Smart Overtaking Bot
(Group 9)

As a course project of CS 308 – Embedded Systems Lab
under the guidance of Prof. Kavi Arya

Saurabh Bhola 09005022
Kanwal Prakash Singh 09005031
Abhishek Kabra 09005037
Kapil Dubey 09005038

Contents:

Diagrams and Figures:

Introduction:

The main idea of our project is to build a close to real world situation where vehicles with higher speeds can safely overtake the vehicles moving ahead of them with lower speeds. The overtaking vehicle is the I-Bot and the vehicles that come on its way (with lesser and constant speed) are vehicle bots.

Components of our Project:

Vehicle-Bot:
A bot that is assumed to be moving at constant speed on the given path

I-Bot:
A bot that senses a vehicle-bot in front of it, analyzes its speed, the distance between the vehicle-bot and itself and then overtakes it based upon the calculation it performs.

Initial Efforts in this direction – 'the shoulders that we stand on':

Cruise Control: Allows driver to set a driving speed

Adaptive Cruise Control: Vehicle is capable of following a leading car by autonomous action on throttle and brake panels.

Overtaking Maneuver:

A driving action involving a rapid and 'intelligent' movement to negotiate an obstacle. It includes movement of steering wheel (path change) and speed increase. But overtaking errors resulting from failure to accurately and timely interpret information about other vehicles in close proximity may result in catastrophic accidents.

How to avoid collisions?

Intelligent algorithms addressing following real time issues need to be incorporated:

1. Calculating proximity to other vehicles
2. Determine when the lane change maneuver should start (most research in this field has been on lane following)

3. Developing optimal and safe trajectories

Key Challenges involved in our Project:
The biggest challenge is to safely overtake the vehicles taking optimum path.
Other challenges include:

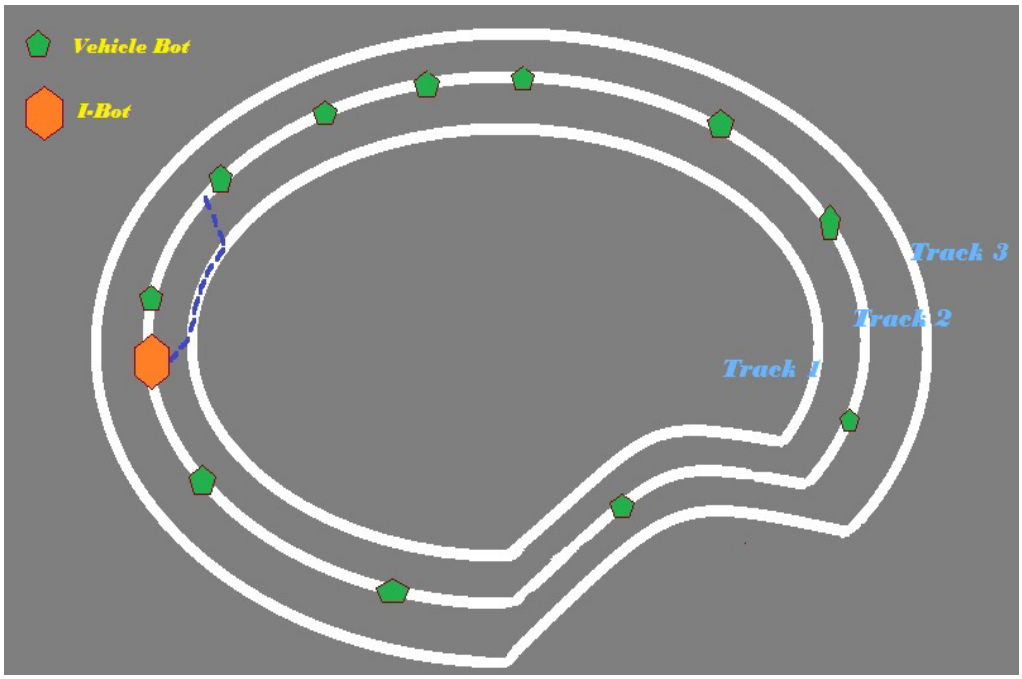
1. Estimation of speed of Vehicle Bot as well as an estimation of the acceleration needed for I-Bot using the sensors
2. Deciding on the optimal path
3. Safely regaining the path back sensing a vehicle(s), if any, in its vicinity

All the above challenges (trajectory planning) should be addressed in the absence as well as presence of vehicles in the passing lane.
Also, a safety distance should be ensured between the vehicles in order to avoid any collision

Addressing the challenges:

The bots move in an arena with **three white lines** – the underlying reason being use of white line sensors for the movement of the bots. The tracks are *curved* and make three parallel loops (to avoid the limitation of straight-line overtaking reaching an end)

Is three track idea a good choice?



The Three Parallel Tracks
It is based on our idea of choosing the shorter path while overtaking over a curve as we know the smaller the radius of curvature of the overtaking path the smaller would be the path to traverse while overtaking.

But this also adds to the complexity involved in keeping control on the speed of the bot over the curved path, when the bot has to constantly monitor its own path so that it doesn't lose its track while overtaking.

All the bots shall move in the middle track in the initial situation in unidirectional manner with I-Bot following the Vehicle-Bots.

Work Flow Diagram:

Innovation and Challenges:

- Further complexities involve **having a number of vehicle-bots** on the middle track, as well as on the overtaking tracks – to check how safely can our I-bot overtake
- Adding more complexity, we have the ***speeds of the vehicle bots varying***, that makes situation even difficult for a safe overtaking as the vehicle bot might increase its speed while the I-bot is overtaking and regaining its path back which might lead to an accident

Raising the bar further – reaching it!

- A large number of obstacles (due to spark bots not functioning well) were introduced in the three tracks which are sensed by the camera installed on our I-Bot and hence it behaves accordingly
- It detects a bot/obstacle through the camera as well as sensors and tries to overtake it safely.
- In case a bot is moving in opposite direction in the overtaking path, it stops following Adaptive Cruise Control
- For accommodating the added feature of varying speeds of vehicle bots, it has left and right sharp sensors, giving it instructions regarding presence of any vehicle in their respective sides so that it does not overtake until one of the paths, preferably optimal, is free and also safely regains back its original path

Problems Faced while run-throughs:

- Firebird didn't work perfectly many times leading to re-running the code everytime for better results
- There were issues like Firebird losing its track despite correct code – the problem was deduced to be with the sensors and sometimes with the reflection of light on shiny black surface
- The results came out to be better with more number of run throughs

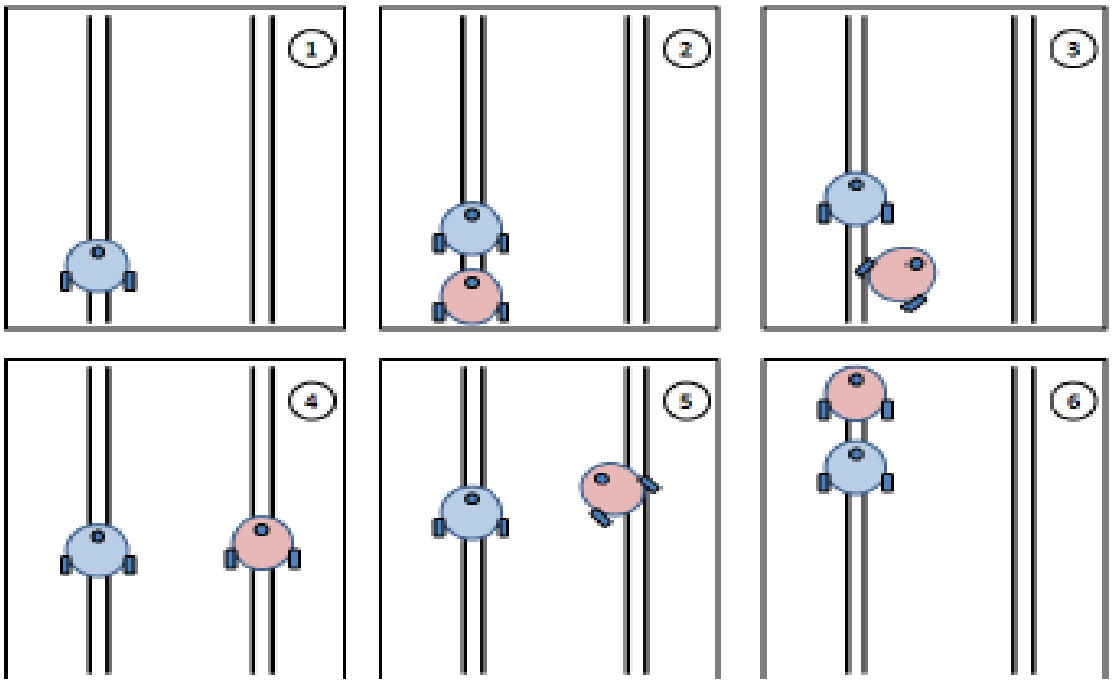
Two Scenarios – 1

1) Simple Overtaking Scenario

I-Bot travelling with velocity V pursuing vehicle bot with a slower velocity W

Overtaking Manoeuvre:

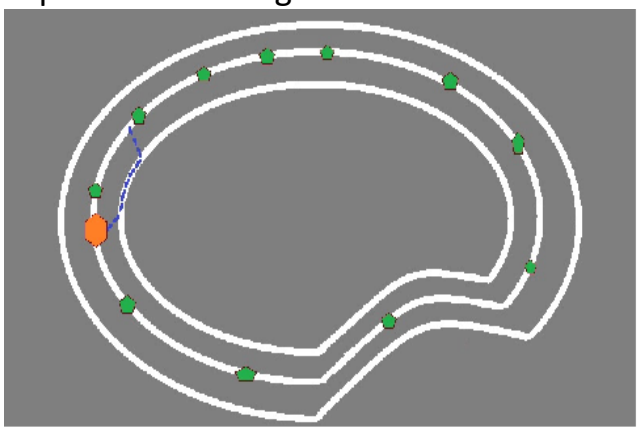
1) Move from driving lane to passing lane
2) Travel in passing lane till you overtake
3) Return to the driving lane



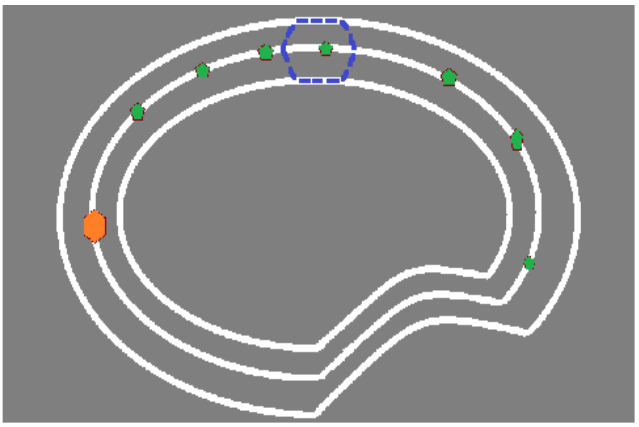
Two Scenarios – 2

2) In presence of another vehicle bots in the passing lane, and bots with varying speed

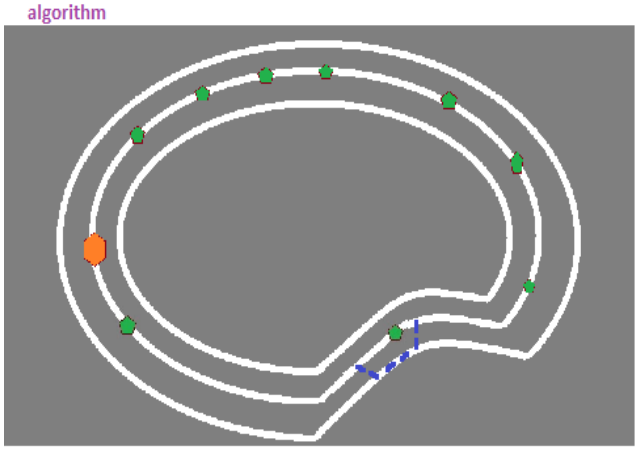
Every situation and the strategy involved while overtaking in it is well explained in the figure on next slide



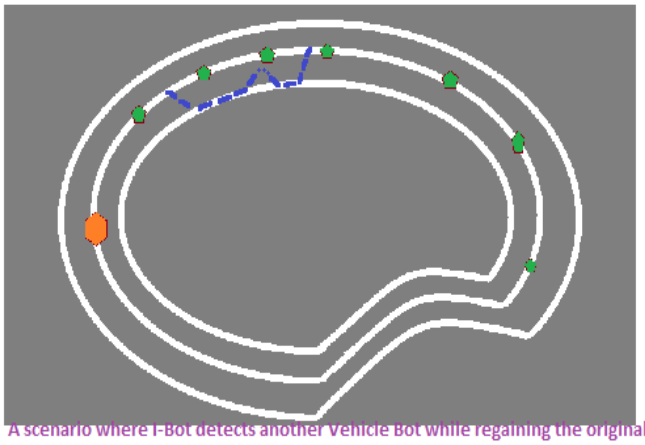
Overtaking over a curvature, I-Bot takes Track 1 to overtake based on its algorithm



Overtaking over a nearly linear path, I-Bot can take either of the two tracks, Track 1 or Track 3 for overtaking



Overtaking over another curvature, now I-Bot takes Track 3 for overtaking

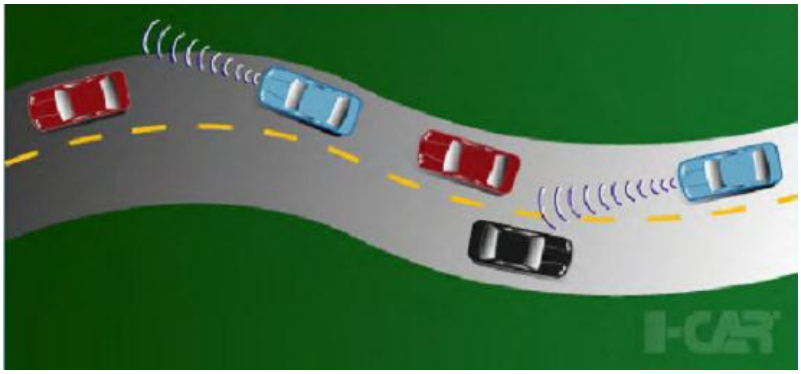


A scenario where I-Bot detects another vehicle Bot while regaining the original pa and thus moves back to the overtaking path, and again tries, this time successfully

Extreme Situations of failure (that can't be avoided):

a) When there is a clump of vehicle bots and no possibility for I-Bot to move in either lane, hence it should take precaution and slow down to the speed of the vehicle ahead.
b) When the path are too curved, and hence keeping the bots following white line becomes too difficult

Refer Figure on next page



An Extreme Situation of Failure

Re-usability of our code and Future Enhancements:

The most important aspect of our project is being in **close proximity with the real life situation** of traffic on roads, where there are a lot if vehicles moving in a single lane, with no vehicle moving at constant speed and also, continuous overtaking between vehicles.
Also it is built upon the idea of Cruise Control and ACC which are too ideal to be implemented in real life
It not only allows overtaking but also has major focus on SAFE OVERTAKING and prevents all kinds of accidents that might happen due to heavy traffic in the original as well as adjacent lanes, and varying speed of the vehicle being overtaken

Some **future projects** that can be built upon our idea:

1. **Smart Race:** A race of overtaking bots taking optimum paths (following algorithms) while overtaking
2. **E-Coolie:** A coolie that follows its master in a crowd, just like a human
3. **Thief Catching Cars** for Police

Regarding reusability of our code - the Codes – **Smart_Overtaking_Bot.c, lcd.c, and matlabgrp9.m** - for overtaking, safely regaining of path and Image Processing for vehicle detection can be re-used and enhanced as said earlier too.
Apart from Firebird, the Spark Bots and their ACC codes would also work for adding traffic

The Image Processing components and the algorithms involved can be made better in future projects
Summarizing, we can say it is a good start for a mega project (A *legacy* for future)

"While it is well enough to leave footprints on the sands of time, it is even more important to make sure they point in a commendable direction."
- James Cabell