Optimal Control Based Approach For Autonomous Driving

Reza Dariani/ Stephan Schmidt/ Roland Kasper
Institute for mobile systems, Otto-von-Guericke University
Magdeburg, Germany
stephan.schmidt@ovgu.de

***Abstract*— In this paper based on vehicle single track model an optimal control strategy for autonomous driving is developed. The reference control values which lead the vehicle alongside a calculated optimal path by satisfying optimization objective criteria such as comfort and safety are generated by real-time path optimization level. In this level to avoid any collision and obstacle, one dimensional potential field obstacle avoiding approach based on the distance between vehicle and obstacle is used. In a prior level, which is called path planning the initial solution of the system which is the input of path optimization level is found. In this paper each level and the effect of initial solution on path optimization level is explained.**

Keywords—Autonomous driving; Path control; Path planning; Obstacle avoidance

I. INTRODUCTION

Based on the research done at the National Motor Vehicle Crash Causation Survey (NMVCCS) conducted from 2005 to 2007 on light vehicles several factors are associated in a crash in which driver error with 94% plays the most important and critical role [1]. This is the reason why modern vehicles are equipped with high number of sensors and driver assistance system to inform and warn driver in case of any danger such as Driver Monitoring System DMS which warns driver if the focus level of driver drops. In another kind of driver assistance system, the system has the ability to take partially the control of the vehicle in common situation like Adaptive Cruise Control ACC, which adjust automatically the vehicle speed in order to keep safe distance from vehicle ahead. Some of these systems which also support driver in critical driving situations such as Electronic Stability Program ESP, are already mandatory in the EU since 2011 [2]. Further development to reduce human error is autonomous driving which means driving without driver. Equipping vehicles with this technology will likely reduce crashes, energy consmtion, air pollution and also congestion cost [3] [4] [5].

Classical approaches for autonomous driving are mainly rules and maneuver based [6]. However predefining all critical situations is impossible, see the case of recent accident of the Google car in early 2016 [7], this is why in this work autonomous driving is based on mathematic formulation which allows taking decision on unforseen situations and restrictions related to the road and traffic situations. In this order when the data about road is known a path between start to end point can be found. Any known static obstacles can be avoided in this levels called graph generator and shortest path algorithm. Based on this path and a simple kinematic vehicle model, inputs of the system which lead the vehicle along this path can be found. These inputs are the initial solution of path optimization level. In this level based on single track model which describes the dynamics of the vehicle an optimal control problem is defined. Aim of this level is to find the inputs of the system which are driving force and steering angle velocity which lead the vehicle along a path which is optimal for different objectives such as energy consumptions, comfort and safety. In this level, in order to avoid any collision with obstacles, one dimentional potential field as obstacle avoidance strategy is added in form of penalty function to the objective function of optimal control problem. The main concept of autonomous driving used for this work is illustrated in Fig.1. Further details of this level are described in section IV.

Applying the inputs found in path optimization level to the vehicle may result different than optimal solution due to the model uncertainty and enviorment disturbances. In that event, path control level is added which generates additional inputs in order to reduce the difference between optimal solution and vehicle behaviour.

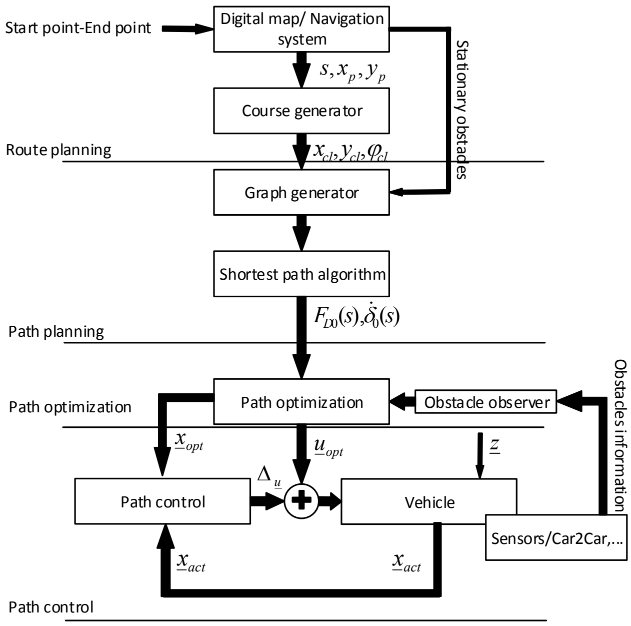

Fig. 1.Hierarchical concept of autonomous driving

978-1-5090-1314-2/ 16/$31.00

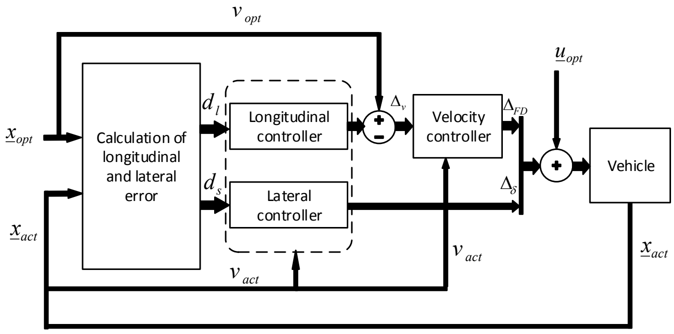

Fig. 2. Path control approach

Difference between vehicle longitudinal position and optimal longifudinal position is used as error in longitudinal direction in order to generate additional velocity which will be applied to velocity controller loop which generates driving force to reduce this error. And the difference between vehicle lateral position and optimal lateral position is used as error in lateral direction in order to generate additional steering angle in order to reduce lateral error. The main concept of path control level in illustrated in Fig.2.

The optimal control problem due to big dimension of the system and length of course is numerically hard to solve and require much calculation time to find optimal solution. In another hand, optimal problem must be updated in order to consider the dynamics of road and traffic situation. To solve the problem in an easier way and make the system real-time capable, a "Moving-Horizon Approach" MHA, is used, in which global optimization problem is partitioned into sequence of local optimization problems with an adequate smaller horizon. Updating and solving the problem with small horizon offers the possibility to update road dynamics which is useful in the case of sudden changes, obstacles or other vehicles on the lane. Fig.3 illustrates the main concept of Moving-horizon approach in which the optimal problem is solved for a horizon of τ second , then a part of this solution ξ called increment is saved which will be applied to the vehicle, then optimal control problem is updated and will be solved for a next horizon.

The outline of the paper is as follows: vehicle model used in this work is explained in chapter II. Chapter III explains the first two levels which are route planning and path planning which find initial solution for path optimization. Path optimization, obstacle avoidance strategy and the effect of intial solution on path optimization are explained in chapter IV and chapter V is conclusion.

II. VEHICLE MODEL

To describe vehicle dynamics single track model presented by Rieckert and Schunk is used [8]. In this model F_D and F_L are driving and load force respectively. The vehicle is regarded as a rigid body moving in *xy*-plane. To simplify the model front and rear wheels are summarized to one single wheel each. In this model roll and pitch angles are neglected and the tire dynamics are approximated by linear tire characteristic with saturation.

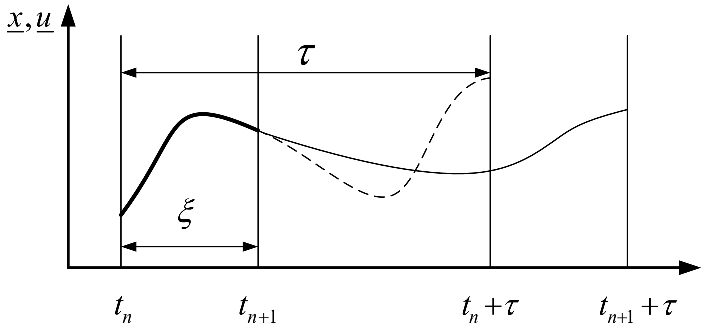

Fig. 3.Moving-horizon apprach

The systems states are coordinated at the center of gravity x and y in a vehicle global coordinate system. The yaw angle ψ and associated yaw rate ω describe the orientation of the vehicle. v describes the actual velocity of the vehicle and β named attitude or vehicle slip angle is the angle between velocity and vehicle longitudinal axis of the vehicle body AB to the direction of the vehicle velocity, v. β describes the difference between yaw angle and track angle, that is the result of the side forces F_{yf} and F_{yr} in lateral direction. The side forces are linear functions of the corresponding slip angles α_F and α_R, which are the nonlinear functions of the state variable β, ω , v and steering angle δ. Index F and R refer to front and rear respectively.

The nonlinear system

$$\underline{\dot{x}} = f(\underline{x}, \underline{u}) \tag{1}$$

With state vector $\underline{x}$

$$\underline{x} = [\beta\ \psi\ \omega\ v\ x\ y] \tag{2}$$

And input vector $\underline{u}$

$$\underline{u} = [\dot{\delta}\ F_{Dr}]^T \tag{3}$$

Is described as

$$\begin{gathered} \dot{\beta} = \psi - \frac{1}{m.v} \cdot \left[(F_D - F_L).\beta + F_{yf} + F_{yr} \right] \\ \dot{\psi} = \omega \\ \dot{\omega} = \frac{1}{J_{zz}} \cdot \left(F_{yf} \cdot l_F - F_{yr} \cdot l_R \right) \\ \dot{v} = \frac{1}{m} \cdot (F_D - F_L) \\ \dot{x} = v \cdot \cos(\psi - \beta) \\ \dot{y} = v \cdot \sin(\psi - \beta) \end{gathered} \tag{4}$$

Where m is vehicle mass, and J_{zz} is the moment of inertia. Tire slip angles for front and rear wheel by assuming that only front wheel can steer, are:

$$\alpha_f = \delta_f - \beta + \frac{l_f \cdot \dot{\psi}}{v} \tag{5}$$

$$\alpha_r = \beta + \frac{l_r \cdot \dot{\psi}}{v}$$

To calculate side forces F_{yf} and F_{yr} , in normal driving situations when the tire slip angle does not reach its critical value, a linear formulation with saturation can be used in which C_α is tire stiffness:

$$F_y = \begin{cases} -C_\alpha . \alpha, & |\alpha| \le \alpha_{max} \\ \pm F_{ymax}, & |\alpha| > \alpha_{max} \end{cases} \quad (6)$$

For further explanation of Single track model and tire dynamics, see [9].

III. PATH PLANNING

Path planning in simple terms means find a path between start to end point [10] [11]. As autonomous driving in this work is based on optimal control, therefore initial solution is needed. Hence a path planning level is used to find a rough path between start to end point and then based on this path and a simple vehicle model initial inputs can be found. In that event path planning can be considered in two stages, see Fig.1. At first stage which is called route planning, by knowing start and end point a course which connect these two points can be found with help of digital maps or a navigation systems. Generating discrete points (x_p, y_p) between start and end point offers the possibility to find the course geometrical approximation with cubic spline interpolation. At second stage, in order to find a path in this course as explained before, set of graphs must be generated. There are different methods such as Rapidly-exploring Random Trees RRTs [12] or Probabilistic RoadMap PRM [13] [14] which are efficient methods to find a path in unconstructed environment but this advantage can be a disadvantage for autonomous driving as both methods develop their vertices and trees in a random way and the probable path between start point to end point may not be desirable for autonomous driving. Also the course may not be uniformly covered. Another important factor which must be considered is to find a path which satisfy also driving fashion. For example an oscillatory path, or a path which is near road boundary when the center line is free is not desirable but they might be valid. As the width of the road in which vehicle must drive is known, definition of the nodes which are used to generate graphs can be predefined instead of probabilistic, which also has the advantage of considering all the workspace in a uniform way.

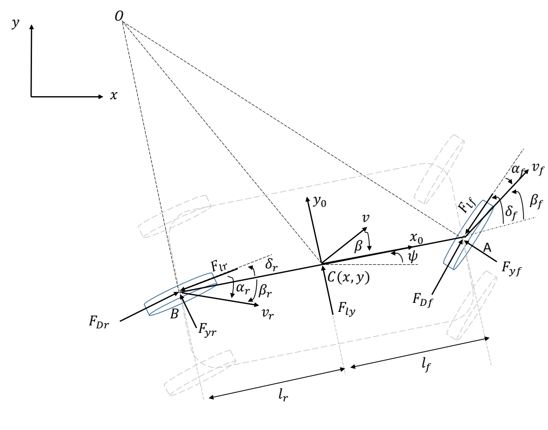

Fig. 4. Single track vehicle model

Using Predefined RoadMap instead of Probabilistic RoadMap offers the possibility to find a path by considering all the course uniformly. A region of connectivity can be defined between nodes which helps to save calculation time [15]. Smaller distance between nodes definition results in a smoother path. As the nodes only added to the free part of the course, therefore no nodes is located to any known static obstacles position. After defining the course in the form of graphs, a shortest path algorithm can be used to find a path between start to end point. Dijkstra Algorithm [16] is able to find the shortest path from a given node u_{start} to all vertices. In this order for a given graph $G = (V, E)$ consisting of a set of vertices V and a set of edges $E \subset V \times V$, and for each edge $(u, v) \in E$ an associated nonnegative cost $c(u, v)$ which is often the distance of the motion is assigned. Another important algorithm is A* [17] which is an extension of Dijkstra algorithm that tries to reduce the total number of edges explored by incorporating a heuristic estimate of the cost to get to the goal point. Therefore instead of finding the shortest path to all the vertices in the graph, it is only interested in a shortest path to a specific goal vertex u_{goal}. In that event a heuristic value $h(u)$ is added to $g(u)$ which is the distance from vertex u to the goal vertex. Both algorithm are explained in the next table. In case of no obstacle the shortest path between start point to end point passes through center line but in case of any known static obstacle as there is no nodes in obstacle position, the obstacle will be avoided.

To adapt the avoidance maneuver with urban rules and driving fashion two more heuristic functions can be added to the algorithm which are left side overtaking and returning back to center line after avoiding maneuver. Considering the distance of each node to center line as a heuristic cost makes the nodes which are close to center line preferable which is useful after overtaking maneuver. Left overtaking heuristic function can be made from center line heuristic function, simply by multiplying the cost of nodes which are situated on the lower part of center line by a factor bigger than one. This multiplication makes the center line heuristic cost asymmetric for the nodes upper and lower of the center line. This multiplication makes the cost of the nodes of the lower part higher, therefor overtaking path is preferred from upper side of the lane which satisfies the urban rules and driving fashion.

TABLE I. Shortest path algorithms

Dijkstra algorithm	**A* algorithm**

for all $u \in V$ **do**
$g(u) \leftarrow \infty$
$g(u_{start}) \leftarrow 0$
Insert u_{start} **into** Q
repeat
u ←element from Q with minimal $g(u)$
Remove u **from** Q
for all neighbours v of u **do**
if $g(u) + c(u,v) < g(v)$ **then**
$g(v) \leftarrow g(u) + c(u,v)$
$bp(v) \leftarrow u$
Insert or update v in Q
until $Q = \emptyset$
	for all $u \in V$ **do**
	$g(u) \leftarrow \infty$
	$h(u) \leftarrow$ distance between u and u_{goal}
	$g(u_{start}) \leftarrow 0$
	Insert u_{start} into Q
	repeat
	u ←element from Q with minimal $g(u) + h(u)$
	Remove u **from** Q
	for all neighbours v of u **do**
	if $g(u) + c(u,v) < g(v)$ **then**
	$g(v) \leftarrow g(u) + c(u,v)$
	$bp(v) \leftarrow u$
	Insert or update v **in** Q
	until $u = u_{goal}$ **or** $Q = \emptyset$

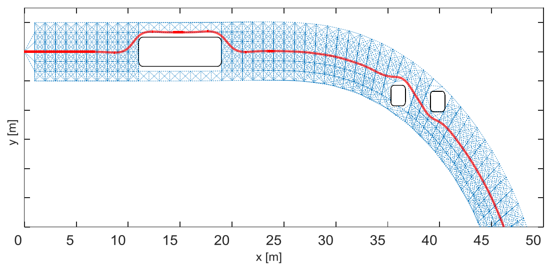

Fig. 5. Path planning with static obstacle

Fig.5 shows the result of path planning with a static obstacles. The graphs are shown with blue lines and obstacles with black rectangle. The red line illustrates the path from start point to end point by avoiding static obstacles.

As mentioned before aim of path planning level is to find initial solution for optimal control problem. Hence by using this path and a simple kinematic model in which the effect of slip angle is ignored a rough initial solution can be found [18]. By using the geometry calculation the corresponding steering angle velocity which lead the vehicle through the path can be found. And as the valid velocity for the given course can be known via digital map or navigation system, the initial solution for the driving force can be conducted based on this velocity for known load of the vehicle.

IV. PATH OPTIMIZATION

The objective of path optimization level is to generate reference actuation values by taking into consideration boundaries and constraints which allow vehicle leading on an ideal curve. The ideal curve is the numerical solution of an optimal control problem, where driver is modeled as penalty function, and car is given as the dynamic system (1). The optimal control problem is solved with sequential quadratic programming SQP solver. NLPQLP solver, solves the nonlinear programming problems by a *SQP* algorithm. This solver is developed by K.Schittkowski [19] which is a Fortran implementation.

The optimization problem is given as

$$\text{Min } J(\underline{x}, \underline{u}) \tag{7}$$

With dynamic system and nonlinear constraints

$$\begin{gathered} \underline{\dot{x}} = f(\underline{x}, \underline{u}) \\ \underline{g_l} \leq \underline{g}(\underline{x}, \underline{u}) \leq \underline{g_u} \end{gathered} \tag{8}$$

As well as states and inputs restrictions

$$\begin{gathered} \underline{X_l} \leq \underline{x} \leq \underline{X_u} \\ \underline{U_l} \leq \underline{u} \leq \underline{U_u} \end{gathered} \tag{9}$$

Objective function definition is critical to find the ideal trajectory. The main focus will be in optimizing the inputs of the systems which are force and steering angle velocity by traveling to end point or destination. In order to minimize the distance to goal, subtraction of the vehicle travelled distance at the end of horizon from the traveled distance at beginning of the horizon can be used. (10) shows the objective function in which aim is to minimize the travelled distance to end point with minimal inputs:

$$J(\underline{x}, \underline{u}) = -R_S . [s(t_n + \tau) - s(t_n)] + \int_{t0}^{tf} \underline{u}^T \cdot \underline{R_u} \cdot \underline{u} dt \tag{10}$$

(10) results to an optimal solution where most distance is covered with minimal control inputs. R_S and R_u in (10) are the weight to manipulate the effect of each term in objective function. The restriction and limitation of inputs and states can be considered inside optimal control problem in the form of (8) and (9) but as in this kind of definition the border of validity/invalidity is sharp which means the considered parameters must be exact at each iteration and as (8) and (9) show the validity is only when the inequalities are satisfied, otherwise the solution is invalid and a new solution which satisfy the inequalities must be found. The disadvantages of these kinds is when the solution of the optimal control problem results in the border of validity-invalidity. The solution in this case may be again near the border of validity-invalidity which is not suitable as it requires more iterations. Let's imagine the case of a vehicle which drives in a course with a given road width. In this case a solution near course border is valid but is not desirable. A way to solve this kind of problem is to present a new type of the constraint named *soft constraints*. In this form, the optimal control problem is formulated without any nonlinear constraints, instead the constraints are added to the objective function in form of penalty functions. Therefore the solution of optimal control problem also satisfy constraints. To generate the penalty function, a margin is taken into consideration and penalty function grown in a quadratic form after this margin. Fig.6 illustrates the soft constraint and constraints as form of inequalities which can be called also as hard constraint. Any type of restriction and limitation such as maximum speed, lateral and longitudinal acceleration can be taken into consideration in this form.

To avoid any collision with static and dynamic obstacles an avoidance strategy must be added in this level. There are several approaches to avoid collision such as classical approaches like configuration space [20] in which obstacle avoidance is considered as a planning problem and the focus is on path planning problem which provides the low level control with a collision free path, therefore the path must be precisely specified and inaccurate model or slight error in the motion may result in a collision.

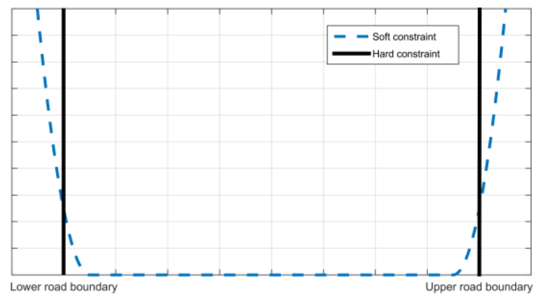

Fig. 6. Hard constraint vs. Soft constraint

Another approach is artificial potential field. The main idea of potential field is that vehicle moves in the field of the forces and the position to reach is an attractive pole and obstacles are repulsive surfaces for the vehicle. Another approach which can be used is velocity obstacle [21] in which the aim is to find the velocity vector which avoid any collision. There are more approaches to avoid any collision but efficiency and simplicity are the main key as remaining real-time is crucial. Simplicity of artificial potential field is beneficial and it can be added to the optimal control level easily as travelling to the end point which is also attractive point is already concluded in optimal control (7), therefore only repulsive surfaces must be defined. In that event a circumscribed shape must be defined around obstacles as repulsive surface. Rectangle shape cover the vehicle shape definition very good as another vehicles are the most probable obstacle on the road, but as its mathematic definition is complicated an ellipse shape can be a good replacement. As this shape must be a repulsive surface, therefore based on the distance of the vehicle to this surface a penalty function can be designed in order to keep the vehicle out of this surface. In that event by using polar coordinate system the distance between vehicle and obstacle is measured, Fig.7 up, and then obstacle avoiding penalty function, Fig.7 down, keeps the vehicle out of the surface. As it is shown in the figure when the distance of the vehicle to the obstacle is less than defined surface, the cost in objective function increase exponentially. Hence optimization solver in order to reduce the cost must find a solution which keep the vehicle outside this surface. l and w in the figure refer to the length and width of the vehicle plus length and width of our vehicle as the vehicle is considered by its geometrical center. While driving in an environment with several uncertainties, the potential field approach easily allows to extend the size of ellipse with respect to the predicted uncertainties. Even when the ellipse grows with larger forecast horizon, the street will not be blocked completely due to the soft constraint character of the approach.

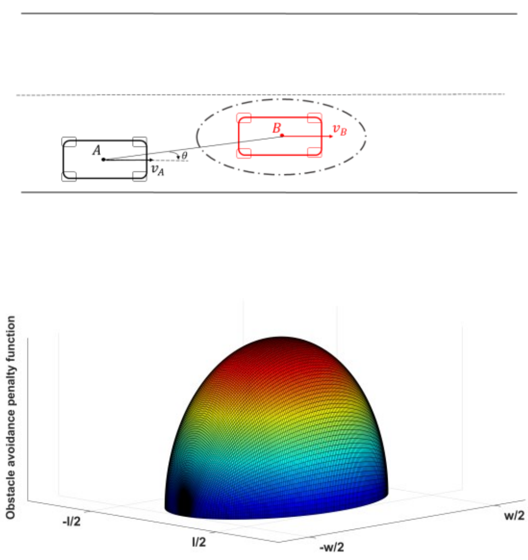

Fig. 7. Up: Avoidance distance calculation. Down: Obstacle avoidance penalty function

Another advantage of this approach is in the case of pedestrian and cyclist. In this case by increasing the width and length of the surface to the width of the road, road can be considered as blocked and vehicle instead of finding an overtaking maneuver, reduces its velocity or stops and lets the pedestrian or cyclist cross the street and accelerate again. In order to show the functionality of optimal control approach with obstacle avoiding, an avoidance strategy is simulated in which aim is to avoid a dynamic obstacle, here another vehicle, which moves with lower velocity than our vehicle in the same direction. Therefore aim is to overtake the vehicle and after return back to initial lane. Street has two unidirectional lanes, each 3m width. Vehicle has the velocity of 5.6 $^{m}/_{s}$ and obstacle $2^{m}/_{s}$. As dynamic obstacle are not considered in path planning level, therefore the initial solution pass through the center line of the lane. Blue line is the position of the geometrical center of the vehicle. In Fig. 8 blue points show the position of the vehicle and red points position of the obstacle for different times. t_0 is the starting time which shows the initial position of the vehicle and obstacle and points are sampled each second.

Fig.9 illustrates a pedestrian avoidance scenario in which the vehicle drives with the velocity of $8^{m}/_{s}$ and $40m$ ahead a pedestrian with velocity of $1.4^{m}/_{s}$ crosses the street. The blue line is vehicle trajectory and blue circle and red stars are vehicle and pedestrian trajectory sampled each one second. As it is shown when the vehicle faces pedestrian it reduces its velocity, pedestrian cross the street and vehicle accelerate again to its previous velocity. In this case, as the ellipse size is consider big, the vehicle reduces its velocity instead of searching an overtaken maneuver. Velocity of the vehicle for all course and given sampled time is shown in Fig.10.

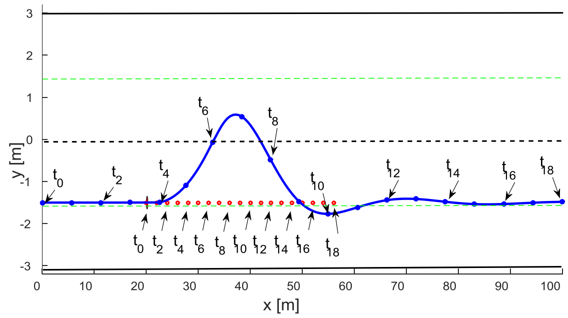

Fig. 8. Dynamic obstacle avoidance

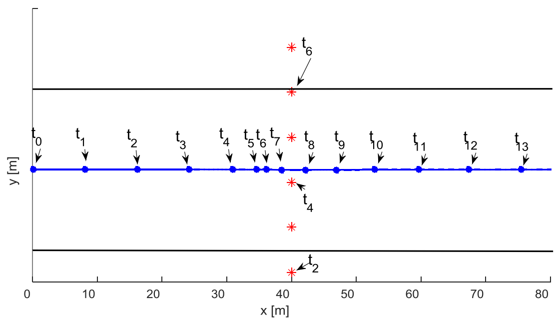

Fig. 9. Pedestrain avoidance

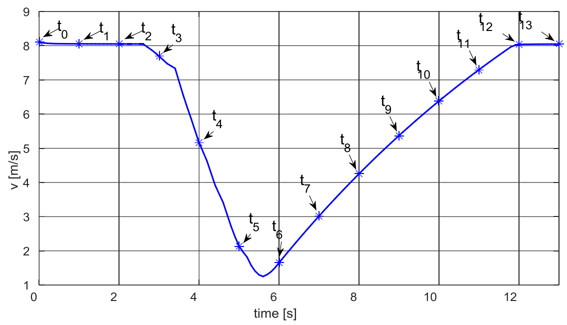

Fig. 10. Vehicle velocity while pedestrain avoidance

In previous examples, obstacle were dynamic and consequently they were not considered in path planning level. In order to show the effect of initial solution on path optimization level an avoidance scenario with static obstacle is simulated once by considering static obstacle in path planning level and once without. In Fig.11 and Fig.12 vehicle trajectory is shown with blue line and initial solution with red dotted line. In Fig.11 static obstacles are not considered in path planning level, therefore the initial solution is through the center line of track and collides with all the obstacles but in Fig.12 obstacles are considered in path planning level and initial solution avoid all of them. As it is shown in both figures, all the obstacles are avoided in path optimization level but the difference remains on the complexity of the problem and consequently on calculation time. When the initial solution has collision with all the obstacles makes the optimal control problem more complicated and path optimization level needs more iteration in order to find optimal solution as the initial solution results in collision which is not desirable and path optimization must find a collision free path. Comparing the number of iterations needed to find optimal solution as a reference of complexity of the problem shows that when obstacle are considered in path planning reduces 28% complexity of the given problem and gives 30% improvement on calculation time. In another hand result of the case that obstacles are not considered in path planning shows that path optimization level is able to find a collision free path but with more calculation time.

As the previous example shows, when the obstacles are considered in path planning level it reduces the complexity of the problem. But as the dynamic obstacles are not considered in path planning level it may happen that taking same maneuver as path planning level is not possible due to a dynamic obstacle at given time. In this case path optimization avoids the obstacles but due to the deviation between taking maneuver and initial solution optimal problem is more complicated. This scenario is illustrated in Fig.13 in which there is a static obstacle on the road which is shown with black rectangle. Therefore the initial solution is generated to avoid it which is shown with dotted gray line. On the second lane of track there is another vehicle which makes the right maneuver impossible. Hence our vehicle must take a left maneuver in order to avoid both static and dynamic obstacles.

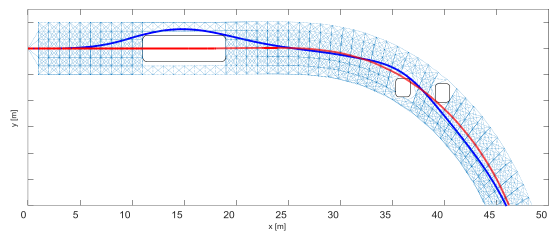

Fig. 11. Static obstacle avoidance without considerint the obstacles in path plannig level

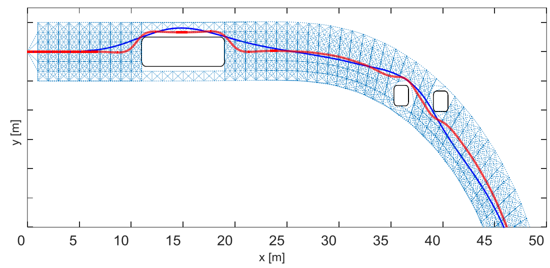

Fig. 12. Static obstacle avoidance with considerint the obstacles in path plannig level

In this case the initial solution is a bad initial solution and increase 60% complexity of the given problem and 52% computational time comparing to the case in which in path planning level, static obstacle is not taken into consideration and initial solution is through center line. In the figure, vehicle optimal trajectory when the obstacle is considered in path planning is shown with blue line and when the obstacle is not considered is shown with red dashed line.

Suggested solution to avoid stated problem without losing the generality of the path planning and path optimization levels and benefit from positive effect of path planning is to update the path planning based on the previous solution of path optimization. In this case when right maneuver for previous example is chosen to overtake the stationary obstacle by path optimization, path planning problem based on vehicle position, which is on the right side of the lane, can be updated and new path from this position can be found.

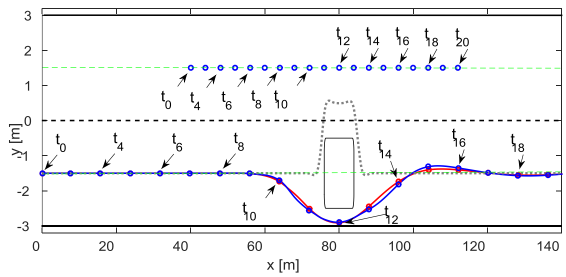

Fig. 13. Static and dynamic obstacle avoidance

In this case the new path to overtake the stationary obstacle is from right side as the shortest path from the actual vehicle position is through right maneuver.

Road dynamics can be also considered in path planning level, which means not only considering the stationary obstacle but also moving obstacle. In this case path planning problem must be updated and new problem must be solved based on the last states of the path optimization problem by considering stationary and dynamics obstacles. In that event, first, initial solution for given course is generated in path planning level for a period of time t_p in which $t_p > \tau$. Then based on this initial solution an optimal path for a horizon τ is found on path optimization level and a part of the optimal solution ξ is sent to the vehicle. Based on the position and velocity of the vehicle at the end of increment ξ, path planning is updated and a new path is found for new t_p and this new path is the initial solution for next horizon τ. Path planning with update from path optimization is illustrated in Fig.14.

Fig.15 illustrates timing between different levels of hierarchical concept with update between path planning and path optimization. As it is shown at beginning in path planning level a path for time interval of $[t_0, t_p]$ is found, then base in this path as initial solution an optimal path is found for time interval of $[t_0 + \xi , t_0 + \xi + \tau]$ and a part of this solution, an increment ξ, is sent to control level. Transition between each level is shown with a diagonal black array. Path planning in this level cannot be updated, because to use the result of path planning it must be updated at $t_0 + \xi$ and as it is shown there is no path optimization on that time to be used to update path planning problem. Therefore, initial path planning is used to run second path optimization starts at $t_0 + 2\xi$ and one increment of this result is sent to control level. At time $t_0 + 2\xi$ based on the vehicle position found in first path optimization $[t_0 + \xi , t_0 + \xi + \tau]$ at time $t_0 + 3\xi$ a new path planning is run for time interval of $[t_0 + 2\xi, t_0 + 2\xi + t_p]$.

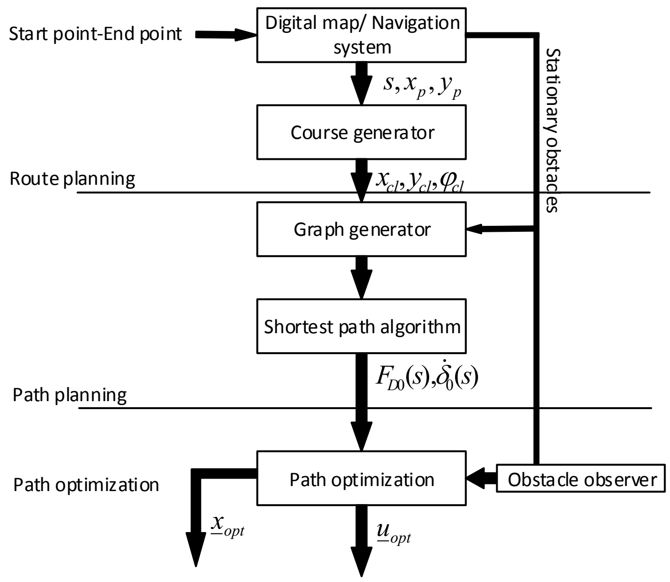

Fig. 14. Path planning with update from path optimization

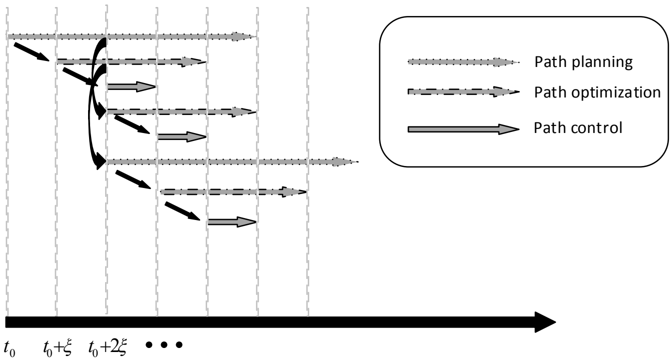

Fig. 15. Timing between different levels of hierachical concept

Transition between this two levels is shown with a curved black array. Initial solution of this path planning is used for next path optimization problem. And this sequence continues.

V. CONCLUSION

In this paper an optimal control based hierarchical concept for autonomous driving has been explained. Path planning which find the initial solution for optimal control problem and its effect on optimal control problem has been explained. Path optimization level and construction of objective function for autonomous driving has been explained. Implementation of a potential field as a strategy to avoid collision has been explained. And at the end a solution to use path planning by considering dynamic obstacle and its linkage with path optimization has been suggested.

[1] „A brief statistical summary. Critical reasons for crashes investigated in the national motor vehicle crash causation survey," National Highway Traffic safety administration, February 2015.

[2] L. D'Orazio, F. Visintainer und M. Darin, „Sensor Networks on the Car: State of the Art and Future Challanges," 2011.

[3] „National Highway TRaffic Safety Administration_Preliminary Statement of Policy Concerning Automated Vehicles," [Online]. Available: www.nthsa.gov/Research/Crash+Avoidance/Automated+Vehicles.

[4] D. S. Zuby, „The Promise and Reality of Crash Avoidance Technology," in *Lifesavers National Conference*, Nashville,TN, 27 April 2014.

[5] J. M.Anserson, N. Klara, K. D.Stanley, S. Paul, S. Constantine und A.Oluwataba, Autonomous Vehicle Technology, A Guide for Policymakers, RAND corporation.

[6] A. Kelly und A. STentz, „Rough Terrain Autonomous Mobility-Part 1: Technical Analysis of Requirements in Autonomous Robots," pp. 129-161, 1998.

[7] N. Lavars, „Google's autonomous Lexus self-drives into the side of a bus," 29 February 2016.

[8] P. Rieckert und A. Schunk, „Zur Fahrdynamik des gummibereiften Krafftfahrzeuges, Ingenieur Archiv," 1940, pp. 210-244.

[9] R. N.Jazar, Vehicle Dynamics. Theory and Application, Springer Science+ Business Media, LLC, 233 Spring Street, Newyork, NY 10013, USA.

[10] D. Ferguson, M. Likhachev und A. Stentz, „A guide to Heuristic-based Path Planning," in *American Association for Artificial Inteligence*, 2005.

[11] O. Khatib, „Real-Time Obstacle Avoidance for Manipulators and Mobile Robots,“ *The international Journal of Robotics Research,* Bd. 5, Spring 1986.

[12] S. LaValle, „Rapidly-exploring Random Trees; A new tool for path planning,“ October 1998.

[13] L. Kavarki und J. Latombe, „Randomized preprocessing of configuration space for fast path planning,“ in *Proc.IEEE int.Conf. on Robotics and Automation*, 1994.

[14] M. Overmars, „A Random Approach to Motion Planning,“ Technical Report RUU-CS-92-32, Universiteit Utrecht, 1992.

[15] J. P. v. d. Berg, „Path Planning in Dynamic Environments,“ 2007.

[16] E. Dijkstra, „A note on two problems in connexion with graphs,“ in *Numerische Mathematik*, 1959.

[17] P. Hart, N. Nilsson und B. Rafael, „A formal basis for the heuristic determination of minimum cost paths,“ *IEEE Transactions on Systems Science and Cybernetics,* pp. 100-107, 1968.

[18] Z. Schiller und Y. Gwo, „Dynamic Motion Planning of Autonomous Vehicles: Robotics and Automation,“ *IEEE Transaction on.In: Robotics and Automation,* pp. 241-249, 1991.

[19] K. Schittkowski, „NLPQLP: A Fortran Implementation of a Sequential Quadratic Programming Algorithm with Distributed and Non-Monotone Line Search - User's Guide,“ 2010.

[20] T. Lozano-Perez, „Spatial Planning: A configuration Space Approach,“ *IEEE Transaction on Computers,* pp. 108-120, February 1983.

[21] P. Fiorini und Z. Shiller, „Motion Planning in Dynamic Environments Using Velocity Obstacles,“ *The International Journal of Robotics Research ,* Bd. 17. no.7, pp. 760-772, July 1998.

[22] Z. Shiller und Y. Gwo, „Dynamic motion planning of autonomous vehicles: Robotics and Automation,“ *IEEE Transactions on.In: Robotics and Automation,* pp. 241-249, 1991.