Optimization for Robot Motion Planning and Control – report2

Replication study: Temporal vs. spatial formulation of autonomous overtaking algorithms

Introduction:

This report presents a replication of the research conducted in the paper "Temporal vs. spatial formulation of autonomous overtaking algorithms". The objective of this study was to reproduce the methodology presented in the original paper.

Brief description of the paper:

This paper compares two formulations in autonomous overtaking algorithms: temporal (time domain in s) and spatial (distance domain in m). The temporal formulation is the most commonly used while the spatial formulation discussed here is actually introduced by another paper: "Predictive cruise control with autonomous overtaking". It is said that the spatial formulation allows a convexification and linearization of the autonomous overtaking.

The Scenario:

The scenario is quite simple with a Leading vehicle going at a speed $v_L = 50km/h$ and the Ego vehicle that will do the overtaking going either at a constant speed $v_E = 70km/h$ or at a slower initial velocity and accelerating to reach that velocity during the overtaking.

Methodology:

I was not able to get from the paper the method they used for their optimization so I was used the code I previously made when replicating another paper: "Optimal Trajectory Planning for Autonomous Driving with Logical Constraints: An MIQP Perspective" which uses big-M method and Mixed Integer Quadratic programming. I then created a function called "time_domain()", just replacing the values and cost function according to the scenario described above. The cost function is the following:

$$\int_0^{t_f} \Big(w_y (y_{\mathrm{E}}(t) - y_r(t))^2 + w_v v_{\mathrm{E}y}(t)^2 + w_a \dot{v}_{\mathrm{E}y}(t)^2 + w_j \ddot{v}_{\mathrm{E}y}(t)^2 \Big) dt. \tag{14}$$

I was not able to find the exact values for w_y, w_v, w_a and w_j so I tried to fine tune it the best I could. I also had to add a cost function so the vehicle moves forward in the x direction and added to this cost function the cost in the x direction (I do not know how they did it in the paper). This part was quite basic, and an animated graph can be seen when we run the py file.

Then, the hardest part was understanding how to get from the time domain to the spatial domain. Here is my basic understanding in simple words of this new formulation from the paper:

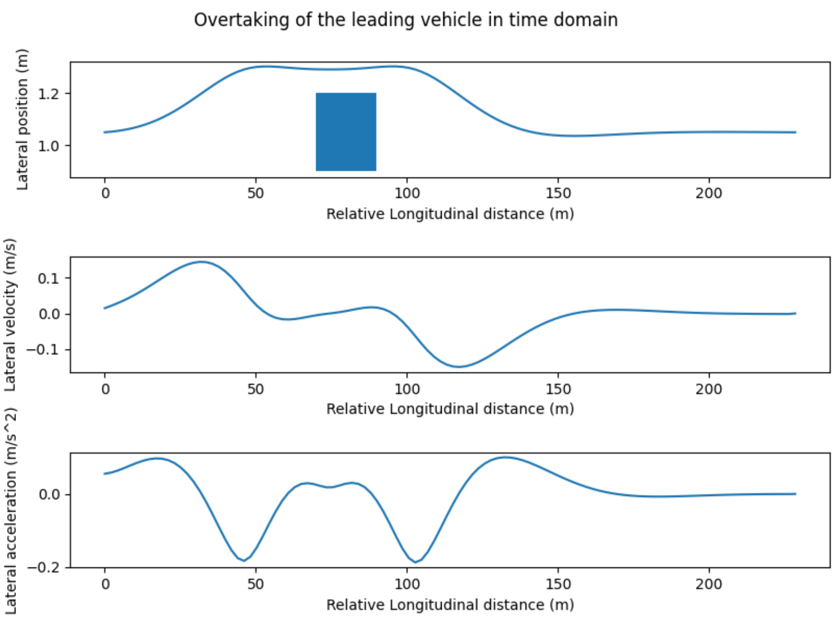
Instead of the time domain where both vehicle moves, we use a moving reference frame going at $-v_L$ m/s with its starting point to be the same as the ego vehicle starting point (0). We can now see the situation as frames taken from a camera where the leading vehicle is at a constant position while the ego vehicle moves forward at a speed approximately of $v_E - v_L$. Each 'frame' is taken with a constant time incrementation (can be considered as FPS). The paper provides us with the total time taken for the overtaking and is $t = 27s$. The number of frames were not given, I went with 110 frames.

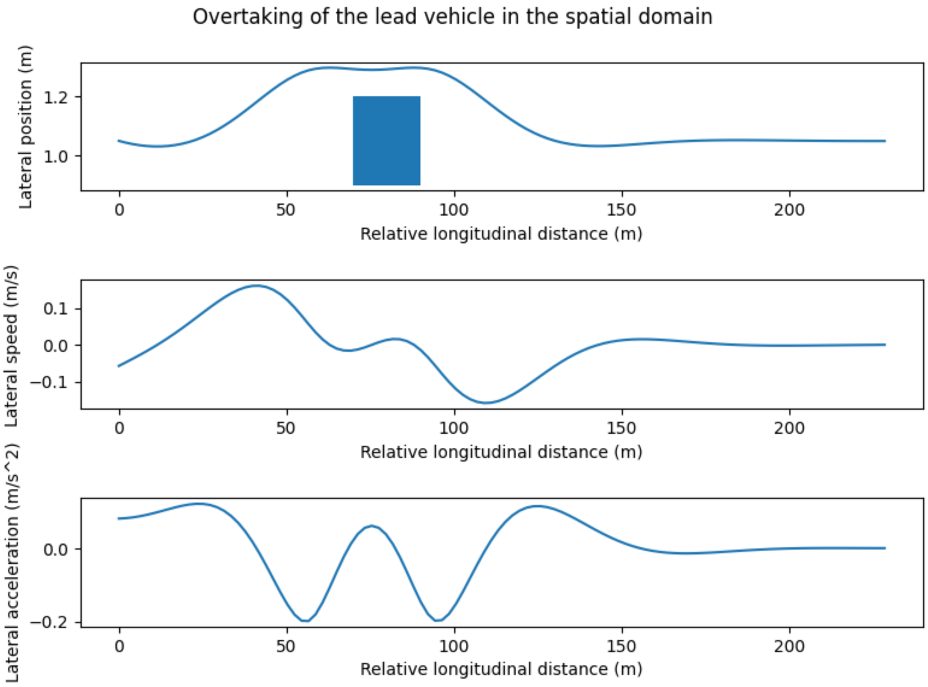
Now, since we are no longer in the time domain, when calculating lateral velocity in the new reference frame, we apply Chain's rule: $v_y(x) = \frac{\partial y}{\partial x}(x) = \frac{\partial y}{\partial t}\frac{\partial t}{\partial x}(x)$

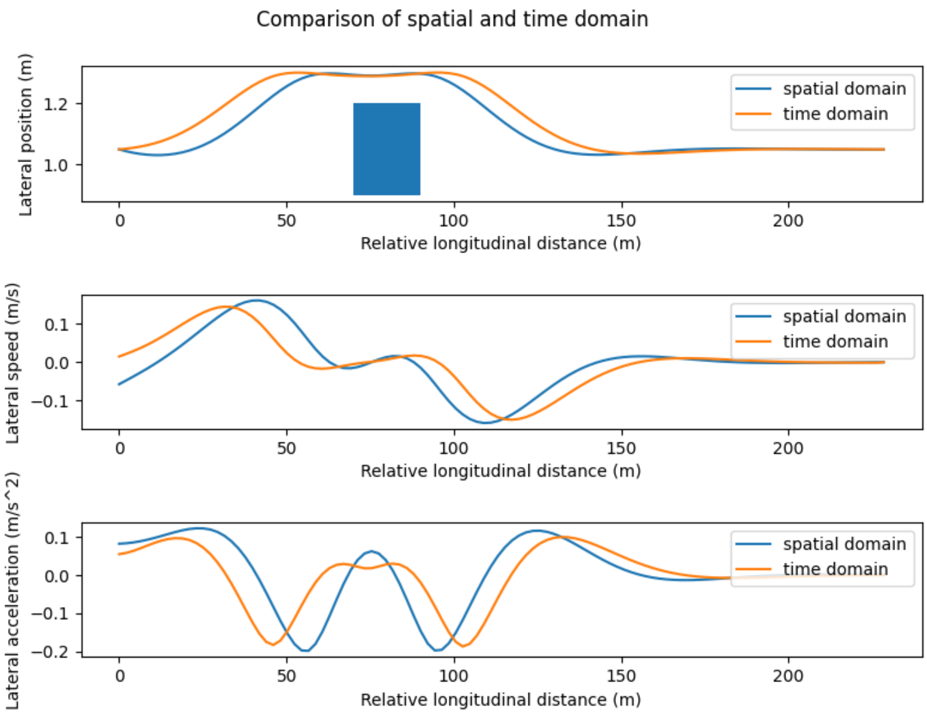
With this, I used the similar optimization method than the time domain but changed the formulation as explained before. The weights were also slightly modified since in the spatial domain we are not concerned with the x displacement (since it becomes the reference frame).

Results:

Here are the results I was able to obtain:







It is important to note that the result from the time domain were adapted to the relative longitudinal distance from the spatial domain. To view a better representation of the time domain, there is a short animation when you launch the code provided with this report. Finally, we can compare with the results from the paper:

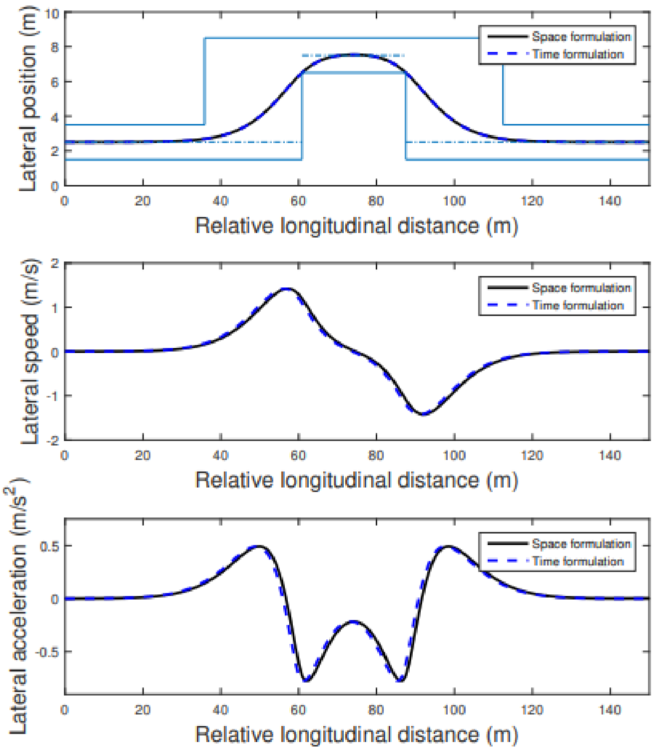


Fig. 2. Scenario where the ego vehicle with a fixed velocity is overtaking a leading vehicle on a road with two lanes. The center of gravity of the ego vehicle is allowed to reside between the limits $y_{\min}$ and $y_{\max}$, depicted by the solid lines. The dash-dotted line is the reference position of the ego vehicle.

We can see that in their case, the two gave really similar result while mine did not. But this is due to the weight being different and more importantly their method not being disclosed or at least not precisely enough so I could catch it and reproduce it.

Nevertheless, we can see that my program works and the solver is faster with the spatial formulation than the temporal one which makes it an interesting approach to autonomous overtaking.

Finally, taking less frames (approx. 90 or 80) will give a result closer to the one obtained in the paper for spatial formulation but further for the temporal one.