Planning of Multiple Autonomous Vehicles using RRT

Rahul Kala and Kevin Warwick
School of Cybernetics, School of Systems Engineering,
University of Reading, Whiteknights,
Reading, Berkshire, United Kingdom
rkala001@gmail.com, k.warwick@reading.ac.uk

Citation: R. Kala, K. Warwick (2011) Planning of Multiple Autonomous Vehicles using RRT, *Proceedings of the 10th IEEE International Conference on Cybernetic Intelligent Systems*, Docklands, London, pp 20-25.

Full Version Available At: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6169129

Abstract—**Criteria such as driving safety and overall travel efficiency have led to increasing attempts towards autonomy of vehicles wherein different vehicles can plan their journey, maneuver as per scenario, and communicate to each other to create an error free travel plan. In this paper we present the use of Rapidly Exploring Random Trees (RRT) for the planning of multiple vehicles in traffic scenarios. The planner for each vehicle uses RRT to generate a travel plan. Spline curves are used for smoothing of the path generated by the RRT, which follows non-holonomic constraints. Priority is used as a coordination mechanism wherein a higher priority vehicle attempts to avoid all lower priority vehicles. The planner attempts to find the maximum speed at which the vehicle may travel and the corresponding path. Experimental results show that by using the approach, multiple vehicles may be planned to travel in a fairly complex obstacle grid. Further, the vehicles exhibited behaviors including vehicle following and overtaking which are commonly seen in everyday driving.**

Keywords-intelligent vehicles; autonomous vehicles; robocars; rapidly exploring random trees; priority based planning; planning; coordination.

I. INTRODUCTION

Autonomous vehicles represent the next generation of driving wherein vehicles will be able to traverse on roads and other terrains without the need for human drivers. These vehicles have capabilities to be more efficient as well as safer for driving. Interest towards the use of these vehicles particularly gained interest during the DARPA Grand Challenge [1, 2]. These vehicles consist of sophisticated hardware and software to realize their abilities. Various components that the software might need to deal with are sensors and sensing units, sensor data preprocessing, image processing, object analysis, map building, localization, planning, control, etc [3]. Such vehicles may also be fitted with communication tools which enable multiple vehicles to talk to each other in order to avoid any possible collision and to achieve an optimal plan. This is referred to as inter-vehicle communication [4] and plays a major role in planning, localization, obstacle discovery, etc.

In cases where the road is perfectly divided into speed lanes this eases the planning process by a significant amount. With such an assumption the task of planning is primarily reduced to decision making regarding speed lane change, if any [5]. Although planning the speed of motion may play some part. However the use of speed lanes may not be efficient, especially where vehicles are of variable widths or non-uniform road widths exist – both of which are practically possible cases [6].

Hence a robust planning technique, in order to fully exploit the road infrastructure, needs to be able to plan in the absence of speed lanes. On top of this, the road may have fairly complex obstacle architecture, in some stretches, which also needs to be considered. Simple obstacle avoidance schemes [7] may do fairly well in the presence of a single or a couple of obstacles, wherein the complete obstacle framework is simple. But it is important for the planning algorithm to be robust in planning for a wide road with a complex obstacle framework as well. Obstacles not only include mapped static obstacles but also other moving vehicles which need to be avoided. Considering that the various vehicles can talk to each other using an inter-vehicle communication scheme, it is important to properly coordinate the various vehicles such that the overall travel plan is optimal and no collision is recorded. Computing results in real-time is therefore an important requirement.

Kuwata et al. [8] recently published their planning algorithm used in the DARPA Urban Challenge 2007 for the team MIT entry Talos, with which they secured 4th place in the contest. Complete specifications of the entry of the team can be found at [9, 10]. Their algorithm used the RRT algorithm for planning. Our work in this paper is different on two counts. Firstly we keep the planning and control modules separate and secondly RRT only does the job of planning. This makes the planner more adaptive such that it can be applied to any vehicle whose dynamics may not necessarily be known. Further it creates scope for well-known control techniques to be used, once a planned path is available. In particular we also extend the approach here to the case of multiple vehicles, devising an effective coordination between them. Further we take into account unstructured and curved roads as well. An enhanced representation scheme is introduced in this paper in order to achieve these goals.

II. ALGORITHM

A. Problem Modeling

The basic problem is to plan the paths of a number of vehicles in a traffic scenario. Each vehicle is assumed to be rectangular in shape with its own length and breadth and has its own maximum allowable speed. The generated plan needs to take into account that the operational speed of the vehicle must never exceed this limit. Each vehicle enters the planning scenario at some time. Only on its emergence is planning for a particular vehicle performed. The vehicles must not collide either with each other or with the static obstacles. Vehicles are non-holonomic in nature and hence only smooth travel paths may be traversable.

For the task of planning, the entire road consisting of the route of the vehicle is broken down into small overlapping segments. Planning of each segment is done separately and is initiated whenever any vehicle enters a segment. Initial experiments revealed that the generated plan of any vehicle may be such that the vehicle ends its journey completely surrounded by obstacles or at an improper orientation. Hence the ends of the travel plan may not be desirable as the planning algorithm does not immediately take into account anything beyond the segment being planned. So, the vehicle's route is re-planned before it completes its journey in a segment, using the generated plan, by the method of overlapping segments.

The approach uses two coordinate axis systems. The cartesian coordinate system (XY) which is used for curve generation and plan specification. The second is the road coordinate system (X'Y'). In this the X' axis is the first boundary of the path to be traversed and the Y' axis is taken as the ratio of the distance of the vehicle from the first boundary to the immediate road width at the vehicle's position. This is shown in Figure 1. Note that the road is assumed to be of any irregular curved shaped. The task of finding feasible vehicle positions inside the road is fairly simple in the road coordinate axis system. Further the road may have variable road widths at different points. Consider point P(x,y) in the cartesian coordinate system. The corresponding point P(x'y') in the road coordinate system is given by equation (1).

$$P(x,y) = P(x'y') = P\left(x', \frac{a}{w}\right). \tag{1}$$

Inter-conversion between the two schemes is important. Knowing all the coordinates of both boundaries, it is fairly easy to convert from the road coordinate system to the cartesian coordinate system. For the opposite conversion a small search technique is used which attempts to find the best match.

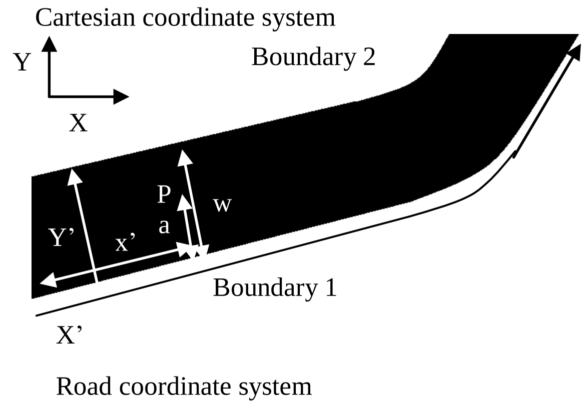

Figure 1. Cartesian and Road Axis System

B. Rapidly Exploring Random Trees

The algorithm used for the planning task is the method of Rapidly Exploring Random Trees [11, 12]. RRTs are branch and bound algorithms which attempt to find a path from source to goal. The algorithm has a tree-based representation for the sampled points used in computing the path. The search process starts with the source node as the root of the tree. Any random point is sampled in the solution space. The closest node in the tree is selected and is extended towards the sampled point by a constant called the *step size*. The extended node is added to the tree with the sampled node as parent if (a) it is feasible to travel from the parent to the extended node without any collision and (b) the extended node is not already in the tree. Nodes very close in the solution space are treated as being the same node.

In this problem the source is the point where a vehicle enters the segment being planned. The goal is any point in the segment end. The RRT planning algorithm attempts to connect the source to the goal by a suitable path. At each iteration the algorithm either results in the addition of a new node to the tree or no addition if the added node was infeasible. A maximum of *maxiter* iterations are used. If the planning algorithm still doesn't succeed in finding a solution, it is assumed that no feasible path exists as per the set criterion.

It is important that the vehicle initially moves in the current direction of orientation. In other words the path generated needs to start at an initial angle of heading that is equal to the vehicle's current angle. Hence the root of the tree (source) has just one child which is a point along the vehicle's length in its current heading direction.

RRT by default generate samples randomly, which leads to the tree to be expanded at any direction. This may take the entire algorithm a significant amount of time and it would eventually explore the entire road, till it reaches the road end or the goal. It is hence preferred to make the search process oriented towards the goal, and centric towards a narrower region, while still having some exploratory potential to search at distant sections of the road. This is controlled by sampling. We assume that the vehicle needs to travel smoothly, hence sampling is biased towards points which have the same value of Y' in the road coordinate system. A number of random samples are generated with probabilistic selection, the probability being proportional to deviation of the Y' coordinate of the generated sample from the current Y' coordinate of the vehicle's position. Once the probabilities are computed, a Roulette Wheel Selection scheme is used for generation of the sample. To ensure all samples are generated within the road, the road coordinate axis system is used. This also takes into account variable road widths.

C. Curve Generation

The path of the RRT, which is a collection of nodes, cannot be a straight line joining the nodes as the resultant path would not be smooth. Hence a curve smoothing technique is inbuilt in the algorithm framework. In this paper we use splines [13], which take a set of points which are used as control points and return a smooth curve. Every node addition is followed by the generation of curve. Firstly this ensures that the path formed by the addition of a node is feasible such that a vehicle lying at any point in the path does not collide with any obstacle. Secondly it means that at every point the curve is smooth enough to allow the vehicle to travel at the set speed and at no point does it have to reduce speed. Thirdly vehicles have to check for possible collisions with other vehicles and for this exact curve information is required.

We additionally check the feasibility of a curve by placing the vehicle at every point with its orientation as the immediate angle of the curve. The vehicle must not collide with any obstacle or other vehicle for which planned trajectories of other vehicles are queried. The speed of the vehicle at any point [14] is given by equation (2).

$$v = \min\left(\sqrt{\frac{\rho}{k}}, v\max\right). \quad (2)$$

Here ρ is a constant whose value depends on the friction between the vehicle and road, *vmax* is the maximum allowable speed of the vehicle. k is the curvature of the curve which may be approximately given by equation (3).

$$k = \left\| \tau(i+d) + \tau(t-d) - 2\tau(d) \right\|. \quad (3)$$

Here $\tau(i)$ is a point in the generated curve at a distance i from start, d is a small constant and ||.|| denotes the Euclidian norm. For the speed of the vehicle to be feasible v must be equal to *vmax*. Note that the higher speeds would require a smoother curve.

The generated curve is a set of points that the vehicle follows. Since there are multiple vehicles, each of them queries the others to ensure there is no collision. To reduce the computation we represent the travel plan as a hash map [15] which maps the vehicle position against travel time using time as the hashing function. This means any vehicle's position at a given time can be computed within unit time.

D. Coordination

A priority based approach [16] is used for vehicle coordination. This approach assumes that each vehicle has a priority attached to it. All vehicles are planned as per their priorities starting with the highest priority vehicle. A higher priority vehicle never considers possible collisions with lower priority vehicles. In this approach we take the priority of the vehicle from its time of emergence into a segment, with a vehicle earlier into the planning scenario having higher priority. This ensures that a vehicle already planned need not be re-planned as a new vehicle enters the segment.

The speed of a vehicle plays a major role in multi-vehicle planning and if an entering vehicle has a high speed with no room to overtake other vehicles a collision is unavoidable. Hence if the RRT planner of the vehicle fails to find a feasible travel plan, the speed of the vehicle is reduced by some amount Δ. Hence vehicle's speed is constantly reduced till the algorithm is able to generate a travel plan. In the worst case the speed of the vehicle is reduced to the speed of the slowest vehicles in the segment in which case it simply follows them. If no feasible path is generated, it is regarded as a blocked route, and the vehicle would require re-planning of its entire journey [17].

The general outline of the algorithm is given by algorithm 1 and 2.

Algorithm 1: Plan(vehicles, map)

```
while not end of simulation
    while true
        plan←RRT(current position of v, segment)
        if plan is not null
            represent plan as a hash map of time
            break while
        else v->speed ← v->speed – Δ
        end if
    end while
    move vehicles as per generated plan
end while
```

Algorithm 2: RRT(source, segment)

root ← source
$child_1$← point at distance *length* of Root at current angle
$child_1$->parent ← root
repeat for a maximum of *maxiter*
    Generate random samples and probabilistically select sample s

p ← node nearest to s in tree
n← node by extension of p in direction of s
if no point close to n lies in tree and vehicle placed at n with direction p to n is obstacle free
generate curve to n
check for collision with higher priority vehicles, static obstacles, and minimum travel speed
if generated curve is feasible
add n to tree with parent p
if n is close to segment end, return path
end if
end if
end repeat
return null

III. RESULTS

The discussed algorithm was developed and tested by means of simulations over an engine developed in MATLAB. The planning scenario is initiated by specifying the speeds, dimensions, entry times, entering orientations of all the vehicles. Different modules are made for RRT planning, curve generation, collision checking, etc. as discussed in section II.

A. *Single vehicle simulations*

The algorithm was tested on a variety of maps. In all the maps the vehicle was generated on the left side of the road segment and was supposed to travel to the other end of the road segment. We discuss here 3 simulations in detail. The first scenario consists of a curved road. Two simple obstacles are placed one after the other. The map and the path traced by the vehicle are shown in Figure 2(a). It can be seen that the vehicle was able to traverse to its goal in a fairly simple path. The path may not be so good at the very end, but as stated the vehicle enters the next segment before finishing its journey. Hence a re-planned path is effectively followed rather than merely the path planned solely in this segment.

The second scenario consists of a straight road with a complex grid of obstacles. The map and the path traced are shown in Figure 2(b). Although multiple paths were possible, the continuous iterative expansion resulted in a path reaching the segment end. It cannot be ascertained that the traversed path is optimal, however in such driving rapid decisions are more essential than spending high computation in ensuring the smallest path length.

The last scenario of study is again a curved road with variable sized obstacles. The characteristic placement of obstacles is such that avoiding an obstacle leaves the vehicle in a harder position to avoid the obstacles that lie ahead¸ still maintaining smoothness and high speed. The map and the path traced by the vehicle are shown in Figure 2(c). The vehicle succeeds in finding a path to reach the segment end. It is visible that in the middle the vehicle had to traverse large distances, this was due to a large step size which resulted in lower computational effort. Again it may be seen that excessive computation is not spent in trying to optimize the trajectory rather it is to generate a feasible trajectory.

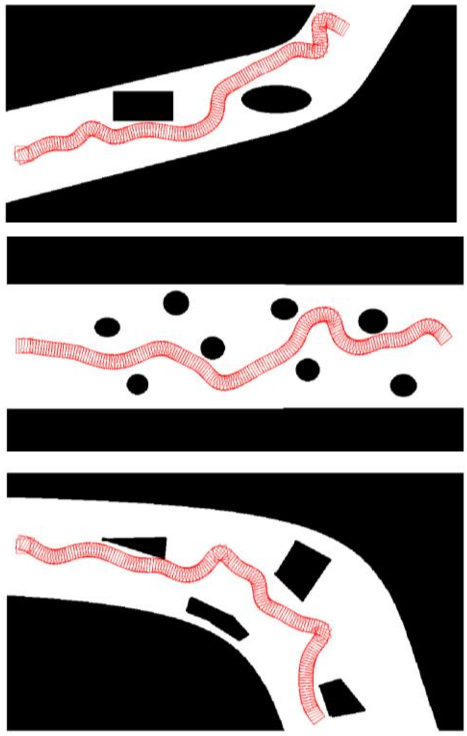

Figure 2. Path traced by vehicle in multiple scenarios

The basic planning algorithm is the RRT. For the second scenario the basic tree with various nodes connected by straight lines is shown in Figure 3(a). The tree produced by curve smoothing by splines is shown in Figure 3(b). The basic methodology of the algorithm can be seen in the figures. As per the design, the intent is to rapidly reach the goal from the source, rather than to explore the complete area. Hence there are areas where the search did not proceed at all. A large step size results in the goal being found after a small number of iterations. This results in a lower computational time for the algorithm. Further nodes very close to other nodes were not allowed thus giving the entire tree a simpler structure which further helps in making the computational time of the algorithm fairly small.

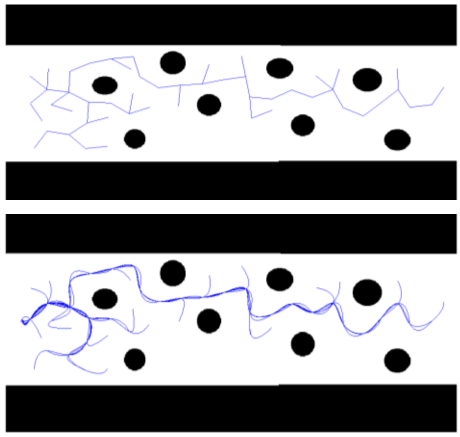

Figure 3. Tree generated in planning by RRT

B. *Multi vehicle simulations*

Single vehicle scenarios enable a good understanding of the manner in which the algorithm works as well as in testing the algorithm for both simple and complex scenarios. We further extend here experimentation to scenarios involving multiple vehicles. In these scenarios a vehicle enters the map, plans and traverses as per its plan. In the middle another vehicle is generated which also needs to plan so as to avoid collision either with the earlier vehicle or with static obstacles. We again discuss three scenarios in detail.

The first scenario consists of a curved road where two vehicles navigate. The first vehicle travels by almost the same path as shown in Figure 2(a). A second vehicle is generated which is capable of travelling at higher speed. However the second vehicle has no space to overtake the first vehicle and is forced to follow the first vehicle. The speed of the second vehicle drops to the speed of the first vehicle and the paths traced are similar. The scenario when the second vehicle enters is shown in Figure 4(a) and the scenario at a random time in the vehicle chase is shown in Figure 4(b). This constitutes vehicle following behavior which is commonly seen in everyday driving.

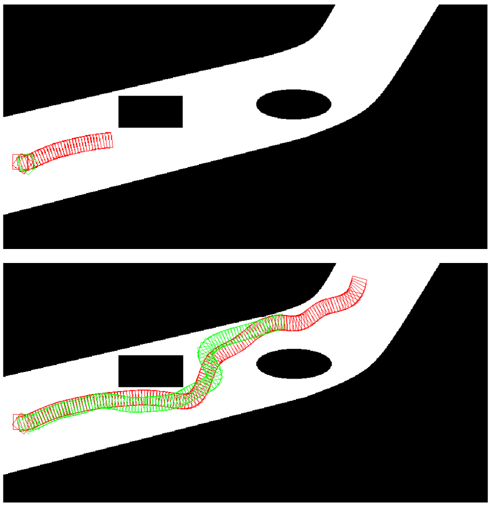

Figure 4. Vehicle following behavior exhibited by vehicles

The next scenario consists of a straight road. Again the second vehicle is generated after the first. In this scenario there was plenty of scope for the second vehicle to avoid the first vehicle and overtake it such that both vehicles traverse to the segment end. The scenario at the time of emergence of the second vehicle is shown in Figure 5(a). The scenario at a random time in the vehicle motion is given in Figure 5(b). This constitutes overtaking behavior of the vehicles which is again common especially when vehicles differ greatly in speeds.

In the last scenario the vehicles emerge simultaneously from either side of the road. Here initially the first vehicle plans and then the second vehicle plans. The second vehicle in its planning needs to account for the coming vehicle and must avoid it. The scenario at the time the two vehicles have just avoided each other is shown in Figure 6(a). The rest of the journey is given in Figure 6(b). The second vehicle had to align itself in a manner so as to avoid the first vehicle whose plan was already decided. The presence of obstacles made the task more difficult. The RRT planner still succeeded in generating a feasible trajectory which the second vehicle could follow. This constitutes the vehicle avoidance behavior of vehicles.

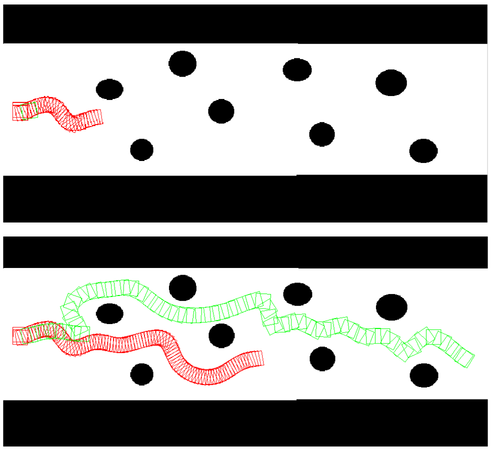

Figure 5. Overtaking behavior exhibited by vehicles

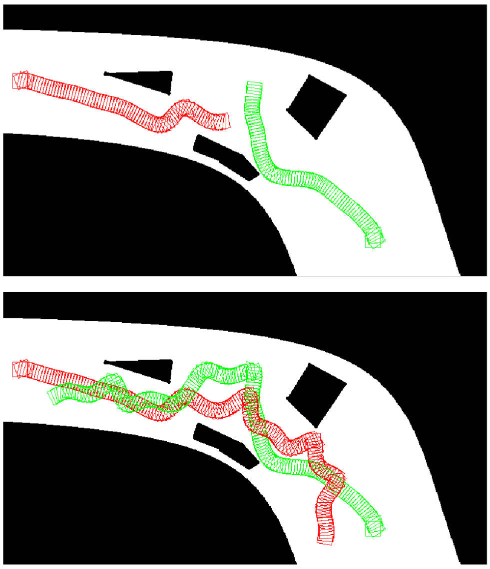

Figure 6. Vehicle avoidance behaviour exhibited by vehicles

We further study here the effect of *step size* used as a parameter of the RRT in the planning of the vehicles. The quality of solutions is judged by the path length and the time needed to generate the solution. Path length is of little importance, since any path constructed would have a path length almost equivalent to the road length. However the ability to generate a solution, if there exists one, is of high importance. Hence the purpose of the analysis is to judge the parameter to allow rapid generation of feasible results in complex scenarios. A large step size makes the algorithm increment by large steps towards the goal. As a result the total number of nodes in the RRT is low. Larger steps mean nodes are fairly wide apart and hence the total number of nodes in the tree is smaller. However it may take time to generate all these nodes, which requires more iterations of the algorithm. Hence a large step size does not mean a smaller computation time. The effect of change in step size is shown in Figure 7(a) for the total number of nodes in the RRT and Figure 7(b) for the total number of iterations used by the algorithm. The number of iterations for a small step size is high as the number of nodes generated in the tree is high. The number of iterations for a large step size is due to the inability to generate feasible nodes. By further increasing the step size the algorithm is unable to generate a feasible path as the turns cannot be modeled.

We further justify the use of RRT for problem solving as compared to the dozens of algorithms available in literature [18]. Graph search algorithms like A* are prominently used for problem solving. The performance of these algorithms depends upon the resolution to which the map has been decomposed to and the number of states to which each state of the algorithm connects to. Higher resolutions and more number of connected states give better paths, but algorithm may not complete in real time. For smaller values the vehicle may not be able to enter into narrow regions, or be able to generate a feasible trajectory with a coarser view of the map. Evolutionary algorithms fail if the resultant path is complex (in terms of number of turns in path), for which they require a high computation time. Both these approaches are useful for getting an optimal path length, which is not an important factor as per the problem modeling, however the computational time is. Behavioral approaches including neural networks are fuzzy systems do not ensure completeness, and vehicle is likely to get struck in scenarios. Further cooperation is too loosely modeled, and decisions to following a vehicle or to overtake it are difficult to make out.

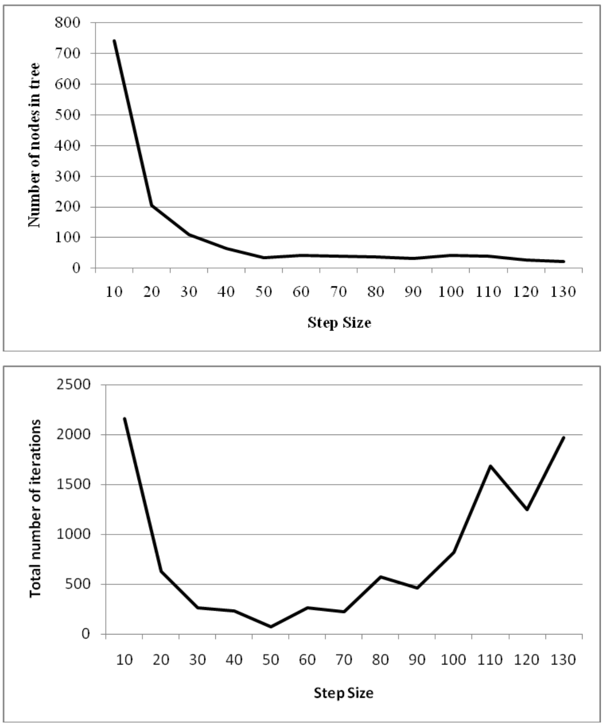

Figure 7. Analysis of the step size parameter of RRT

IV. CONCLUSIONS

In this paper we have presented an RRT based planning technique for the task of motion planning of multiple autonomous mobile vehicles. Planning was done in road segments. Prioritization was used as a means of coordination between vehicles where vehicles that entered the planning scenario earlier had a higher priority. The basic planning technique used a novel road coordinate system. An attempt was made to generate more samples so that vehicles occupy the same lane. Experimental results over a number of scenarios show that vehicles were able to navigate in fairly complex environments. Behaviors of vehicle following, overtaking, and vehicle avoidance were also displayed.

Though the present work is simulation only, experimental verification with real vehicles may be done. The algorithm also needs to have a provision for non-autonomous vehicles in traffic as well as vehicles without inter-vehicle communication.

ACKNOWLEDGMENT

The authors wish to thank the Commonwealth Scholarship Commission in the United Kingdom and the British Council for their support of the first named author through the Commonwealth Scholarship and Fellowship Program - 2010 - UK award number INCS-2010-161.

REFERENCES

[1] M. Buehler, K. Iagnemma, S. Singh, *The 2005 DARPA grand challenge: the great robot race*, Berlin, Heidelberg: Springer-Verlag, 2007.

[2] G. Seetharaman, A. Lakhotia, E. P. Blasch, E.P, “Unmanned Vehicles Come of Age: The DARPA Grand Challenge,” *Computer*, vol. 39, no. 12, pp 26-29, 2006.

[3] C. Crane et al., “Team Gator Nation's Autonomous Vehicle Development for the 2007 DARPA Urban Challenge”, *J. Aerosp. Comput. Inform. Commun.*, vol. 4, no. 12, pp 1059-1085, 2007.

[4] S. Tsugawa, “Inter-vehicle communications and their applications to intelligent vehicles: an overview,” *Proceedings of the IEEE Intelligent Vehicle Symposium Vol. 2*, 2002, pp. 564- 569.

[5] C. Hatipoglu, U. Ozguner, K. A. Redmill, “Automated lane change controller design,” *IEEE Trans. Intel, Transport. Syst.*, vol.4, no.1, pp. 13- 22, 2003.

[6] J. Sewall, J. van den Berg, M. C. Lin, D. Manocha, D, “Virtualized Traffic: Reconstructing Traffic Flows from Discrete Spatiotemporal Data”, *IEEE Trans. Visualization Computer Graphics*, vol. 17, no. 1, pp 26-37, 2011.

[7] M. Montemerlo et al., “Junior: The Stanford entry in the Urban Challenge”, *J. Field Robotics*, Vol. 25, No. 9, pp 569–597, 2008.

[8] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. P. How, G. Fiore, “Real-Time Motion Planning With Applications to Autonomous Urban Driving,” *IEEE Trans. Control Syst. Technol.*, vol.17, no.5, pp.1105-1118, 2009.

[9] J. Leonard et al., “A perception driven autonomous urban vehicle,” *J. Field Robotics*, vol. 25, no. 10, pp. 727–774, 2008.

[10] J. Leonard et al., A Perception-Driven Autonomous Urban Vehicle, *Springer Tracts in Advanced Robotics Vol. 56*, Berlin-Heidelberg, Springer-Verlag, 2009, pp 163-230.

[11] J. J. Kuffner, S. M. LaValle, “RRT-connect: An efficient approach to single-query path planning ,” *Proceedings of the IEEE International Conference on Robotics and Automation vol.2*, 2000, pp.995-1001.

[12] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999, pp. 473–479.

[13] C. de Boor, *A Practical Guide to Splines*, Berlin-Heidelberg: Springer Verlag, 1978.

[14] E. K. Xidias, P. N. Azariadis, "Mission design for a group of autonomous guided vehicles", *Rob. Autonom. Syst.*, vol. 59, no. 1, pp 34-43, 2011

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, "Chapter 11: Hash Tables", *Introduction to Algorithms, Second Edition*, MA: MIT Press, 2001, pp 221-245.

[16] M. Bennewitz, W. Burgard, S. Thrun, "Optimizing schedules for prioritized path planning of multi-robot systems", *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2001, pp 271 – 276.

[17] C. Urmson, C. Baker, J. M. Dolan, P. Rybski, B. Salesky, W. L. Whittaker, D. Ferguson, M. Darms, "Autonomous Driving in Traffic: Boss and the Urban Challenge", *AI Magazine*, vol. 30, no. 2, pp 17-29, 2009.

[18] R. Kala, A. Shukla, R. Tiwari, S. Roongta, R. R. Janghel, "Mobile Robot Navigation Control in Moving Obstacle Environment using Genetic Algorithm, Artificial Neural Networks and A* Algorithm", *Proceedings of the IEEE World Congress on Computer Science and Information Engineering*, Los Angeles/Anaheim, USA, pp 705-713.