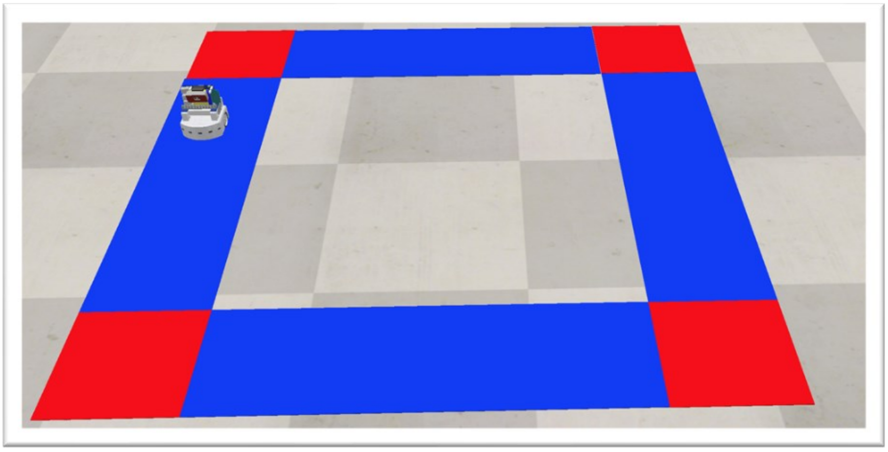


Figure 1

The challenge we set to ourselves is the following: is it possible to make the Thymio consistently follow the shape of the map staying approximately in the middle of the paths by just using odometry and ground sensors? The answer is, in short, yes. It seems trivial at first, but there are some issues that will later be discussed. The Thymio can follow the path perpetually, more than 90% of the laps meet our objective, that is being centred in the path. In the rare instances where an error occurs the Thymio is able to adjust itself automatically regaining the right position and continue doing correct laps around the map.

We are going to discuss the relevant challenges of this problem and what was our approach to solve them. After we explained our solution, we will try to implement stress and disturbances in the system to see its strong points and weaknesses.

II) Recognizing colour:

In Ros the color recognizing process is handled as an Event to which we subscribe. The colours used are mainly Red and Blue. For one of the stress tests green has been used. Another colour is represented by the Coppelia floor itself, which is neutral (basically behaves like pure white would). The ground sensors of the thymio work in the following way: the light reflected by the ground where the thymio is placed is transformed as an integer value between 0 and 1023. 0 indicates black and 1023 indicates white. It is important to know that there are 2 ground sensors, one on the right side and one on the left side. The distance between them is very small but crucial as we will discuss later.

Here You can find a table of the colours with the relative integer value:

Blue = 862

Red = 730

Green = 672

Floor = 1023

The simplest operation is made when both of the thymio's ground sensors sense 862. In The blue area the robot just moves Forward with 0 angular velocity.

III) Turning by 90 degrees:

The red squares of our map are the turning point and, ideally, the robot should turn by 90 degrees. I said ideally since the robot can make errors, especially if the angular velocity is high. Doing this was relatively easy using ground sensors in conjunction with odometry and Deepcopies, which are copies of the robot's odometry that we created in specific moments. The following images can be very useful to the explanation:

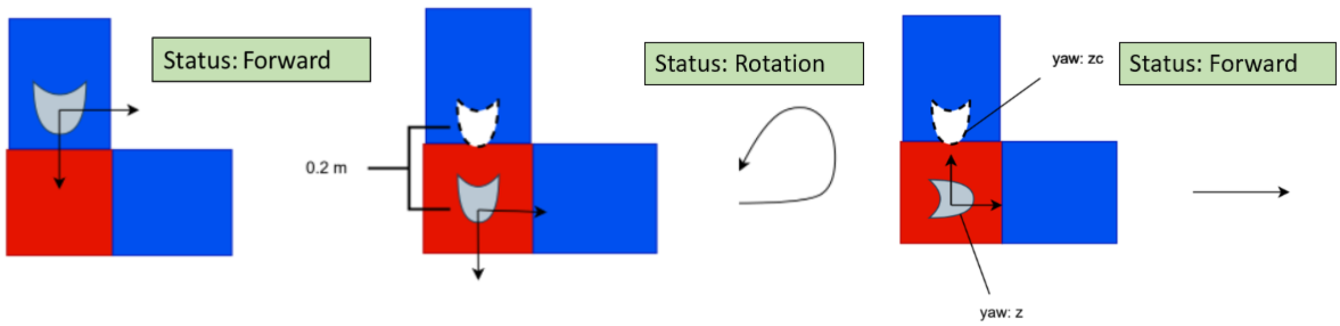


Figure 2

There are mainly 3 steps in this process:

Step1: The thymio is moving forward because both sensors sense 862. Additionally a deepcopy of the robot's odometry is created constantly, until the thymio is in the blue section the copy's odometry is equal to the current actual odometry. When sensors sense 730 in the red area no new command is published and therefore thymio keeps moving forward. Same goes for the deepcopies, the last copy is left at the end of the blue lane before the red square.

Step 2: Once the distance between the thymio's odometry and Deepcopy odometry with respect to x or y is above 0.2 m the thymio stops the forward motion and beings to turn around the z axis. To make this turn as close as possible to 90 degrees we always used the deepcopy together with trigonometry. More specifically we used the following formula:

$$\cos(z - zc) \geq 0$$

z → Thymio yaw

zc → deepcopy yaw

we put this condition on the if statement that makes the robot rotate. When it just starts to rotate the value of this expression is 1 as z and zc are equal and the cosine of 0 is 1. As the rotation takes place the value of the expression gradually gets closer to 0 and when it happens, we know that the robot performed a 90-degree turn expect for errors. Then with another statement we made so that once the turn is completed it moves very slowly (we will see why) towards the exit of the red square.

Step 3: Once the two sensors see 862 again the thymio starts moving again at normal speed and the process is then repeated indefinitely.

IV) Adjusting:

Problem: This is the most interesting part of the project. Until now we described the scenario and what the robot is supposed to do in the environment. In this part we will discuss the solutions we have found when the robot makes errors, more specifically when it comes to the 90 degrees turn. The robot performs very small errors in each red square turning a little less or a little more than the desired angle. Without our adjustment ideas these small errors would soon pile up, leading the thymio to go out of the square. From the tests we've run no more than 6/7 good quality* laps can be made without adjusting the trajectory. Another problem is represented by the start, the thymio could be positioned in a slightly off optimal way and, as the rotation errors, the starting error contributes to the thymio losing the main path. One of our stress tests is based on purposely placing the thymio in strange ways and see if, with our adjustments, it can regain the right way.

*From now on with good quality we refer to laps were all 4 of the segments are crossed by the thymio on the middle of them as it is intended to be.

Solution: as we previously did, we provide a series of images that can help to better explain our idea:

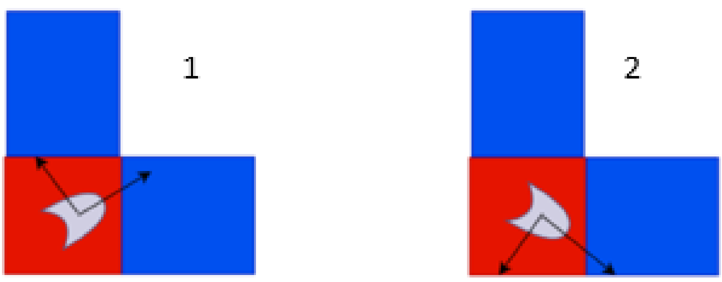


Figure 3

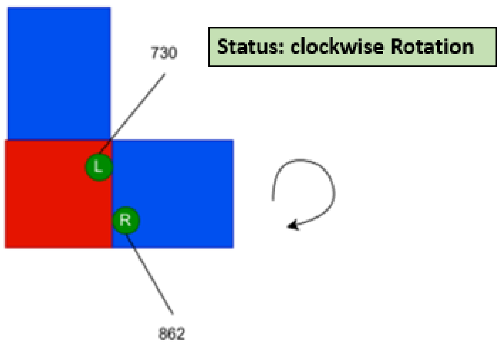


Figure 4

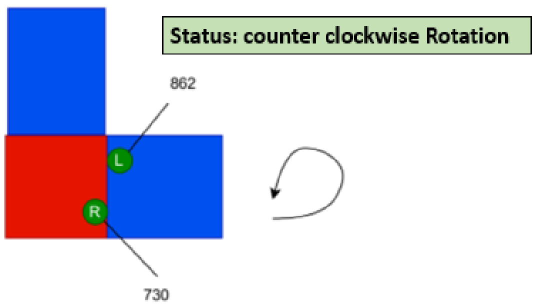


Figure 5

As you can see the adjustments we want to make are on the exit of the red square phase. That is, once the robot performed the 90 degree turn, it adjusts his position before going on the blue lane. Now you can see why having 2 different sensors, R and L, is important. The idea is to adjust the position based on the information given by the sensor. Odometry cannot be used to adjust the position because we are trying to mitigate the errors of odometry itself, the only viable solution is sensing the environment and take action based on it. In this case the robot can be wrong in 2 ways, either it is inclined too much facing the diagonal of the square (the whole scene itself, figure 4) or facing outwards (figure 5). If the robot moves extremely slowly, approximately 0.01 m/s while coming out of the red circle with a wrong inclination, there is going to be a moment where the two sensors sense different things as shown in the figures. In this moment a clockwise or counter-clockwise rotation is made according to the situation. In most cases this allows to reach this position:

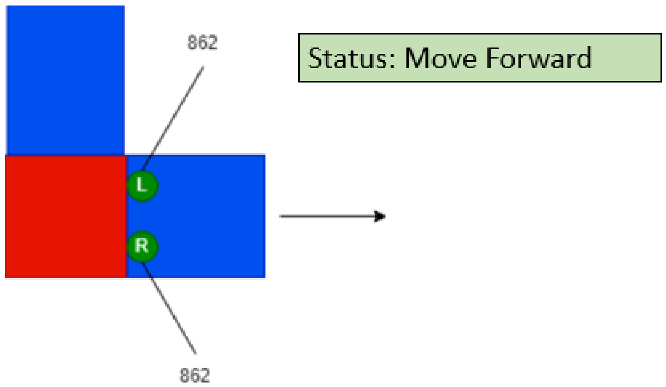


Figure 6

Once both sensors read 862 the robot moves forward as usual. I said most cases because for some angles the thymio cannot sense 862 on both sides. This problem was solved by making it perform a 20-degree turn using a similar method for the 90-degree one previously discussed involving deepcopies, sensors and odometry. This is more like an assurance that covers us in case the normal approach doesn't work.

Why sometimes it doesn't work? → the two sensors are extremely close to each other; therefore, this adjustment starts to work when the error is significant. That is also the reason why the exit of the red square is made extremely slowly. If the robot moves at regular speed, it is not able to sense a difference because the distance of the sensors makes it almost imperceptible, and we are therefore unable to give it the command to adjust its position accordingly. In short it is a very high precision adjustment that thymio is not suited for, so low speeds are necessary.

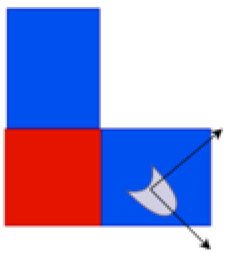
V) Safety net/out of bounds:

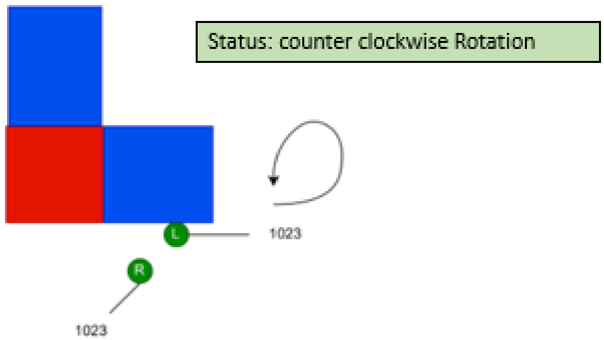
Theoretically speaking the robot can now go on forever and stay on the right path. There are two cases that make it impossible:

Problem 1, Unexpected errors: there are errors that can happen involving variables outside of our control. In rare instances, as an example, the thymio keeps moving straight on the red square because the deepcopy is not created/doesn't work properly.

Problem 2, Bad starting placement: this is going to be one of our stress tests, if we put the robot in a purposely terrible starting position (maybe facing outside of the square on a blue segment) the robot will simply move out of it and never return

To make up for these extreme circumstances we implemented a simple safety net that makes sure that the robot never leaves the square. As usual an image is very useful to visualize the process:





Once both sensors receive back 1023 it means that the thymio is facing out of bounds. When it happens it creates a deepcopy and rotates about 0.3 radiants. If the rotation isn't enough, suppose for example that the thymio was placed perpendicular to the border, it just creates a new deepcopy as both sensors will still receive 1023 and does another 0.3 radiants turn. Eventually it will always go back and start doing good quality laps around the map.

We implemented this kind of adjustment only using a counter-clockwise turn because the robot only goes out on the external border of the square, not the internal one. Plus, even if it did, with an always counter-clockwise rotation it is eventually able to return on track. We did an experiment positioning the thymio facing inward the scene square and it works. Ideally, we should give it a command like the one discussed in chapter IV, but the closeness of the sensors makes it very difficult. As we said this is more a safety net for extreme circumstances rather than a precise adjustment.

VI) Stress tests:

We now added some disturbances in the system to see which are the strengths and weaknesses of our node and how it could potentially be improved.

Test 1, Velocity changes: as we discussed previously the adjustments the thymio makes are possible only when it comes out of the red square very slowly at 0.01 m/s speed. We tried to change speeds to see what happens and the result are summarized by the table below:

Speed (S)	Good quality laps performed (L)
0.01 m/s ≤ S	$L > 90\%$

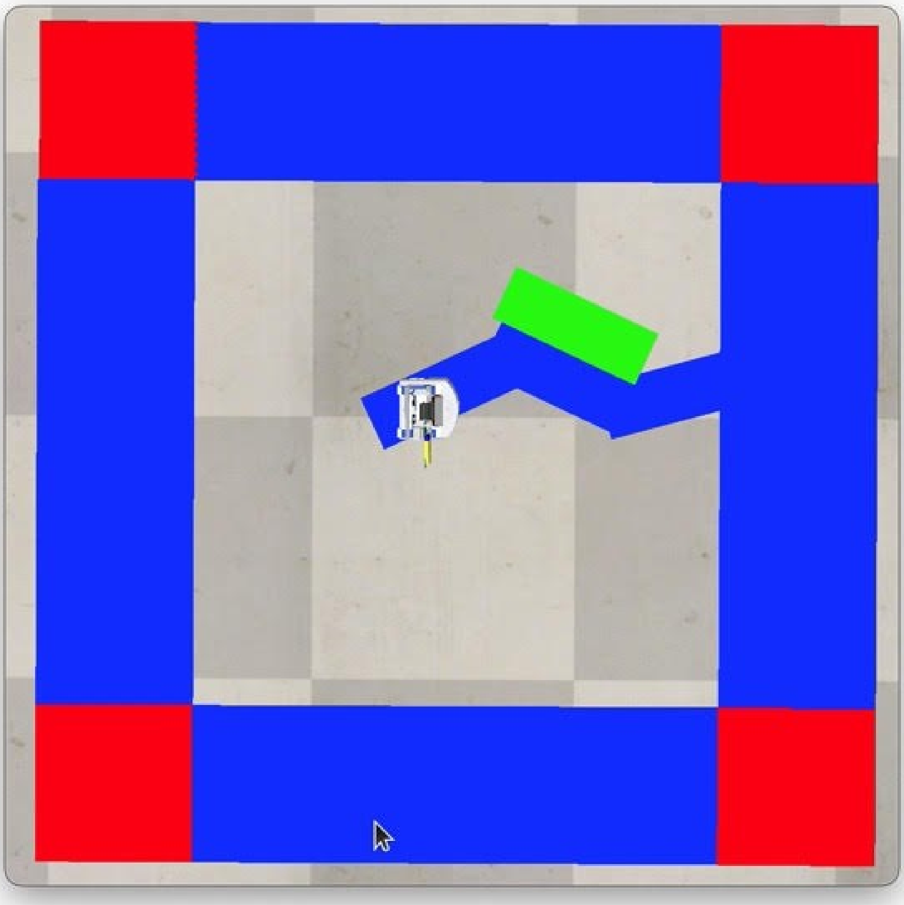
0.1 m /s ≥ S	$L \leq 10\%$

Possible solution: we think that this cannot be solved as this is a problem that is very specific to the thymio, which is a robot that is simply not suited for this kind of tasks. The best thing to do is to change robot, having one that has more precise ground sensors and a higher number of them, placed in the different parts of the robot body.

Any other velocity changes (as an example the linear speed in the blue area or the turning speed) doesn't cause any problems after the adjustment methods have been implemented.

Test 2, angle changes: we tried to voluntarily add a substantial error the turn made in the red squares. Instead of doing it by 90-degree we made it perform a 100-degree turn. To our enjoyment the robot is still able to perform around 90% of the times good quality laps.

Test 3, Strange start: we made the thymio start outside of the circuit, below you can find a screenshot of the situation:



We combined the Test 2 angle error and the thymio was still able to eventually reach the desired path/direction. It takes one lap to adjust its position and run as intended. Test 2 and 3 are, in our opinion, pretty successful overall.

VII) Possible improvements and conclusion:

Working and exploring the functionalities of the thymio was very interesting to us, it was an occasion to learn something new and refresh our knowledge on previously learned concepts such as odometry and trigonometry. The only case where our thymio would not be able to perform anything is when the starting place is outside of the square. A possible solution could be to tell it that, if it senses 1023 on both sides for more than tot amount of time, it starts exploring randomly the room in search for the coloured square. Another improvement could be the addition of proximity sensors that we used in assignment 2 to let it handle possible obstacles over the road or, why not, other thymio's.