CSC 714 – Spring 2014

NXT Autonomous Retriever

FINAL REPORT

Aakash Dave

Contents

Introduction

There is a need for automation in the industry as it reduces cost and complexity of operations. Autonomous bots cater to this need by performing various functions in places like warehouses and assemble lines. They can be programmed to perform repetitive tasks with great efficiency. Also they can adapt to the environment for performing the task at hand. If properly implemented bots can work better and faster than humans in most cases. Current examples can be found in the form of Warehouse Management systems which provide retailers with a complete supply chain solution.

Apart from industrial uses autonomous bots are also moving into our homes. Automation of routine tasks like cleaning, monitoring with the help of bots is becoming more common. As they grow smarter and more adept at these tasks most daily household work won’t need much human attention in the near future.

In this project the focus is on huge indoor spaces like warehouse, libraries and assembly lines where we can see a shift towards automation. The idea is to use a bot to follow certain path and look for certain locations where it stops and then can continue its normal operation.

Problem Statement

Most innovations in this field involve a great degree of improvement in basic robot algorithms like path searching and following along with obstacle detection. The ability to follow a complex path with branches, different path sizes and turns is the focus of this project. Also there is a need to implement a stop/start mechanism where the bot can wait at a location/take input wirelessly from a controller.

The aim of the project is to implement an autonomous vehicle which is used for following a path and perform start/stop at a given location on the track. The bot is to take input from a Bluetooth controller.

Implementation

(i) *Outline*

1. The bot will follow a path/track based on user input from a Bluetooth enabled NXT.

2.User will be able to pair the bot with a controller NXT.

3. Once paired it can take input from the controller and implement various path maneuvers like which branch to follow.

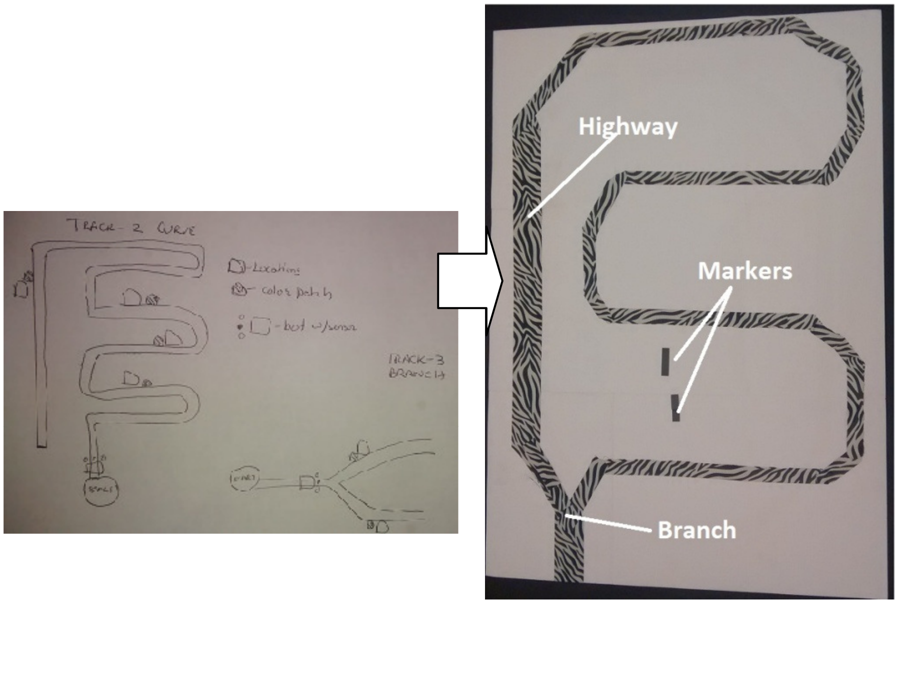
4.The bot can be programmed to stop at a certain location. After the bot reaches the location on the track it will wait for a while before it starts to follow the path again. This emulates the retrieval mechanism.

5.Also the design allows the bot to follow tracks of different width. Hence an algorithm can direct the bot to take the wider track(highway) which is faster if the given location is on the other side of the track.

(ii) *Track Design –*

Since this project simulates retrieval of items via start/stop method track design is very important. It determines program logic and bot design. Various track designs were looked into which can be used in a warehouse/assembly line. Branch and Conveyer line are the most preferred options.

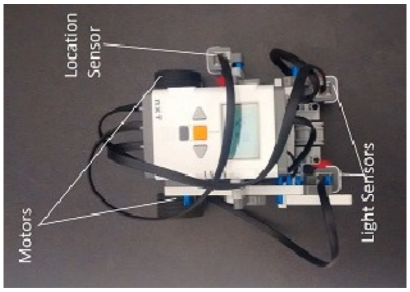
Below is the sketch of initial design and its implementation for final run.



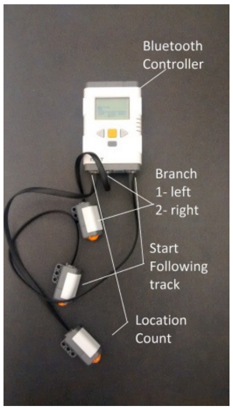
(ii) *Bot/Controller Design –*

Bot -

The bot uses a 3-wheel two motor chassis to accurately follow the track. The front wheel is free so as to allow easier turns on the paper surface. Two light sensors are placed in the front for following the line. Also another light sensor is used to detect markers to denote stop locations.

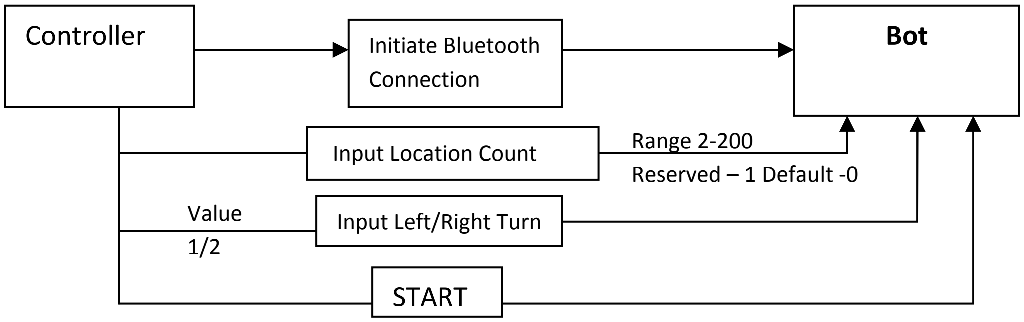


Controller – Sends commands via Bluetooth. The bot waits to establish a connection and input from the controller. A touch button is used as start command.



Working and Experiments

The bot is started first and waits to pair with the controller at the start position. Once paired it waits until all inputs including start command are received. The design allows for commands to be received only once before the bot starts the run. All inputs are implemented as touch counters where a single press indicates a unit value. The first input is location, based on the count the bot does stop/wait at value -> location -1.After the stop is made however the bot ignores the counter and does not stop even if it detects any other stop markers. Second is branch, here the bot gets the value from the counter to determine if it is supposed to take a left or right turn when it encounters a branch. Current design supports a single branch at the start of the track so as to either take the main path or a shortcut also called the highway for easy access to a far off location. Once start command is given the above values are transmitted from the controller to the bot and it starts the line follow logic. Below is a flowchart of the explanation -



Experiments -

1) BT- yes/Location – 3 /Turn – 2 /Start – 1 -> Correct
   The bot starts and takes a right turn when it detects a branch and follows the track. The counter decrements whenever the location sensor detects a marker and the bot stops at the second(location -1) marker for a given time and then starts again. Clockwise follow confirmed.
2) BT-yes/Location – 2/Turn – 1/Start -1 -> Correct
   The bot does the above and takes a left turn along with stopping on the first location marker after the highway. Counter clockwise follow confirmed
3) BT-yes/Location – 0/Turn – 1,2/Start – 1 -> Incorrect
   Bot does not start as default stop event is active even a start command is received.
4) BT-yes/Location – 2-200/Turn – 1,2/Start – 0 -> Correct
   Bot waits on start until input is received.
5) BT-no/Location – 2-200/Turn – 1,2/Start-1 -> Incorrect
   Bot does not start even if start is 1 as a connection was not established before inputs were entered.

Challenges

Bluetooth

1) Pairing NXT’s via Bluetooth works by starting the bot first and then the master other wise the connection is not established.
2) Transferring data between devices is done via a 32-byte buffer hence the different inputs need to be assigned in the proper format.
3) There were issues in getting the color sensor and Bluetooth to work together which was solved by initializing and setting all the sensors in init before Bluetooth connection is initiated

Control logic

1) Combination of cyclic alarm and event based tasks needs careful attention when writing code.

2) Bluetooth input control between the bot and controller is also an issue since it is a one shot design and the values can easily be missed if a procedure is not used to input values. This can be solved by sending values only after a connection has been established and all inputs have been entered. A start button is used to send those values for which the bot waits in its default state.

3) Branching is tricky as the bot has to make a left/right turn whenever it detects a branch. The implementation demands a routing which stops the line follow and instead makes a turn and then hands over control.

4)Accurate detection of markers so that the bot stops at the precise location.

Future Work

There is a lot that can be improved in the above application. Currently a basic track with branches and sharp turns has been implemented. This can be changed to a grid based design which would require a compass either from an Android device or stand alone. The grid track also demands a change in bot design which has to be more agile than the current 3 wheel design to make 90 degree turns.

Once implemented the bot can implement a map based algorithm and target given location on the grid. Also multiple paths can be found before going to the location via shortest path algorithms to increase efficiency. This is scalable to industrial standards as many current products like Kiva systems use these techniques to effectively manage warehouses.

A Real time system consisting of autonomous bots along with a central controller could effectively help with the problem of item retrieval from various locations in the future.