Line following system

The lines are 18 mm wide and are white on a black background.

There are also red and green lines at the boxes where the cubes need to be deposited.

It is quite hard to distinguish the coloured tape from the white tape. I therefore suggest not knowing, and simply keeping track of where the robot is on the table, so it knows where to go. Taking an analogue input was tested but found to be unsuccessful.

We are supplied with four IR sensors that are suitable for the line following system.

The track will be split into paths and nodes.

* There 10 types of nodes, see diagrams below.
* Some paths contain corners, which are curved with a radius of curvature of roughly 30 cm

I propose using an array of four IR line-sensors, arranged as shown:

IR sensor

IR sensor

IR sensor

15 mm

15 mm

Figure - Arrangement of line sensors

IR sensor

10 mm

Direction of Travel

The idea behind this is that the robot will probably oscillate from side to side as it follows the line, and so if it is straight and centred, the middle two should be white and the outer two should be black. If it goes a bit to one side, one of the middle two sensors will go black while the other stays white, and then the robot knows to turn back and go slightly the other way. If more than two sensors read white, then either the robot has reached a junction, or it is at such a bad angle that it is almost perpendicular to the line.

Critical angle where it is possible for three sensors to read white on a straight line:

I think it is reasonable to assume that the robot will not be at an angle more wrong than 45°, but this is an edge case that will still need to be considered in the code.

When the robot approaches a T junction, from one of the side branches, assuming it is driving straight and centred along the line, the central two sensors will read white, as will one of the far side sensors, while the other side sensor remains black.

If we keep track of where the robot is on the table, it will be expecting to come across this junction, and will know which way it wants to go. If it wants to go straight on, it should drive straight forward to pass the line off to the side, and hopefully stay on track. It can then update its known position. If it wants to go off to the side, it can reverse slightly and then turn to that side and the fourth sensor that was black should briefly become white as it crosses the line, and then hopefully find the new track and be able to follow it off to the side. Diagram

Description automatically generated

1st

2nd

3rd

Figure - Map of the table with each position labelled

Each node and path section have been numbered as shown in the image above. This will allow the robot to be able to keep track of where it is on the table and define where it needs to go next. Some assumptions about the route we will decide on will temporarily be made:

* We will go anti-clockwise around the table, crossing the ramp to get to the “collection” half, and going under the tunnel to get to the “delivery” half.
* Begin in the start box
* Node 1, go straight across
* Path 2
* Node 3, turn right
* Path 4
* Node 5, go straight on
* Path 6, with a 90° curve and over the ramp
* Node 7, go straight across
* Path 8
* Collect cube
* Node 9, go straight across
* Path 10
* Node 11, go straight across
* Path 12, with a 90° curve
* Node 13, go straight on under the tunnel
* Path 14, with a 90° curve

If the block is low density:

* Node 15, turn right
* Path 21
* Node 22, go straight across and deposit block
* Node 22, reverse back across
* Path 21, reverse back
* Node 15, reverse left
* Path 14, go forwards
* Node 15, go straight across
* Path 16
* Node 3, go straight on
* Path 4
* Node 5, go straight on

Or, if the block is high density:

* Node 15, o straight on
* Path 16
* Node 3, go straight on
* Path 4
* Node 5, turn right
* Path 17
* Node 18, go straight across and deposit block
* Node 18, reverse back across
* Path 17, reverse back
* Node 5, reverse left
* Path 4, go forwards
* Node 5, go straight across

Then repeat similarly for the 2nd and 3rd blocks

Node Types:

