Planning Autonomous Vehicles in the Absence of Speed Lanes using an Elastic Strip

Rahul Kala and Kevin Warwick

Citation: R. Kala, K. Warwick (2013) Planning Autonomous Vehicles in the Absence of Speed Lanes using an Elastic Strip. *IEEE Transactions on Intelligent Transportation Systems*, 14(4): 1743-1752.

Full Text Available At: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6549134

***Abstract*—Planning of autonomous vehicles in the absence of speed lanes is a little researched problem. Yet it is an important step towards extending the possibility of autonomous vehicles to countries where speed lanes are not followed. The advantages of having non-lane oriented traffic include larger traffic bandwidth and more overtaking, features which are highlighted when vehicles vary in terms of speed and size. In the most general case the road would be filled with a complex grid of static obstacles and vehicles of varying speeds. The optimal travel plan consists of a set of maneuvers which enable a vehicle to avoid obstacles, overtake vehicles in an optimal manner and in turn enable other vehicles to overtake. Desired characteristics of such a planning scenario include near-completeness and near-optimality in real-time with an unstructured environment; with vehicles essentially displaying a high degree of cooperation and enabling every possible (safe) overtaking procedure to be completed as soon as possible. Challenges addressed in this paper include a (fast) method for initial path generation using an elastic strip, (re-) defining the notion of completeness specific to the problem, and inducing the notion of cooperation in the elastic strip. Using this approach, vehicular behaviors of overtaking, cooperation, vehicle following, obstacle avoidance, etc. are demonstrated.**

***Index Terms*—Cooperative Systems, Intelligent Vehicles, Motion Analysis, Multirobot Systems**

I. Introduction

The problem of planning autonomous vehicles deals with all aspects of decision making which include selecting the route to reach the goal [1], selecting the manner of avoiding obstacles and other vehicles [2], trajectory generation for motion [3], determining the lane of travel [4], dealing with special incidents and blockages [5], etc. Planning may be different for scenarios of straight roads, junctions, intersections, diversions, etc [6, 7]. The specific problem tackled in this paper is planning for straight roads.

In general navigation, in most countries, consists of vehicles organized laterally within speed lanes. The advantages of organized traffic are a higher degree of safety, clearer intentions of other vehicles and fewer lane changes or lateral movements which signify a more comfortable driving experience and hence shorter travel distances and times.

A number of countries however allow unorganized traffic where vehicles may place themselves laterally anywhere in the road. In such a scenario the planner can construct trajectories keeping the vehicle anywhere inside the road boundaries, whilst varying the speed of travel. The advantages of such a traffic system include higher traffic bandwidth and more overtaking (with each overtake signifying better travel for the vehicle which completed the manoeuver). However, unorganized traffic has an apparent higher risk of accidents due to uncertain movements of other vehicles. Consider Indian traffic as an example. Bicycles, motor bikes, 3-wheeled auto rickshaws, small cars, buses, overloaded trucks, hand-pulled vehicles etc., all share the same road. Details of the dynamics of such traffic can be found in [8, 9].

This paper looks into the problem of the navigation of autonomous vehicle for non lane-based traffic. The motivation is to move autonomous vehicles towards present day unorganized traffic. Further, an increase in the number of autonomous vehicles may bring increased diversity to the currently organized traffic landscape which may continually diminish the use of speed lanes, up to the point when the entire traffic becomes unorganized.

Completeness is an essential requirement in the choice of algorithm for solving any problem. This refers to the guarantee of the algorithm in finding a solution, if a solution exists. However it is usually possible to make reasonable assumptions to enable an algorithm to compute the results in a limited time. Such algorithms are referred to as near-complete as they ‘almost guarantee’ the generation of a result, as long as the assumptions hold. Similarly optimality is another essential requirement which refers to the guarantee of the algorithm to return the best solution, if one exists. Algorithms of this type in which reasonable assumptions or approximations must be made are referred to as near-optimal.

Elastic Strips [10] have been used for the planning of a mobile robot. A number of homotopies may lead the robot from its source to the goal. The preferable homotopy is selected and represented as a strip which marks the robot’s trajectory. For computational reasons, the strip is discretized to a number of waypoints. A change in the environment is

Manuscript received Month Date, 2012. The first named author was supported by Commonwealth Scholarship and Fellowship Plan (2010) – UK through award number INCS-2010-161.

The authors are with the School of Systems Engineering, University of Reading, Whiteknights, Reading, Berkshire, RG6 6AY, UK (phone: +44-7424752843; e-mails: rkala001@gmail.com, k.warwick@reading.ac.uk).

marked by movement of these points, the addition of points or deletion of points such that the resulting trajectory is collision-free. The intent is to have the least waypoints mostly near the obstacles. Each waypoint is acted on by a repulsive force from the obstacles which makes the trajectory lie far from the obstacles. There is an additional internal force which pulls the waypoints towards each other, resulting in a shortening of the path. Given that two obstacles do not eventually intersect such that the robot is planned to travel in between them, the resulting travel plan is complete and reactive to real time changes.

Given an initial optimal travel plan, the algorithm is near-optimal to small changes in the environment. With some additional computational time, the algorithm is, as a result, better than the Artificial Potential Field method [11, 12] and similar approaches which are neither complete nor optimal. The algorithm is in fact similar to the Elastic Roadmap [13, 14] and related approaches which maintain a roadmap in a dynamic environment.

A preliminary version of this paper was presented at the 2012 Intelligent Vehicles Symposium [15]. The paper used lateral potentials for planning movement of a vehicle. This paper tackles two major limitations of that work. (i) Potential methods are neither near-complete nor near-optimal. This paper solves both the problems. (ii) The paper answers the questions of deciding on a strategy of avoiding obstacles: this was done earlier using heuristics which could be problematic in many situations.

The key contributions of the paper are: (i) Design of a method to quickly compute the optimal strategy for obstacle avoidance, and the resulting trajectory. (ii) Real-time optimization of the trajectory as the vehicle moves, making the resulting plan near-optimal. (iii) Using heuristics to ensure the travel plan is near-complete. (iv) Design of the algorithm to allow cooperation between vehicles. (v) Enabling vehicles to travel at near-optimal speeds. (vi) Displaying complex behaviors of overtaking, vehicle following, obstacle avoidance, cooperation, etc. in a multi-vehicle scenario.

II. Related Research

Most research in this area is based on organized or a lane-oriented traffic. We list here some notable research on unorganized traffic or which is extendable to unorganized traffic. Kuwata et al. [16] used Rapidly-exploring Random Trees (RRT) with a biased sampling technique for the planning of a single vehicle. The algorithm however lacks global optimality. Treating vehicles and obstacles alike can lead to loss of completeness. Anderson et al. [17] solved the problem of trajectory generation for a single vehicle using constrained Delaunay triangles for a structured environment. Chu et al. [18] constructed a number of candidate paths from which the best path was selected. The strategy can be used for avoiding obstacles by a single maneuver only. Other limitations of these algorithms include non-cooperative vehicles and the possibility of a vehicle steer causing a collision with another vehicle to the rear in a diverse vehicle speed scenario. In lane based travel this is equivalent to the fact that a (slower) vehicle may suddenly change a lane and push in front of a (faster) vehicle, thus requiring sudden braking.

Kala and Warwick [19] designed a multi-level graph search algorithm for the planning of multiple vehicles based on the assumption that vehicles were connected via a communication framework. Layers corresponded to route planning, obstacle avoidance strategy computation, vehicle coordination (placement) and trajectory generation. The resulting algorithm was though not scalable. The application of Elastic Bands for the problem can be found in [20] where it was used to model vehicle following behaviour. The band was attached to the vehicle being followed. The approach is clearly not extendable to overtaking. Moreover the strategy used for obstacle avoidance for the leading vehicle may not be the same for the following vehicle.

While the notion of cooperation seems to be a challenge for non-lane based navigation algorithms, for lane-based traffic a comparative study for cooperative overtaking was shown by Frese and Beyerer [21]. The authors studied mixed integer programming, tree search, elastic bands, random priorities and optimized priorities. A direct implementation in non-lane based environments can though be computationally expensive. Kala and Warwick [22] designed a set of discrete behaviours for the cooperative navigation of a vehicle in unorganized traffic. The discrete nature of the definitions was a limitation, as the transitions could sometimes not be smooth.

For planning in a lane based system automata techniques are widely used. Furda and Vlacic [23] modelled the problem using deterministic state automata with multi-criterion decision making. Schubert et al. [24] used lane markings and distances from vehicles as inputs to decide the optimal lane of travel.

Interesting similar algorithms exist in the domain of planning for mobile robots. Baxter et al. [25] used the Artificial Potential Field method for planning multiple robots. The authors enabled robots to share potentials to rectify environment perceptions. Gayle et al. [26] modeled the cooperation by social potential, where different types of robots applied different potentials. The social interaction framework was displayed within an adaptive roadmap in [27]. Such an approach is cooperative, near-real-time, near-optimal, and near-complete (with the exception of robots mutually causing a deadlock or congestion). These and similar reactive approaches are not applicable for a narrow road like structure where the robot may be found oscillating within the road, not allowing possible overtaking (where two robots are symmetrically ahead and behind each other) or poorly allowing possible overtaking (where two robots are almost ahead and behind each other). Further, unlike mobile robots, any turn or lateral movement (lane change) in a vehicle scenario threatens a collision with a vehicle to the rear.

Based on these works it is clear that generalized planning algorithms for vehicles are non-cooperative and general lane-based approaches cannot be applied for non-lane based travel.

The presence of a narrowly bounded road structure, overtaking and vehicle following as the primary underlying dynamics and unknown time and place of emergence of vehicles in a continuous traffic scenario makes the problem of autonomous vehicle planning fundamentally different from multi-robot path planning. Any algorithm design needs to be validated against these differences, and may possibly use this knowledge about the operational scenario as heuristics for computational speedup. Due to this heuristic the algorithm presented here also performs better than any probabilistic sampling method which may explore much more; or a graph search approach which may be computationally expensive for high resolution maps, while (due to the nature of the problem) resolution across the lateral axis cannot be reduced.

III. Problem Definition

A limited map of the road is assumed to be given. It is considered that the road does not contain any junctions or diversions. The road is characterized by its left and right boundaries. It is assumed that all obstacles (including vehicles) can be sensed with some degree of certainty. Let any such general vehicle R_i be located at position L_i (x_i', y_i', θ_i'), where θ_i' denotes the heading direction and (x_i', y_i') correspond to the center; and travelling with speed v_i. Let at any general time the vehicle being planned (Q) be at position L_q (x_q', y_q', θ_q') with linear speed v_q ($\leq vpref_q$) and angular speed ω_q ($\leq \omega_q^{max}$), where $vpref_q$ is the preferred (maximum) speed of travel and ω_q^{max} is the maximum angular speed. The linear acceleration is bounded by acc_q^{max}. All positions are denoted using the longitudinal (X') and lateral (Y') axis system [15]. Let the free workspace be given by ζ_{free}^{static} which excludes any region with static obstacles or outside the road boundaries/road segment. Static obstacles and moving vehicles behave differently and are handled separately. Broken down vehicles cannot move, hence these are taken as static obstacles.

A. Objectives

The purpose of the algorithm is to construct a travel plan τ. Let $\tau(t)$ {=(x_q', y_q', θ_q')} be the planned position at time t and T denote the time up to which the vehicle is planned. The objectives in (non-strictly) decreasing order of importance are:

(a) vehicle should go as far as possible, or maximize $\tau(T)[X']$, the longitudinal position at the end of travel,

(b) maximize the minimum lateral clearance of the trajectory, where lateral clearance is the minimum distance of the vehicle from any obstacle measured in the lateral axis.

(c) minimize T,

(d) maximize lateral cooperation, the net lateral movement (measured in lateral axis) of a vehicle traversed solely with the intention of enabling some other vehicle to the rear to obtain a better plan.

(e) In case overtaking an obstacle may be equally advantageous (as per the above mentioned objectives) from both the left and right side, the strategy used in [15] (section IV) is regarded as better.

B. General Speed Bounds

Consider a point sized object at state s moving with speed v_q ($\leq vpref_q$). Consider that an obstacle i (in static obstacles and a pool of vehicles R) lies ahead of it at a longitudinal distance d_f. The maximum speed (v_a^s) by which the object can move so as to avoid a collision with i is given by (1). v_a^s should be low enough to allow the object to stop before a collision takes place, if there is no other alternative than following i; while it is assumed that i continues to travel with its current speed (v_i if vehicle, 0 if obstacle). This corresponds to slowing with the maximum uniform retardation of $agg_q.acc_q^{max}$.

$$v_a^s = \min\left(v_i + \sqrt{2.acc_q^{\max}.agg_q.d_f}, vpref_q\right) \tag{1}$$

Here agg_q ($0<agg_q \leq 1$) is the aggression factor which limits the planned acceleration. Lower values would indicate a more comfortable drive, while higher values sacrifice comfort for travel time. A minimum threshold distance of d_{unc} must always be maintained which is excluded from the measured d_f. This is employed to overcome uncertain speed changes of the vehicle in front or other uncertain environment changes.

Further consider that an obstacle i (in static obstacles and a pool of vehicles R) lies behind a point sized object at a longitudinal distance of d_b. The minimum speed (v_b^s) by which the object can move so as to avoid collision with i is given by (2). Using the concepts from (1), v_b^s should be high enough to allow stopping i traveling with speed v_i (0 if obstacle) before it collides with the object, if i follows the object.

$$v_b^s = \max\left(v_i - \sqrt{2.acc_i^{\max}.agg_i.d_b}, 0\right) \tag{2}$$

Hence for safe travel, $v_b^s \leq v_q \leq v_a^s$

C. Plan Feasibility

A driver only considers vehicles ahead of it while formulating his/her travel plan, as is the case in deciding the feasibility of a travel plan τ. The only exception is making a turn (lane change) when one might accidently drive in front of a vehicle which may not have enough time to slow down to avoid collision. Hence the resulting pool of vehicles considered for feasibility consists of all vehicles R_i which either lie completely ahead of Q (or whose longitudinal coverage is completely ahead of the longitudinal coverage of Q), or do not lie in the same lane as Q (or whose lateral coverage is completely disjoint from the lateral coverage of Q). The set of vehicles is given by (3).

$$R = \left\{ \begin{aligned} & R_i : L_i \otimes R_i[X'] > L_q \otimes Q[X'] \\ & \vee L_i \otimes R_i[Y'] \cap L_q \otimes Q[Y'] = \phi \end{aligned} \right\} \tag{3}$$

A plan τ is called a feasible travel plan if

(a) No collisions occur with static obstacles or road boundaries, or (4)

$$\tau(t) \otimes Q \in \zeta_{free}^{static} \forall t \leq T \quad (4)$$

(b) No collisions occur with the projected motion of other vehicles, or (5)

$$\tau(t) \otimes Q \notin \bigcup_{i, R_i \in R} \left(\left(L_i \pm \Delta L_i \right) + t\left(v_i \pm \Delta v_i \right) \right) \otimes R_i, \forall t \leq T \quad (5)$$

All vehicles are projected to travel straight (longitudinally), maintaining their current orientation, from the sensed initial position in the range (L_i- ΔL_i, L_i+ ΔL_i) with a constant sensed speed in the range (v_i- Δv_i, v_i+ Δv_i). A collision is said to have occurred if Q intersects with any projected position of the vehicle. Here ΔL_i and Δv_i denote the uncertainty in measurements of position and speed of the vehicle R_i.

(c) At all times the speeds are within the desirable bounds, or (6)

$$v_b^s \leq v_q \leq v_a^s \forall s \in \tau(t) \otimes Q, t \leq T \quad (6)$$

Checking speed bounds ensures that the trajectory being followed can be terminated at any instance (due to changed environment dynamics), and the vehicle can be made to drive straight ahead, giving enough time for every vehicle to adjust.

IV. Planning using Lateral Potentials

This section summarizes the approach used in [15]. The algorithm used lateral potential to compute the desirable angle to longitudinal axis, based on which a steering attempt was made (say $LP(s)$ for a state s). The potential is computed from 4 sources, which are the forward potential, side potential (left side and right side), diagonal potential (forward left and forward right), and back potential. The various sources are illustrated in Figure 1.

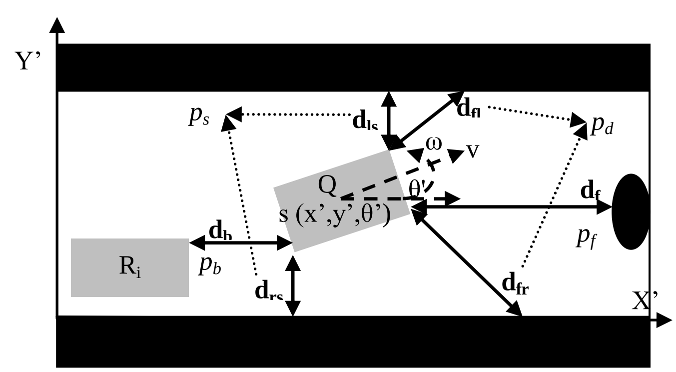

Fig. 1. Sources of potential. d_f, d_{ls}, d_{rs}, d_{fl}, d_{fr}, and d_b denote distance from ahead, left, right, forward-left, forward-right, and back obstacles respectively. p_f, p_s, p_d, and p_b denote forward, side, diagonal, and back potentials respectively.

Forward potential at a point s is a reaction to seeing an obstacle i longitudinally ahead (along the length of the road, irrespective of vehicle orientation) at a distance of d_f. In [15] a simple heuristic was used for deciding on the direction. In the case of a static obstacle the strategy was to turn right if the obstacle was more towards the left of the road and vice versa. In the case of a vehicle, the direction was right if the other vehicle was more laterally towards the left and vice versa. This heuristic is though neither complete nor optimal. In the current approach, let τ_{strat} be a book-keeping variable which directly returns the direction of turn $\tau_{strat}(i)$ for obstacle i. Time to collision, instead of the commonly used distance, was used as a metric; while the potential was taken to be inversely proportional to the square of the metric (time to collision), given by (7).

$$p_f = \begin{cases} 0 & v_i \geq vpref_q \\ \tau_{strat}(i) \left(\dfrac{vpref_q - v_i}{d_f} \right)^2 & v_i < vpref_q \end{cases} \quad (7)$$

The side potential is computed using the free lateral distance on the left side (d_{ls}) and right side (d_{rs}). Each side applies potential in the opposite direction given by (8).

$$p_s = p_{ls} + p_{rs} = -\max\{(1/d_{ls})\}^2 + \max\{(1/d_{rs})\}^2 \quad (8)$$

The diagonal potential has magnitude indicated by the diagonal free distances of forward left (d_{fl}) and forward right (d_{fr}), while the directions are opposite to the side considered given by (9).

$$p_d = p_{fl} + p_{fr} = -(1/d_{fl})^2 + (1/d_{fr})^2 \quad (9)$$

The back potential is responsible for the cooperative behavior of the vehicle. In [15] the direction was given by the heuristic whether the vehicle behind was more towards the left or right. Here the same heuristic is used, however cooperation is only applied when the direction of overtaking is clear. Similar to (7), the magnitude is indicated by the time to collision with the vehicle i to the rear (if any) given by (10). s[Y'] denotes the Y' component of state s.

$$p_b = \begin{cases} \left(\dfrac{v_i - vpref_q}{d_b} \right)^2 & v_i > vpref_q, L_i[Y'] < s[Y'] \\ -\left(\dfrac{v_i - vpref_q}{d_b} \right)^2 & v_i > vpref_q, L_i[Y'] > s[Y'] \\ 0 & otherwise \end{cases} \quad (10)$$

The potential directions are based on the X'Y' coordinate axis system and are independent of vehicle orientation. Each potential is measured across all possible points, and the maximum potential recorded is used for computation of the resulting potential given by (11). Figure 1 shows the sources corresponding to the maximum potential values.

$$LP = sen_{X'}.p_f + sen_{Y'}.p_s + sen_{X'Y'}.p_d + coop.p_b \tag{11}$$

where $sen_{X'}$, $sen_{X'Y'}$, and $sen_{Y'}$ denote the various sensitivities and *coop* denotes the cooperation factor.

V. The Algorithm

Let the trajectory being followed at any time be τ. The algorithm additionally defines τ_{obs} to denote the trajectory constructed by considering only static obstacles. This term is only defined if the vehicle cannot compute a collision-free trajectory which assures collision-free avoidance of static obstacles. τ_{strat} denotes the operational strategy and is a specification of direction (*left* or *right*) by which any obstacle needs to be overcome. Initially τ contains the immediate position only while τ_{obs} and τ_{strat} are both null.

A. Plan Extender

Consider a travel plan τ (known to be feasible as per the current traffic scenario) constructed using a strategy τ_{strat}. The task is to extend τ. The assumption is that the entire extended plan thereafter would be followed at a pre-specified speed v_q. The extension is carried using *LP* (section IV) with the difference that:

(i) The speed of travel is kept constant. The sampling frequency is taken to be inversely proportional to the speed indicated by *LP*. The sampling time indicates the timespan after which the steering action of *LP* is applied. When the (projected) vehicle is nearer to obstacles, the speed indicated by *LP* is smaller and hence frequent *LP* actions are applied, and vice versa. Every call to *LP* consumes computation. This methodology is hence adaptive to obstacle placements.

(ii) The strategy parameters appearing in section IV are looked up in τ_{strat}. For every new obstacle witnessed (which does not have an entry in τ_{strat}) all combinations of strategies (e.g. Figure 2) are separately computed and the best one as per the performance criterion set in section III is chosen.

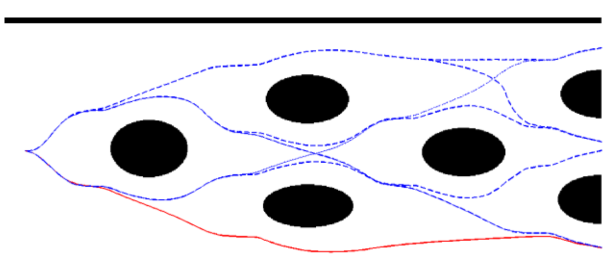

Fig. 2. Plan extension by every strategy. Blue (dashed) lines represent all possible strategy plans (only strategies experienced by the vehicle). Red (continuous) line shows the optimal plan.

At every step the extended plan needs to be feasible as per the criterion stated in section III. The plan extension may terminate for any one of three reasons: (a) End of road segment being planned is reached, (b) Not possible to move any further due to feasibility criterion due to a static obstacle, and (c) Not possible to move any further due to the feasibility criterion of another vehicle.

The algorithm can be regarded as complete for the cases which end in (a). In the case of (c) the algorithm has an option to travel as per the computed plan, and then to start following the vehicle in front. Human drivers can be reliably followed, while following an automated vehicle can be regarded as near-complete if the vehicle ahead is following a near-complete algorithm. Since the recurrence is applied to a forward vehicle, the planning algorithm due to termination at (c) can be regarded as complete.

In the case of termination due to (b), there needs to be an assurance of avoiding the static obstacle at a later time. Hence an attempt is made to compute a plan τ_{obs} considering only static obstacles and all ways of avoiding them. Out of the range of different possibilities the best plan is considered. While completeness in construction of τ_{obs} cannot be guaranteed, all possible ways of avoiding static obstacles are considered and no other vehicle is considered, the algorithm may therefore be stated as near-complete. It may be assumed that the vehicles ahead would eventually clear making way for the vehicle to follow the plan τ_{obs}. In lieu of these points the entire algorithm is near-complete and (combined with section V-B) near- optimal.

Algorithm 1: Extend1(τ, τ_{strat},v_q)

while τ is feasible
  $\tau \leftarrow \tau \cup LP(\tau[end], \tau_{strat}, v_q)$
  if τ[end] encounters an obstacle i and $\tau_{strat}(i)$=null
    $<\tau^a, \tau^a_{strat}> \leftarrow$ Extend1(τ, $\tau_{strat} \cup$ <i, left>,v_q)
    $<\tau^b, \tau^b_{strat}> \leftarrow$ Extend1(τ, $\tau_{strat} \cup$ <i, right>,v_q)
    $<\tau, \tau_{strat}> \leftarrow$ better of $<\tau^a, \tau^a_{strat}>$ and $<\tau^b, \tau^b_{strat}>$
    break while
  end if
end while
return $<\tau, \tau_{strat}>$

Algorithm 2:Extend(τ, τ_{strat}, v_q)

$<\tau, \tau_{strat}> \leftarrow$ Extend1(τ, τ_{strat}) with all obstacles
if τ ends with static obstacle
  $<\tau_{obs}, \tau_{stratObs}> \leftarrow$ Extend1(τ, τ_{strat}) with only static obstacles
  $<\tau, \tau_{strat}> \leftarrow$ Extend1(τ, $\tau_{stratObs}$) with all obstacles
end if
return $<\tau, \tau_{strat}, \tau_{obs}>$

The extension is hence given by Algorithm 1 and Algorithm 2. Algorithm 1 extends a plan consisting of τ and τ_{strat}. Every time a new obstacle is discovered, both avoidance from the left and avoidance from the right are tried, and the best is retained. For obstacles encountered previously, the strategy indicated by

τ_{strat} is used. Algorithm 2 computes τ_{obs}, in case τ as computed by Algorithm 1 ends in a static obstacle. $\tau_{stratObs}$ denotes the strategy to avoid static obstacles. *Extend1* is called again to make τ and τ_{strat} consistent with the strategy used in τ_{obs} that is $\tau_{straObs}$. In implementation all strategies may be stored in line 1 of *Extend*() and fetched in line 4 instead of a new function call.

B. Plan Optimizer

Consider a travel plan τ which, as per the current traffic scenario, needs to be followed with a pre-specified travel speed of v_q. The travel plan is optimized as the vehicle moves (and scenarios change). The plan τ is converted into a set of coarsely located waypoints (τ') vaguely representing the plan τ. The optimization of τ' is based on the analogy of a spring with each waypoint τ'_i representing a virtual vehicle with a movable clamp. τ'_i is modeled as a clamp attached to the lateral axis (Y'). Hence by the application of forces τ'_i can move along Y', but not along X'. The initial position $\tau`_0$ is fixed. This constraint disallows two waypoints to come close to each other which may slow down the optimization process. τ'_i is influenced by four forces, which are:

(i) *Lateral Force:* The force is applied by obstacles laterally left and laterally right in opposing directions, and the magnitude (F_l) is given by (8).

(ii) *Spring Extension Force:* Each way point τ'_i is attracted by the waypoint ahead τ'_{i+1} (if any) and behind τ'_{i-1} (if any) with a force proportional to the extension given by (12). Two points clamped to their lateral axis can have a minimal separation equal to their longitudinal separation. Any separation in excess is considered as an extension.

$$F_s = \left(\left\|\tau'_{i+1}-\tau'_i\right\| - \left(\tau'_{i+1}[X'] - \tau'_i[X']\right)\right).u\left(\tau'_{i+1}-\tau'_i\right) + \left(\left\|\tau'_{i-1}-\tau'_i\right\| - \left(\tau'_i[X'] - \tau'_{i-1}[X']\right)\right).u\left(\tau'_{i-1}-\tau'_i\right) \quad (12)$$

Here $u(x)$ denotes the unit vector in the direction of x.

(iii) *Cooperation Force:* A plan τ may initially be made only considering the vehicles in the scenario. Additional vehicles may appear later at the rear, and they might then aim to overtake. Extension of τ does not account for cooperation in overtaking, and hence the same is modeled in optimization. The force (F_{coop}) is the same as is given by (10).

(iv) τ_{obs} *Drift Force:* A non null value of τ_{obs} indicates that the current plan τ cannot overcome a static obstacle, while the plan τ_{obs} can overcome the static obstacle subjected to the absence of the other vehicles. Following τ would mean ending up close to the static obstacle and then having to steeply steer to avoid it, whenever feasible. Following τ_{obs} by waiting for the vehicles to clear and at every step computing the highest possible speed may mean excessive slowing down initially or for a large part of the journey. An attempt is made to induce advantages of both the techniques by following τ, but slowly drifting it towards τ_{obs}. The force (F_o) is proportional to the distance between the closest waypoint in τ_{obs} (say τ^i_{obs}) applied in same direction. This is given by (13).

$$F_o = \left\|\tau^i_{obs} - \tau'_i\right\|.u\left(\tau^i_{obs} - \tau'_i\right) \quad (13)$$

The total force is given by (14).

$$F_{total} = k_l.F_l + k_s.F_s + k_{coop}.F_{coop} + k_o.F_o \quad (14)$$

Here k_l, k_s, k_{coop} and k_o are the associated weights of the different factors.

The lateral component of F_{total} is used for deviating τ'_i. Only changes resulting in a feasible plan are admitted. For a sample path, the optimization is shown in Figure 3.

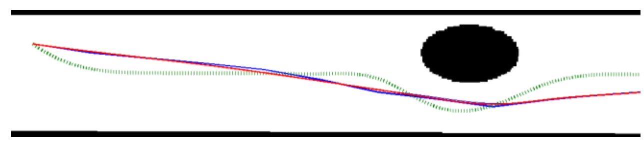

Fig. 3. Optimization of the plan. The blue line represents the initial plan, which after optimization is given by the red line. If optimized with the aim of maximizing average clearance, the plan is given by the green (dashed) line.

C. Complete Framework

The basic hypothesis behind the algorithm is simple. Compute the highest speed (v_q) which the vehicle can have as per the current scenario (step *A*); use v_q to trim (step *B*), extend/construct (step *C*) the plan such that the resulting plan is feasible; and optimize the plan (step *D*). The vehicle, at any time, may have two modes of operation which are Mode I: travelling with a plan ending at some static obstacle (τ_{obs}=null); and Mode II: traveling with a plan not ending at some static obstacle ($\tau_{obs}\neq$null). Each of the steps is applied for both modes (denoted *I A*, *I B* … *II D*), while switching between modes (denoted *I*→*II* and *II*→*I*) is monitored. These steps are applied as the vehicle moves and the scenarios change. Hence at any time the vehicle may be seen to show initial signs of reacting to any new obstacle (or an obstacle whose motion has changed a lot as per expectation), deliberating over later course of actions, adapting the plans to any changes in the scenario and optimizing any previously sub-optimal plan. The algorithm is summarized by Figure 4 and Algorithm 3.

We first discuss each of the 4 steps (*A* to *D*) for each mode of operation (*I* and *II*), and then discuss how these assemble into Algorithm 3. Step *A* is different for both modes *I* and *II*. In *II A*, the speed v_q is the highest speed possible for the vehicle as per the feasibility plan, taking into account the acceleration limits, aggression factor and deceleration limits. This update rule is formulated in realization of the fact that a fast moving vehicle at a distance, on being unable to overtake a slow moving vehicle in front, would be found to exhibit high speed up to the point when it comes close to the slower vehicle, post which a non-aggressive (as pre-specified) deceleration takes place. Hence at every step the vehicle displays maximum speed, which naturally plays a role in decreasing the total travel time.

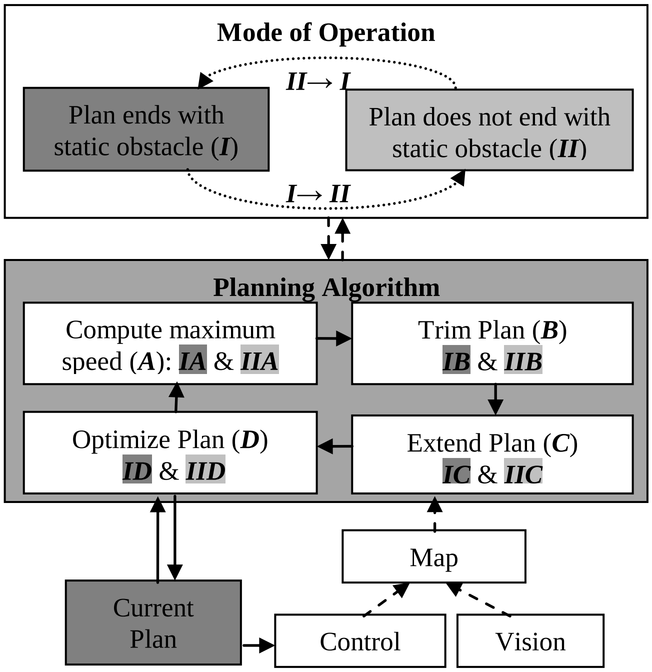

Fig. 4. General Algorithm Framework.

Algorithm 3: Plan(τ_{obs}, τ, v_q)

I B	if (I) and τ_{obs} is not collision-free τ_{obs}=null, τ=null
II A	if (II), compute v_q using (1) with acceleration/ aggression limits
I/II B	trim τ till it is feasible
	if (I)
I C	$< \tau, \tau_{strat} > \leftarrow$ Extend1(τ, τ_{strat}, max(v_q, v_{min}))
I→II	if τ satisfies (II), τ_{obs}=null
	else
II C, II→I	$< \tau, \tau_{strat}, \tau_{obs} > \leftarrow$ Extend(τ, τ_{strat}, max(v_q, v_{min}))
	end if
I/II D	Optimize(τ, v_q)
I A	if $\tau_{obs} \neq$ null $v_q \leftarrow$ maximum speed to stop at $\|\tau\|$-d_{obs}
	return $< \tau_{obs}, \tau, v_q >$

An optimal travel plan consists of optimal trajectory computation as well as optimal speed settings. Working in the joint space of trajectory and speed is not possible due to computational and modeling constraints. However, by constant adjustment of speeds and trajectories separately, a near-optimal path can be obtained.

For *I A*, speeding up is disallowed, which arises from the commonly seen behavior that drivers tend to slow down on seeing a static obstacle if a lane change is not allowed. It is expected that the vehicle would come to a standstill at a distance of d_{obs} before the static obstacle, where d_{obs} is the minimum longitudinal distance needed to overcome the static obstacle given by (15) and accordingly the speed is iteratively reduced.

$$d_{obs} = d_{obs}^{min} + \Delta\tau_{obs} \tag{15}$$

Here d_{obs}^{min} is a small fixed distance to be maintained from the static obstacle at all times, and $\Delta\tau_{obs}$ is the deviation between τ and τ_{obs} which is taken as the distance between last point in τ and the closest point in τ_{obs}. If the vehicle continues to follow τ it is expected that it would have to stop at the end due to no possible subsequent moves and wait for the other vehicles to clear. The distance d_{obs} gives enough scope for a vehicle to turn to avoid the static obstacle, in the absence of other vehicles. If the vehicle is directly ahead of the static obstacle, a larger steer would be required as compared to the case when only a small part of the vehicle is ahead of the static obstacle. A larger steer implies a requirement of a greater longitudinal distance which is modeled by $\Delta\tau_{obs}$. In case the vehicle actually stops, or almost stops, the subsequent extend operation (to move the vehicle when path is clear) is called with a small predefined speed v_{min}.

Trim without ending at static obstacles or *II B* is simply based on the feasibility of the plan τ as per the changed environment (if any) using the set feasibility constraints. For *I B* this needs to be performed for both τ and τ_{obs}. An infeasible τ_{obs} (Algorithm 3, Line 1) implies emergence of a new static obstacle, in which case everything needs to be invalidated and re-computed to ensure a collision free τ_{obs}, while τ follows the same concepts as *II B*. In conception, *extend* or step *C* is the same for both modes, however they are called differently in the algorithm to eliminate re-computation of τ_{obs} which is already known. The *extend* algorithm also monitors for mode changes. Step *D* is same for both modes.

From Algorithm 3 it can be seen that the 4 steps are performed one after the other for mode II. Mode checking conditions are introduced if a step is particular to a mode. For mode *I*, the algorithm starts with *I B* and ends with *I A* through *I D*. *I B* (Algorithm 3, Line 1) is placed at the start since its action invalidates the plan; while *I A* is kept at the end since in mode *I* speed is trajectory dependent, although once initiated, it does not matter which was the first step of the loop and which was the last. An exception to the algorithm is when the vehicle is initially to be found placed in an infeasible manner, or when not even a single step is feasible, in which case the immediate move is as directed by *LP* with maximum possible deceleration of $-acc_q^{max}$.

VI. Results

A. Simulations

To test the algorithm, a number of diverse scenarios were constructed, each aimed at testing a different aspect of the algorithm. In all the experiments vehicles are named in order of their appearance. We have included a supplementary video file showing the results. This is available at http://ieeexplore.ieee.org. In all of the experiments, initially the road segment is empty with only the static obstacles (if any). The vehicles are made to emerge at the positions and emergence time as visible in the results/supplementary video. All vehicles are generated in the direction of the road segment. Nothing is known about a vehicle by any other vehicle before it is visible in the scenario. In other words the vehicles suddenly appear in the scenario requiring other vehicles to adjust their plans accordingly.

The first experiment tests the ability of the algorithm to optimally drive a vehicle within a grid of complex obstacles (Figure 5(a)). An initial expectation was that the vehicle would take the central route, however the extra maneuvers around the obstacles made the trajectory larger. The vehicle always maintained comfortable distances from all the obstacles, while travelling with the maximum permissible speed.

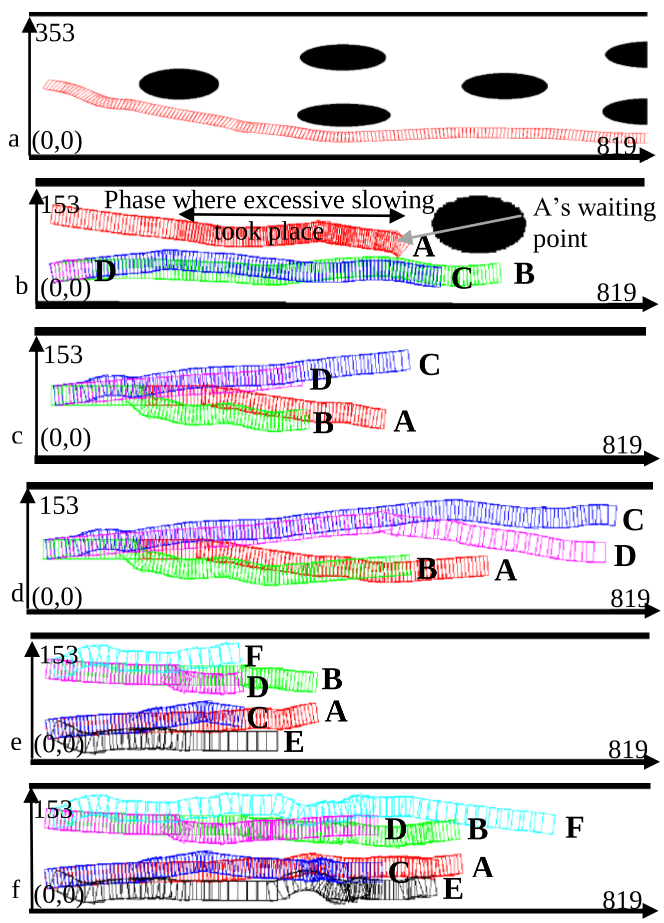

Fig. 5. Simulation Results (a) Computing the optimal path in a complex grid. (b) Slowing down to avoid a static obstacle while waiting for vehicles to clear. (c-d) Multiple overtaking, each from the optimal side. (e-f) Multiple overtakings, each after waiting for the correct time and in the right order.

Figure 5(b) shows a different scenario where *A* could not overcome a static obstacle and hence had to slow down and wait for the other vehicles (*B* and *C*) to clear. At a later stage it comfortably places itself before *D*. The scenario tests the algorithm's ability to restrain a vehicle from accidently pushing in front of another vehicle, which it may not be prepared for. Consider that *D*'s entry is made much earlier, in which case *A* would wait for it as well. If *A* is synthetically made to move, *D* would obediently follow. If *D* is synthetically made to slow down, *A* would not wait any longer. Hence individually the vehicles show expected behaviors, however having already waited for 2 vehicles to pass through *A* would be called too courteous to be waiting for *D*, which the algorithm currently simulates.

The next scenario tests the ability of the vehicle to decide the optimal direction for overtaking. The overtaking vehicles have infeasible entry conditions and the plans are made in multiple extension operations. *C* overtaking the two vehicles (*A* and *B*) in front is visible in Figure 5(c), while it is likely to be overtaken itself by *D*. Figure 5(d) shows how *D* overtakes *C*, while the other vehicles (*A* and *B*) themselves orient to enable the overtaking procedure to happen. All vehicles travel nearly at their maximum permissible speeds.

The last scenario is introduced to consider the traffic dynamics when overtaking is not initially possible and later still rather difficult due to competing vehicles. *E* entered the scenario before *F* and hence overtook the initial set of vehicles (*C* and *D*) earlier (Figure 5(e)). However it appeared that there was no room for overtaking the vehicle set ahead (*A* and *B*) and hence *E* had to rapidly slow down, so as to follow *A* ahead, while *F* succeeded in overtaking the initial set of vehicles (*C* and *D*). Subsequently *F* occupied enough room and succeeded in overtaking the vehicle set ahead (*A* and *B*, Figure 5(f)). Initially F slowed while space was being created, but later accelerated during overtaking. Henceforth the vehicles drifted to the other side to accommodate overtaking *E*. It must be emphasized that the scenario is largely driven by Lateral Potentials.

B. Parameters

The effects of the parameters of LP was demonstrated in [15]. The settings used in this paper however favor more sensitive settings, which led to the disadvantages of introducing oscillations and steep turns, while the advantages included the capability to avoid obstacles, however far or near. The limitations are however eliminated by the *optimize* algorithm, while the advantages remain.

Trajectory planning always involves the problem of a tradeoff between path length and average clearance. These factors are controlled by the parameters in the *optimize* algorithm. Experiments over a single obstacle scenario were carried out, for which the participating parameters are k_l and k_s. Both a large clearance and a short path length cannot be simultaneously achieved, which is clear from Figure 6. A large clearance leads to the vehicle quickly placing itself in the middle of the road and then between the obstacle and the road edge, while attempts to minimize the path length led to the vehicle travelling very close to the obstacle (Figure 3). Only parameter settings leading to feasible results are plotted. k_{coop} has a role similar to that presented in [15].

The worst case complexity of the extend algorithm is $O(2^{(|O|+|R|).L})$ where $|O|$ represents the number of static obstacles ahead, $|R|$ represents the number of vehicles ahead with a speed lower than the vehicle being planned, and L is the length of the segment being planned. In the presented scenarios, computing the trajectory for each vehicle takes less than a second, while the initial plan generation can take 2.5 seconds if τ_{obs} also needs to be computed. Scenario 1 takes 8 seconds to compute in total due to the computation of a single trajectory overcoming all static obstacles.

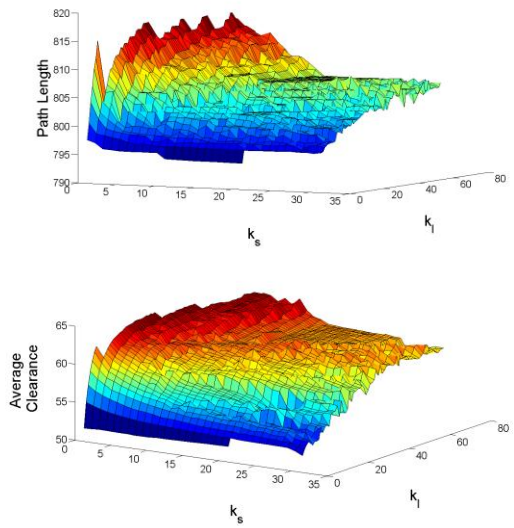

Fig. 6. Effect of change of algorithm parameters on obstacle avoidance (a) Path Length (b) Average Clearance

The high complexity is not a factor of concern considering that the length of segment planned can be adjusted to account for the number of obstacles. Further, for a high number of obstacles, the algorithm may fail to find a single trajectory which simultaneously avoids all the obstacles, forcing the algorithm to terminate.

The algorithm does not (in initial plan construction) have the liberty to reduce the speed. The uncertainties in projecting a vehicle's future motion increase with the vehicle's speed and with longer projections into the future. High uncertainties show as elongated projected positions of the vehicles which may be hard to surpass. In such a case the algorithm would construct a trajectory to overcome the initial set of obstacles first, and as the vehicle moves, the trajectory would be modified to overcome latter obstacles.

Avoiding closer obstacles is easier due to the high degree of certainty in position and speed, and preferable speed settings as compared to the obstacles which are much further away. The inability to overtake a vehicle may however result in vehicle following.

VII. Conclusions

This paper emphasized the notion of the unorganized nature of traffic in order to make travel efficient in a diverse transportation system. The problem of planning on a straight road was solved using the elastic strip concept. The algorithm was near-complete and near-optimal, while the computational cost was only just larger than reactive techniques. Using the algorithm the ability to maneuver in a complex obstacle framework was presented, while showing complex overtaking, vehicle following and waiting behaviors. This paper is hence a step towards bringing autonomy to currently unorganized traffic systems, whilst also providing a possible future alternative for currently organized traffic systems.

In the future, maximum lateral acceleration needs to be added in the feasibility definition to prevent trajectories being generated which cannot be traced. The simulation framework also needs to be extended to a real city map including intersections, diversions, crossings with traffic lights, etc. Currently there are no data sets recording unorganized traffic movement to validate such algorithms, which may further give better intuition of the traffic. Many present algorithms for vehicle tracking, obstacle detection, localization, etc. are currently lane based and these need to become non-lane based for such algorithms to be implementable on real roads.

Glossary

For Any general vehicle R_i (vehicle being planned denoted by Q, with all variables indexed q)

L_i (x_i', y_i', θ_i'): Position (X', Y', orientation), ΔL_i: uncertainty
v_i: Linear speed, $vpref_i$: Preferred linear speed, Δv_i: uncertainty
ω_i: Angular speed, ω_i^{max}: Maximum angular speed
acc_i^{max}: Maximum acceleration, agg_i: Aggression factor
ζ_{free}^{static}: Free workspace

For trajectory τ

$\tau(t)$: Planned position at time t
τ_{obs}: Trajectory considering only static obstacles
$\Delta\tau_{obs}$: Deviation between τ and τ_{obs}
τ_{strat}: Strategy (+1/left or -1/right) to avoid obstacles
$\tau_{strat}(i)$: Strategy to overcome obstacle i
T: Time till which motion is planned
$\tau(T)$ $\{\tau(T)[X'], \tau(T)[Y'], \tau(T)[\theta']\}$: Last planned position
d_{unc}: Minimum distance to maintain to overcome uncertain speed changes of vehicles
d_o: Minimum distance to overcome a static obstacle ahead
d_o^{min}: Minimum distance to be maintained from the static obstacle while overcoming it

For any general state *s* considered in planning

[v_a^s, v_b^s]: Speed bounds for safe travel
R: Set of vehicles considered for feasibility
(d_f, d_{ls}, d_{rs}, d_{fl}, d_{fr}, d_b): Distance from obstacle (ahead, left, right, forward-left, forward-right, back)
(p_f, p_s, p_d, p_b, LP): (Forward, Side, Diagonal, Back, Lateral) potential
$sen_{X'}$, $sen_{Y'}$, $sen_{X'Y'}$, $coop$: Lateral Potential parameters denoting sensitivities along axes/cooperation

For any general waypoint τ'_i in trajectory

$\tau`_0$: Initial position/ waypoint
τ_{obs}^i: Point in τ_{obs} closest to τ'_i
(F_l, F_s, F_{coop}, F_o, F_{total}): (Lateral, Spring extension, Cooperation, Drift, Total) force

k_l, k_s, k_{coop}, k_o: Optimization parameters denoting contributions of each force

References

[1] R. Claes, T. Holvoet, and D. Weyns, "A Decentralized Approach for Anticipatory Vehicle Routing Using Delegate Multiagent Systems," *IEEE Trans Intell Transport Syst*, vol. 12, no. 2, pp. 364-373, Jun. 2011.

[2] F. Jiménez and J. E. Naranjo, "Improving the obstacle detection and identification algorithms of a laserscanner-based collision avoidance system," *Transportation Research Part C Emerging Technologies*, vol. 19, no. 4, pp. 658-672, Aug. 2011.

[3] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouvelière, "Maneuver-Based Trajectory Planning for Highly Autonomous Vehicles on Real Road With Traffic and Driver Interaction," *IEEE Trans Intell Transport Syst*, vol. 11, no. 3, pp. 589-606, Sep. 2010.

[4] R. W. Hall and C. Caliskan, "Design and evaluation of an automated highway system with optimized lane assignment," *Transportation Research Part C Emerging Technologies*, vol. 7, no. 1, pp. 1-15, Feb. 1999.

[5] J. B. Sheu and S. G. Ritchie, "A new methodology for incident detection and characterization on surface streets," *Transportation Research Part C Emerging Technologies*, vol. 6, no. 5–6, pp. 315-335, Dec. 1998.

[6] J. Lee and B. Park, "Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicles Environment," *IEEE Trans Intell Transport Syst*, vol. 13, no. 1, pp. 81-90, Mar. 2012.

[7] M. Sarvi and M. Kuwahara, "Microsimulation of Freeway Ramp Merging Processes Under Congested Traffic Conditions," *IEEE Trans Intell Transport Syst*, vol. 8, no. 3, pp.470-479, Sep. 2007.

[8] D. Mohan and P.S. Bawa, "An analysis of road traffic fatalities in Delhi, India," *Accident Analysis & Prevention*, vol. 17, no. 1, pp. 33-45, Feb. 1985.

[9] L. Vanajakshi, S. C. Subramanian, and R. Sivanandan, "Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses," *IET Intell Transp Syst*, vol. 3, no. 1, 2009, pp. 1-9.

[10] O. Brock and O. Khatib, "Elastic strips: A framework for integrated planning and execution," in *Experimental Robotics VI, Lecture Notes in Control and Information Sciences Volume 250*, Springer, 2002, pp. 329-338.

[11] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," in *Proc. 1985 IEEE Intl. Conf. Robotics and Automation Vol. 2*, St. Louis, Missouri, IEEE, 1985, pp. 500-505.

[12] Y. K.Hwang and N. Ahuja, "A potential field approach to path planning," *IEEE Trans Robot Automat*, vol. 8, no. 1, pp. 23-32, Feb. 1992.

[13] Y. Yang and O. Brock, "Elastic roadmaps-motion generation for autonomous mobile manipulation," *Auton Robot*, vol. 28, no. 1, pp. 113-130, Jan. 2010.

[14] S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," in *Proc. 1993 IEEE Intl. Conf. Robotics and Automation*, IEEE, 1993, pp. 802-807.

[15] R. Kala and K. Warwick "Planning autonomous vehicles in the absence of speed lanes using lateral potentials," in *Proc. 2012 IEEE Intelligent Vehicles Symposium*, Alcalá de Henares, Spain, IEEE, Jun. 2012, pp. 597-602.

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

[17] S. J. Anderson, S. B. Karumanchi, and K. Iagnemma, "Constraint-based planning and control for safe, semi-autonomous operation of vehicles," in *Proc. 2012 IEEE Intelligent Vehicles Symposium*, IEEE, Jun. 2012, pp.383-388.

[18] K. Chu, M. Lee, and M. Sunwoo, "Local Path Planning for Off-Road Autonomous Driving With Avoidance of Static Obstacles," *IEEE Trans Intell Transport Syst*, vol.13, no.4, pp.1599-1616, Dec. 2012.

[19] R. Kala and K. Warwick, "Multi-Level Planning for Semi-Autonomous Vehicles in Traffic Scenarios based on Separation Maximization," *Journal of Intelligent and Robotic Systems*, 2013, DOI: 10.1007/s10846-013-9817-7.

[20] S. K. Gehrig and F. J. Stein, "Collision Avoidance for Vehicle-Following Systems," *IEEE Trans Intell Transport Syst*, vol. 8, no. 2, Jun. 2007, pp. 233-244.

[21] C. Frese and J. Beyerer, "A comparison of motion planning algorithms for cooperative collision avoidance of multiple cognitive automobiles," in *Proc. 2011 IEEE Intelligent Vehicles Symposium*, IEEE, Jun. 2011, pp.1156-1162.

[22] R. Kala and K. Warwick, "Motion Planning of Autonomous Vehicles in a Non-Autonomous Vehicle Environment without Speed Lanes," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 5-6, May-Jun. 2013, pp. 1588–1601.

[23] A. Furda and L. Vlacic, "Enabling Safe Autonomous Driving in Real-World City Traffic Using Multiple Criteria Decision Making," *IEEE Intell Trasp Syst Magz*, vol. 3, no. 1, pp 4-17, Spring 2011.

[24] R. Schubert, K. Schulze, and G. Wanielik, "Situation Assessment for Automatic Lane-Change Maneuvers," *IEEE Trans Intell Transport Syst*, vol. 11, no. 3, pp. 607-616, Sep. 2010.

[25] J. L. Baxter, E. K. Burke, J. M. Garibald, and M. Normanb, "Shared Potential Fields and their place in a multi-robot co-ordination taxonomy," *Robotics and Autonomous Systems*, vol. 57, no. 10, pp. 1048-1055, Oct. 2009

[26] R. Gayle, W. Moss, M. C. Lin, and D. Manocha, "Multi-robot coordination using generalized social potential fields," in *Proc. 2009 IEEE Intl. Conf. on Robotics and Automation*, IEEE, 2009, pp.106-113.

[27] R. Gayle, A. Sud, E. Andersen, S. J. Guy, M. C. Lin, and D. Manocha, "Interactive Navigation of Heterogeneous Agents Using Adaptive Roadmaps," *IEEE Trans Visual Comput Graphics*, vol. 15, no. 1, Jan. 2009, pp. 34-48.

Rahul Kala received his B.Tech and M.Tech degrees in Information Technology from Indian Institute of Information Technology and Management Gwalior, India. He is currently working towards his Ph.D. degree from the School of Systems Engineering, University of Reading, U.K.

He has authored 3 books and over 60 papers. His recent book is on robotic planning titled Intelligent Planning for Mobile Robotics: Algorithmic Approaches (Hershley, PA: IGI-Global Publishers, 2013).

Mr. Kala is a member of AAAI. Mr. Kala is the recipient of Commonwealth Scholarship and Fellowship Program scholarship from ACU and British Council; Lord of the Code Scholarship from RedHat and IIT Bombay; and GATE scholarship from MHRD.

Kevin Warwick received the B.Sc. degree from Aston University, Birmingham, U.K. and the Ph.D. degree from Imperial College, London, U.K. He has been awarded higher doctorates (D.Sc.) by Imperial College, London and the Czech Academy of Sciences, Prague.

He is Professor of Cybernetics at the University of Reading, Reading, where he is involved in research into artificial intelligence, control, robotics and cyborgs. He is the author or coauthor of more than 600 research papers and is perhaps best known for his experiments using implant technology.

Prof. Warwick was the recipient of The Future of Health Technology Award in MIT, was made an Honorary Member of the Academy of Sciences, St. Petersburg and received the IEE Senior Achievement Medal in 2004, the Mountbatten Medal in 2008, the Golden Eurydice in 2009, the Ellison-Cliffe Medal in 2011. He has also received Honorary Doctorates from the Universities of Aston, Coventry, Bradford, Robert Gordon, Portsmouth and Bedfordshire.