

AI Robotics

Project 2

A Line Following Robot

Department of Computer Science
Texas Tech University

Rajat Goyal
goyal@cs.ttu.edu

Karan M. Gupta
gupta@cs.ttu.edu

Abstract

This report discusses the implementation of a line following behavior for autonomous robot navigation using reactive behaviors. It includes the various design issues encountered at different phases of this project. The project was implemented using Lego MindStorms Robotics Invention System 2.0 and the Not Quite C programming language construct.

Software Design

Specification & Analysis: Ecological niche

The Task

The task for the robot was to follow a path laid out on marble speckled flooring using a black electrical tape. The black tape was 3/4" thick. The dynamics of line were not specified as part of the project specification. It may not be straight and may even cross over itself. There were no turns greater than 90 degree. The project was to be graded on the following criteria:

1. The robot must traverse the line in less than ten minutes from when it started.
2. The robot is not to leave the line. If it does, it must start over from the beginning; however the overall time will not restart.

It would additionally be evaluated for the following criteria:

1. Creativity of design and function
2. Overall speed and effectiveness of design
3. Decorative style

The Robot

The robot was supposed to be built using the LEGO MindStorms Robotics Invention System, and thus inherits the limitations and constraints introduced by the said system. It was supposed to be an autonomous agent, i.e. does all its computing using an on-board processor with no communication link to an off-board processor.

The agent was running the RCX 2.0 firmware. The Not Quite C (NQC) programming language was used to implement the behavior-based (reactive)

algorithms. The project was implemented on the Windows XP platform with the BricX NQC Development Environment.

The sensor suite consisted of 3 LEGO light sensors attached to the front of the robot facing downwards. The sensors were attached adjacent to each other, with the middle one protruding a little forward. These light sensors are active sensors that consist of a LED, and a sensor which detects the reflected light. The LEGO light sensor reads light from 0.6 lux through 760 lux. It is sensitive only to the intensity of light and cannot differentiate between colors. The robot has a turning radius of around 75 degrees. It is powered by 2 9V motors attached to the front wheels. The gearing ratio was 1:5. The rear wheels do not have a power source attached to them.

It is divided into 2 segments, the front segment carrying the motors and the sensors and the rear segment (trailer) carrying the RCX brick, connected through a pivot which allows the front part to steer irrespective of the trailer.

The Environment

The path is laid out on a marble speckled floor using a black electrical tape. The width of the tape is 3/4". The exact length or shape of the course was unknown until the day of the demonstration. The analysis of the environment revealed only one affordance; the absence or presence of the black tape. The robot had only to check for the black tape to know if it was on the course or not. The robot could easily extract this affordance from the environment using the 3 light sensors. The left and the right sensors act

like the "eyes" of the robot, albeit disconnected, offering no stereoscopic perception of the environment. The middle sensor is the "ESP" (Extra Sensory Perception) of the robot. It being a little ahead of the other sensors afforded a look-ahead for the robot.

The only thing the robot required to do was to keep the black line beneath the middle sensor and it would stay on track. The other 2 sensors are used for detecting the direction in which the middle sensor has strayed away from the path.

Design

Behaviors

The perceptual schema for this robot consisted of detection of black line for the ESP. As long as the black line was being detected by the ESP the robot would keep moving forward. If the ESP lost track of the black line then the other 2 eyes would be polled to check which one was detecting a black line. The motor schema would turn the robot in the direction in which the black line was detected until it brought the ESP back onto the black line.

The initial behavior table is shown in Figure 1.1 (page 4).

The "follow line" behavior is a stimulus-response behavior and is reflexive, i.e. the response lasts only as long as the stimulus. The "find black line" behavior is a stimulus-response behavior and is taxis, i.e. it orients the robot relative to the stimulus.

The behaviors are *implicitly chained* together with their releasers. The

emergent behavior is to follow the black line at all costs.

Implementation & Testing

Implementation

The implementation of the behaviors was done using NQC. The 2 behaviors selected were based on the affordance offered by the environment, viz. the black line.

The presence of the black line acts as a releaser for the “line following” behavior. The motor schema moves the robot forward if it detects a black line under the ESP, irrespective of what the eyes are sensing.

The absence of a black line inhibits the line following behavior and in turn releases the “find black line” behavior. The placement of the eyes ensures that at least one eye will be focused on black when the ESP deviates from the path. The eyes are polled one by one to check which of the eye is sensing a black line and the robot is turned that direction until the ESP is once again located on top of a black line.

The sequence in which the eyes are polled might suggest a tendency towards favoring one eye over the other. One might be tempted to think that the behavior would fail if both the eyes were pointing at black at the same time. This fallacy is overcome by polling the other eye once the turn procedure has been initiated and if it detects black too then the motor schema is inhibited and control passes onto the check for the other eye. The same check is performed there and if this too determines that both eyes are poised over black then there is a high probability that the robot is at a

junction (a crossing). Therefore the turn is inhibited in this motor schema too and the move forward behavior is activated once more.

The new behavior table is shown in Figure 1.2 (page 4).

The first 2 behaviors retain their properties. The “inhibit turn” behavior is a stimulus-response behavior and is a fixed-action pattern.

Testing

The individual behaviors were tested in a dry run. The robot lay upside down and the program was run. The individual sensors were triggered for different configurations, using a piece of black cloth. The direction of the spin in the wheels was used to determine what motor schema was active at that moment.

System Testing

The final testing of the combination of behaviors was done by a sample track laid down on the floor. This track had acute angle turns, obtuse angle turns, 90 degree turns, perpendicular junctions, skewed junctions, and curves and bends.

Figure 1.1
Initial Behavior Table

Releaser	Inhibited by	Behavior	Motor Schema	Percept	Perceptual Schema
Always on	ESP not on black line	Follow line	Move forward	Black line under ESP	Check for black line
Always on	ESP on black line	Find black line	Turn until ESP back on black	Black line under an eye.	Check for black line.

Figure 1.2
Final Behavior Table

Releaser	Inhibited by	Behavior	Motor Schema	Percept	Perceptual Schema
Always on	ESP not on black line	Follow line	Move forward	Black line under ESP	Check for black line
Always on	Both eyes on black	Find black line	Turn until ESP back on black	Black line under an eye.	Check for black line.
Both eyes on black	Other eye not on black	Inhibit turn	Halts the turn	Black line under both eyes	Check for black line at other eye

Problems Encountered & Solutions

Initially this robot was built to work using only one sensor. There was a single wheel in the front which could turn left or right, much like the front wheel of a motorcycle. A single light sensor was attached to this front wheel enabling both to turn in the same direction. There was one motor rotating the rear wheel axle. The light sensor

would move left or right (covering roughly 180 degrees) and "sniff" for the black line. When it would find the black line, it was to give power to the motors pushing the robot forward in the direction dictated by the front wheel. However, this design was abandoned at an early stage because it would not be possible for it to differentiate between an intersection and an actual curve. This could cause the robot to take the wrong direction at a curve.

The robot design was changed completely for the second attempt. The motors were placed in the front, one for each wheel, thus giving the robot a front-wheel drive. The light sensors were situated adjacent to each other and very close to the ground. The light sensors were programmed to basically stay off the black line. If a light sensor was to come over black, it would cause the robot to turn in the opposite direction, effectively keeping the robot centered over the black line. However, even with design there were problems when both the sensors came over the black line at the same time. The robot would not know whether to make a 90 degree turn (that too, in which direction?) or to go straight assuming it is an intersection. The behavior specification did not handle this situation effectively. The robot was overshooting at turns and entering into a situation which could not be handled with just two sensors. Several attempts were made to reengineer this design to somehow improve the light detection and the turning. The light sensors were placed a little far apart which helped in making the correct turns, as there was very little chance that both the sensors would come on the black line at the same time. The wheels were geared down, to decrease the speed of the robot. A double-axle pivot was added to the design of the robot. This separated the robot into two parts, one which was the front-wheel drive, and the rear which was a trailer for the RCX. This last change greatly helped in turns. The front wheels now had only their own weight to move. There was power coming in to both the outer and the inner wheel during a curve, allowing the robot to turn almost about its center. However, the robot would still end up in situations it was not designed

to handle, like getting completely off the black line, and at times, both sensors would come over the black line.

In the third design, another light sensor was added to the robot which would follow the black line, unlike the previous two sensors which basically, kept themselves off the black line. This sensor was placed between the other two sensors. The middle sensor is to behave like a third eye, keeping the robot on the black line. As long as the middle sensor is detecting black, the robot must go straight irrespective of what the side sensors are detecting. If the middle sensor starts to see white, the side sensors were polled and turned accordingly depending on which one was detecting black first. There were still some problems. The robot was moving too fast and overshooting the turns, sometimes putting all three sensors over a white area, effectively going blind. To solve these problems the middle sensor was placed a little ahead of the other two sensors. This provided a certain look-ahead and immediately caused a release of the behavior which would check the side sensors for black line detection. The pivot was modified. There was to be only one single axle on which both parts could rotate, thereby reducing the amount of free rotation. Gearing of the front-wheels was changed and the robot was made even slower than it was. This solved almost all of our problems and the robot had no problems with simple turns, intersections and definitely no problems with straight lines. However, sometimes at a 90 degree the turn the robot would still end up in a position so that the side sensors were both on the black line, thereby confusing the robot. It would either just get stuck, or go off the black line

completely turning away from the track. So the side sensors were spaced further apart and that solved this problem. There was only one last problem left to take care of. The robot could orient itself to the black line such that if there is a 90 degree turn, say to the right, the middle sensor would see white, and the left sensor would see the black line before the right sensor. This would cause the robot to believe that there is a left turn and then it would keep turning left until the middle sensor saw black again. This would happen when the robot would have taken a complete u-turn. This situation occurred when the robot would travel into a turn. To solve this, the behaviors were modified: stop turning if before the middle sensor detects black the other sensor has detected black.

and both over the black line. The percepts would then be to see which sensor has left the black line and turn the robot accordingly to come back onto the black line.

Other Ideas

Three separate implementations were attempted to try and complete this project successfully. The first was to sniff for the black line using only one sensor. Another implementation was using two sensors to detect which sensor has detected the black line and accordingly move away from it. The final attempt was a combination of the above two and used three sensors. The middle sensor kept the robot on the black line, and the side sensors avoided the black line.

Another way to solve this project would be to use two light sensors arranged to form a “L” shape such that one of them is on the white and the other is on the black line.

Yet another method could involve using two sensors kept very close to each other