



HAL
open science

A three level trajectory planning for autonomous transportation systems

Duc Thang Nguyen, Hadi Zaatiti, Mohamed Tlig, Jean van Frank

► **To cite this version:**

Duc Thang Nguyen, Hadi Zaatiti, Mohamed Tlig, Jean van Frank. A three level trajectory planning for autonomous transportation systems. CONGRÈS : SIA Simulation Numérique digital, Apr 2021, En ligne, France. hal-03202335

HAL Id: hal-03202335

<https://hal.science/hal-03202335v1>

Submitted on 19 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A three level trajectory planning for autonomous transportation systems

Duc Thang Nguyen, Hadi Zaatiti, Mohamed Tlig and Jean Van Frank

Institute of Research and Technology SystemX

8 Avenue de la Vauve, 91120, Palaiseau

Email: duc-thang.nguyen@irt-systemx.fr

hadi.zaatiti@irt-systemx.fr

Autonomous vehicles are considered as means of transport for the future. However, the design and the development of autonomous systems such as public transport, robot taxi and good delivery are challenging. On one hand, it involves the maturity of autonomous vehicles. On the other hand, it relates to the management of these new services. One of the major considerations for these systems is how to safely navigate autonomous vehicles to fulfil their missions. In this paper, we propose a method for navigating autonomous vehicles through a three level trajectory planning. This method includes: 1) the computation of a global trajectory for each vehicle, 2) a guidance for hundreds of meters in front of the vehicle and 3) a real time computation for an immediate trajectory based on the current situation. This method provides the guidance which is easy to interpret by the autonomous vehicles for their decision making so that the safe navigation is insured. Additionally, the approach reveals to be flexible for incorporating trajectory adjustments.

Keywords: Autonomous vehicle, navigation, trajectory planning, public transportation

1 Introduction

Autonomous transportation systems are fundamentally challenging to design, to develop, to verify and to validate. To navigate the autonomous vehicle, the trajectory planning remains a difficult task as it involves a lot of parameters such as the dynamics of the vehicle, the internal and external disturbances and the traffic situation. That means, some of the needed parameters for the trajectory planning are known beforehand while others are known only in real time operation. Several important works [1–4] studied the problem of elaborating accurate, yet computationally inexpensive, trajectories for autonomous vehicles. For example [5] proposes an algorithm that performs a two-step computation, a first step chooses manoeuvres according to data from the environment that would minimise the risk of

collisions, and the second step chooses the trajectory that best preserves resources and ensures on board comfort. [1] studied trajectory planning in the case of multi-modal transportation systems by combining multi-agent methods with classic optimisation. These works focus on the trajectory planning from the vehicle point of view. In this paper, we consider the trajectory planning from a system point of view which is applicable to different contexts of autonomous services.

We address use cases relative to future smart cities, specifically intelligent transportation system networks for good deliveries, passenger transportation and on-demand taxis like services. These use cases require, always, the management of the services and accomplishing the mission, while insuring security and safety.

The paper is organised as follows: Section 2 provides a global overview of the proposed methodology by briefly presenting the three levels of the trajectory, the next sections detail each level of the trajectory. Section 6 presents an implementation of the method and simulation results.

2 Method overview

The objective of the method is to effectively organise the information of the trajectory so that the autonomous vehicle could easily understand and incorporate them into its decisions. The method focuses on defining the following three levels of the trajectory:

1. The global trajectory: The trajectory that the autonomous vehicle has to fulfil for the mission. The global trajectory is computed based on the Digital City Map (DCM) which provides information regarding the route network such as: road lane, road restrictions, position of traffic lights, type of infrastructure (intersection, roundabout).
2. The intermediate trajectory: the one which provides a guidance in hundreds of meters ahead of the vehicle.

3. The immediate trajectory: the one which provides the speed reference and the steering reference based on the current driving situation.

The proposed method requires the following assumptions:

1. The vehicle low level controls (including the lateral actuator controller and the longitudinal actuator controller) are well designed and developed.
2. The sensors of the autonomous vehicle are powerful enough to be able to detect and represent the environment involved in the vehicle navigation.
3. The autonomous vehicle runs always on the road with detectable road marks.

The first assumption insures that the vehicle can follow the immediate trajectory (the speed reference and the steering reference) and the second one insures the reliable perception for the driving algorithm. The third assumption ensures that the vehicle is not in off-road operation.

3 Global trajectory

The computation of the global trajectory is based on the following information:

1. The Digital City Map (DCM): The DCM like Google map is based on the Global Positioning System (GPS). It provides a complete set of information about the road network. Once the trajectory is defined between a point A and a point B on the DCM, all information along the trajectory can be collected and used for the navigation of autonomous vehicles.
2. Global Positioning System (GPS): This information is used to estimate the position of the vehicle in order to define the departure point of the global trajectory. An example of the use of the GPS position is to compute the trajectory for a taxi that should pick a customer or to start to drive the customer to his/her destination. The position provided by the GPS is not required to be very accurate because it is not used for the navigation.
3. Information regarding road blocks and other events that could potentially impact the trajectory and that are known beforehand (e.g., weather conditions, demonstration at a specific time and so on).

Figure 1 provides an example of a global trajectory which is extracted from the Google map. Note that the map visualisation is intended for human use. When this trajectory is used for autonomous vehicle, the information transmitted to the vehicle could be much different.

Mathematically, the city road map can be seen as a large weighted graph of nodes and edges. Each node represents a position where the vehicle could be offered one or several choices, and an edge is a sector

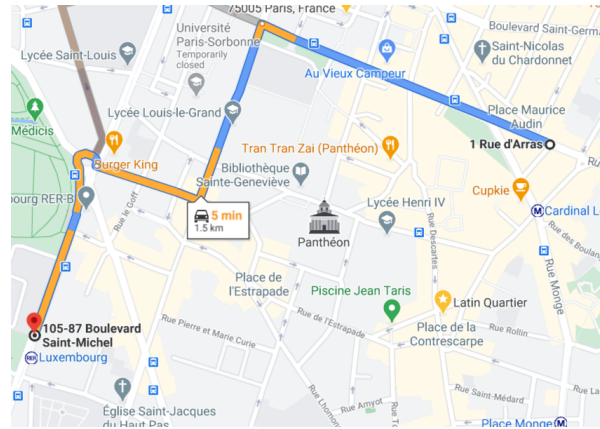


Fig. 1. Example of a Digital City Map (DCM) provided by Google Maps: the DCM provides a global trajectory from current position to destination

of road on which the vehicle action is limited to only going along the road, if not it will get off the road.

By considering the city road map as a graph of nodes and edges, the search for a global trajectory from a point A to a point B can therefore be reduced to a multi-objective optimisation problem which is very well studied in the literature. For example [2, 5–7] are based on known methods such as generalised Dijkstra, Breadth First Search (BFS) or AStar algorithms. With such algorithms in hand, updating dynamically the global trajectory when dealing with new constraints such as a road block or serious traffic jam could be possible.

3.1 Global trajectory optimisation

The computation of the global trajectory is formally an optimisation operation that finds the best global path given the graph. The main input for this optimisation task is an informative and accurate graph which requires the use of big data collected from the DCM. Once the input is provided, an effective search is employed to find the optimal path. Some constraints for the optimisation could be taking the shortest crossing time and/or consuming the least energy. Simple examples of objectives for different services are found in Table 1.

Use case	Use case specific constraints
Bus	Time of arrival should be close to predefined timetables
Taxi	Time of arrival should be minimal while having a comfortable ride
Delivery	Time of arrival should be minimal

Table 1. Examples of use-case specific objectives for finding an optimal global trajectory

3.2 Trajectory instruction

Once the global trajectory is identified, the next step is to “translate” it into a comprehensive machine data and then transmit it to the autonomous vehicle. One effective way, which is employed in this work, is to consider the global trajectory as a list of consecutive road sector and events, and the obligations attached to each of them on the list.

Let us consider an example of a global trajectory for the bus use case. The sequence of the global trajectory can be seen as: *(Start) - (Go on lane Sector 1) - (Turn right at Event 1) - (Go on lane Sector 2) - (Stop at the Bus stop 1) - (Go on lane Sector 3) - ... - (Go straight at Event 2) - (Go on lane Sector n) - ... - (Stop at the terminal)*.

4 Intermediate trajectory

The intermediate trajectory is a zoom on hundreds of meters on the global trajectory which counts from the vehicle position. This intermediate trajectory provides the vehicle all preliminary actions to be performed and will now be detailed.

Figure 2 is a visualisation of the three trajectory levels where the intermediate trajectory is represented by the orange path.

4.1 Handling events

Considering the instruction given in the Section 3.2. Between any two events, the intermediate trajectory computation algorithm translates the guidance as a more detailed sequence of instructions: *(Go on lane to the next event) - (Prepare for the next event) - (Do actions at the event)*. This process is repeated once the vehicle has finished the actions at the current event. The vehicle will always be in the state *(Go on lane)* while the vehicle is navigating between two events. When there are two successive events and little preparation time to handle the next event after dealing with the first event, the vehicle should prepare the preliminary actions for the two events instead of only one. The medium term trajectory is calculated from the list of events on the global trajectory and the position of the vehicle. The next event is the closest one so that the distance from the vehicle to events in hundreds of meters will be taken into consideration.

4.2 The next event to deal with

The next event is the event right ahead of the vehicle. The vehicle has to run along its current road to approach the next event. In approaching the next event the vehicle has to prepare actions to get through the event.

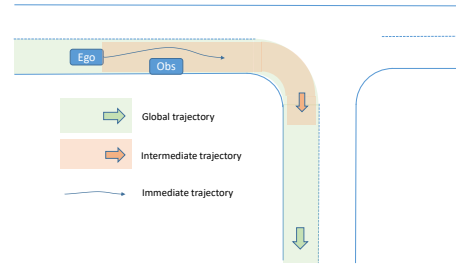


Fig. 2. A visualisation of the three level trajectories of an autonomous vehicle *Ego* with an obstacle *Obs* along its path.

4.3 Action planning for an event

Each type of event requires that the vehicle has to do a list of actions. For example, to turn right at an intersection without traffic light, the vehicle needs to turn on the right indicator when it is approaching the intersection. Then it has to reduce its speed. In entering the intersection, the vehicle has to give the way to the others which are already in the intersection. If there is a light at the entrance of the intersection, in addition to other preparing actions, the vehicle has to respect the light state. The vehicle then has to get in the intersection and to look for the corresponding exit. Once the vehicle exits the intersection, it has to turn off the indicator and has to continue on the road. From the example above, it is required that once the next event has been defined by the vehicle, the list of actions which corresponds to the event needs to be loaded. Based on the position of the vehicle, the actions of this list have to be executed in the defined order.

5 Immediate trajectory

Objective of this section is to define reference speed and reference steering for the vehicle

5.1 Vehicle speed reference

The speed reference of a vehicle is the speed that the vehicle must achieve. The speed reference is computed as:

$$V_{ref}(t+1) = f_v(V_{veh}(t), dyn_{veh}(t), dyn_{targ}(t), nextEvent, constrains) \quad (1)$$

Where:

- $V_{ref}(t+1)$ - The speed reference at time $(t+1)$
- $V_{veh}(t)$ - The speed of the vehicle at time t
- $dyn_{veh}(t)$ - The dynamics (other than the speed) of the vehicle at time t
- $dyn_{targ}(t)$ - The dynamics of the targets at time t
- $nextEvent$ - The next event from the vehicle position

constrains - The constrains on the road such as: authorised speed, the state of the traffic lights

Since the calculation of the speed reference is based on the actual dynamics of the vehicle, the smooth change of its value between the two consecutive calculation steps is insured. Moreover, the error of the vehicle speed controller at the last time steps will be taken into account.

5.2 Vehicle steering angle reference

The control of the vehicle steering is more complex than the control of the vehicle speed because of the complexity of the vehicle lateral dynamics. Moreover, the vehicle lateral control is affected by lots of internal and external disturbances such as the load on the vehicle, under-inflated tire(s), road inclination, road bend, lateral wind ... Thus, the vehicle lateral control is normally a combination of the angle (position) controller and a torque controller. The reference steering angle applied to this work is the angle that the vehicle must keep or achieve until the next calculation time.

Equation 2 shows the strategy of steering reference calculation.

$$\theta_{ref}(t+1) = f_s(\theta_{veh}(t), dyn_{veh}(t), dyn_{targ}(t), trq_s(t), \text{road}, \text{nextEvent}, \text{constrains}) \quad (2)$$

Where:

$\theta_{ref}(t+1)$ - Steering angle reference calculated for the time $(t+1)$

$\theta_{veh}(t)$ - The steering angle of the vehicle at time t

$dyn_{veh}(t)$ - The dynamics (other than the steering angle) of the vehicle at time t

$trq_s(t)$ - The measured torque at the steering system at time t

road - The detection of the road lines or road limit in front of the vehicle t

$dyn_{targ}(t)$ - The dynamics of the targets at time t

nextEvent - The next event to be dealt with on the vehicle trajectory

constrains - The constrains on the steering of the vehicle

6 Simulation

6.1 Scenario

A predefined road has been chosen as the global trajectory for the autonomous vehicle. On the trajectory, the autonomous vehicle has to deal with different traffic situations as well as to deal with the road signalisation. The mission of the autonomous vehicle is to start from any position on the trajectory then run toward the next bus stop. It has to stop at the bus stop for serving the passengers. Then it continue to get to the next bus stop.

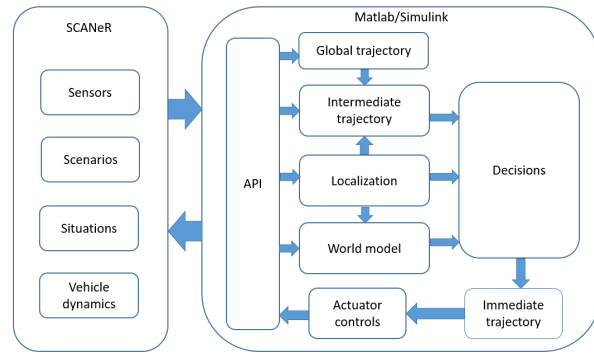


Fig. 3. The configuration of the system simulation

6.2 Situations

In our simulation, the followings are some of the situations which have been taken into consideration:

- Pedestrians crossing and walking along the road
- Vehicles stopping on the road
- Vehicles stopping on the road and there is another vehicles coming on the other lane
- Traffic lights and pedestrian crossing just before the bus stop

6.3 Simulation configuration

The simulation utilises SCANeR and Matlab/Simulink. The list of events on the global trajectory has been recorded and stored in Matlab/Simulink. This data is then passed to the autonomous driving model. A model Callas of SCANeR is used as the dynamic model of the vehicle to be controlled. The structure of the system simulation can be seen in Figure 3.

6.4 Simulation results

In our simulation, the vehicle calculates the next event that it has to deal with. By analysing the obligations at this event, the vehicle loads the necessary actions in approaching the event. In parallel with these actions, the vehicle calculated its immediate trajectory to insure the safety and that it does not break the law. The vehicle can handle simultaneously different situations such as: the traffic light, pedestrian crossing, pedestrians while entering a bus stop; vehicle coming in front while doing obstacle avoidance; pedestrians on the road and so on. The results are shown by some figures which are taken from the related videos. These figures represent the top view of the scene and the view of the integrated front camera. The Figure 4 shows the situation that there is an obstacle (the black car) on the road so that the autonomous vehicle has to take the left lane to avoid this obstacle. However, the distance to the bus which is coming on the left lane does not allow the autonomous vehicle to do the obstacle avoidance. The vehicle takes the decision to stop behind the obstacle and to wait for the clearance road. The situation shown in Figure 5 is more complex. There is a traffic



Fig. 4. Give way before doing the obstacle avoidance



Fig. 5. Give way to pedestrians while preparing for the traffic light and entering the bus stop

light right before the bus stop. Some pedestrians are walking not only on the crossing but before the light. Our autonomous vehicle deals well with the situation. It turns the right indicator on, slows down to ready to stop at the light, gives way to the pedestrian and starts to get close to the pavement.

7 Conclusion

The use of the three levels representation of the trajectory has been proved to be one appropriate option for autonomous transportation systems. The simulation, which is presented in this work, has demonstrated the efficiency of exploiting the intermediate and immediate trajectory. The computation of the global trajectory, in general, is not yet implemented in the simulation results, instead we chose a single bus line as the global trajectory. Nevertheless, the global trajectory is necessary in situations such as one road section becomes closed or to define the trajectory for a taxi mission or even for a delivery service. This task will be done in our future work.

References

[1] Wang, Z., 2015. "Optimisation avancée pour la recherche et la composition des itinéraires co-

modaux au profit des clients de transport". Theses, Ecole Centrale de Lille, Dec.

- [2] Gong, M., Chen, K., Zhao, K., and Chang, P., 2016. "Trajectory planning based on multi-objective optimization for redundant robots". In 2016 3rd International Conference on Systems and Informatics (ICSAI), pp. 133–138.
- [3] Alia, C., Gilles, T., Reine, T., and Ali, C., 2015. "Local trajectory planning and tracking of autonomous vehicles, using clothoid tentacles method". In 2015 IEEE Intelligent Vehicles Symposium (IV), pp. 674–679.
- [4] Leonard, J., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., Fletcher, L., Frazzoli, E., Huang, A., Karaman, S., et al., 2008. "A perception-driven autonomous urban vehicle". *Journal of Field Robotics*, **25**(10), pp. 727–774.
- [5] Glaser, S., Vanholme, B., Mammar, S., Gruyer, D., and Nouveliere, L., 2010. "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction". *IEEE Transactions on intelligent transportation systems*, **11**(3), pp. 589–606.
- [6] Marín, P., Hussein, A., Martín Gómez, D., and de la Escalera, A., 2018. "Global and local path planning study in a ros-based research platform for autonomous vehicles". *Journal of Advanced Transportation*, **2018**, 02, pp. 1–10.
- [7] Zhang, F., Xia, R., and Chen, X., 2020. "An optimal trajectory planning algorithm for autonomous trucks: Architecture, algorithm, and experiment". *International Journal of Advanced Robotic Systems*, **17**(2), p. 1729881420909603.