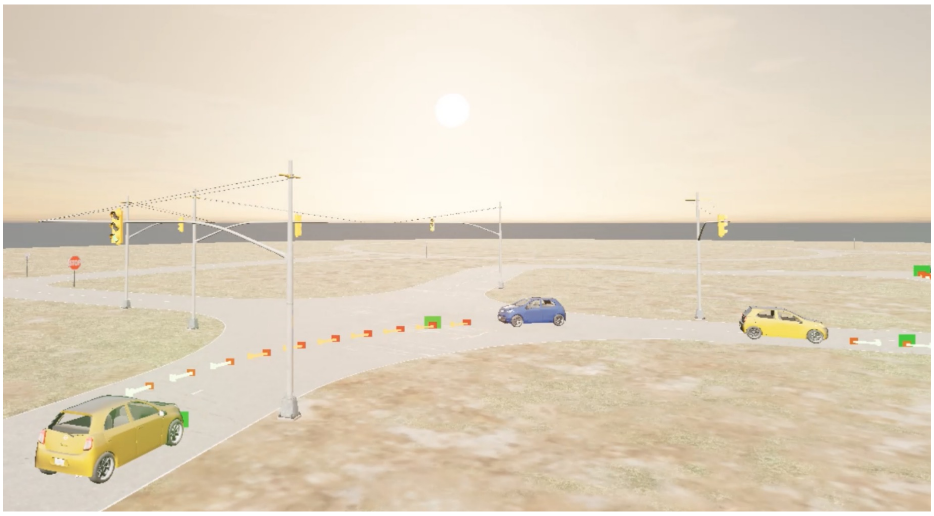
TX n°7491
Control and decision-making architecture for a fleet of shared autonomous vehicles in a dense urban environment

MOUGOUE Ruben
GUEGAN Theo



Summary

1 Introduction

The need for efficient, safe, and sustainable public transportation has never been greater in today's rapidly urbanising world. By 2050, nearly 70 per cent of the global population is projected to reside in urban areas, creating unprecedented challenges for urban mobility. In response, our team within the European UTAC Future Vehicle Challenge is dedicated to revolutionizing autonomous vehicle navigation in dense urban environments.

1.1 Transforming Urban Transport : The Need for Innovation

Urban congestion costs European economies over €100 billion annually, with commuters spending an average of 100 hours per year stuck in traffic. Traditional public transport systems struggle to keep pace with growing urban populations and the increasing use of personal mobility devices like electric scooters. Our project addresses these issues head-on, proposing a novel solution to enhance public urban transport through advanced autonomous vehicle technology.

1.2 Our Key Innovations

Our group's task is to develop and implement a cutting-edge control and decision-making architecture that ensures safe, efficient, and comfortable navigation for autonomous vehicles (AVs) in dense urban settings. Our approach includes the following key innovations :

1. Target navigation (based on a waypoint given by the Global Planning team) instead of trajectory planning (which provides a more flexible, reliable and real-time adaptative control, as we will discuss in this report).
2. Adaptative Cruise Control (ACC) based on real-time vehicle offset.
3. Static obstacle avoidance, performed by creating new target in real-time following limited-cycles vector field.

By addressing the challenges of autonomous vehicle navigation in dense urban environments, our work paves the way for safer, more efficient, and reliable urban mobility. We are not only enhancing the current state of public urban transport but also setting the foundation for the future of autonomous transportation, where intelligent vehicles seamlessly integrate into the urban fabric, providing a reliable and efficient mode of transport for all.

2 Model and specifications of our vehicle

2.1 Vehicle Model

For simplification purposes and to converge with what is globally done in autonomous vehicle research, we went on a tricycle vehicle model (it is also easier to implement our path-following strategy (based on waypoints) with a tricycle model). Our experimental vehicle is devoted to urban transportation. As mentioned earlier, our vehicle modelling is performed relying on a tricycle model as shown below :

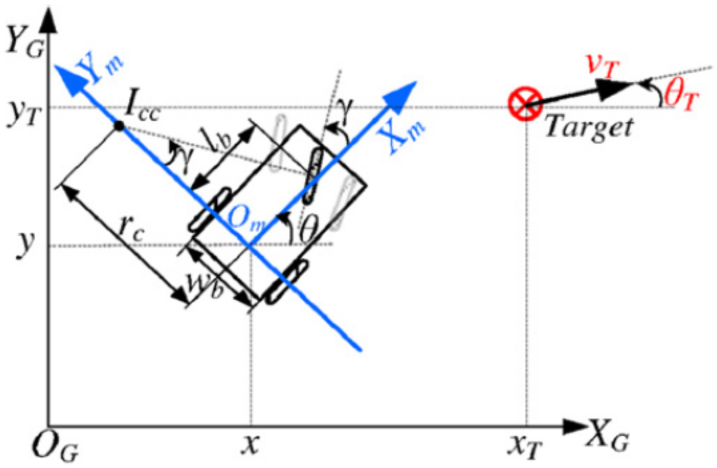


FIGURE 1 – Vehicle and target configuration in Global (X_G, Y_G) and Local (X_m, Y_m) reference frames - Extract from article [2].

$$\begin{cases} \dot{x} = v\cos(\theta) \\ \dot{y} = v\sin(\theta) \\ \dot{\theta} = v\cos(\gamma)/l_b \end{cases} \tag{1}$$

where O_G and O_m are respectively the origins of the global and local reference frames, (x, y, θ) is the pose (configuration state) at vehicle point O_m, γ is the orientation of the vehicle front wheels, and v is the linear velocity at vehicle point O_m. l_b and w_b are respectively the wheelbase and the track width of the vehicle (cf. Fig. 1). I_{cc} is the instantaneous centre of curvature of the vehicle trajectory. The radius of curvature r_c is given by :
$r_c = l_b/\tan(\gamma)$ and
$\mathrm{cc} = 1/r_c$ is the curvature of the vehicle trajectory.
Let us consider a dynamic target modelled as a point with non holonomic constraints (cf. Fig. 1).

This model allows us to use the general model of robot motion and also to simplify the controller equations. Its kinematic characteristics are given by :

$$\begin{cases} \dot{x_T} = v_T \cos(\theta_T) \\ \dot{y_T} = v_T \sin(\theta_T) \\ \dot{\theta}_T = (\omega_T) \end{cases} \tag{2}$$

where v_T and ω_T are respectively the linear and angular velocities of the target. The radius of curvature is computed by
$r_{\mathrm{cT}} = v_T/\omega_T$.
An important consideration for target reaching is $v_T \leq v_{\max}$ and $r_{\mathrm{cT}} \geq r_{\mathrm{cmin}}$, where $v_{\max}$ and r_{cmin} are respectively the maximum linear velocity and the minimum radius of curvature of the vehicle, given by
$r_{\mathrm{cmin}} = l_b/\tan(\gamma_{\max})$.
For static target reaching (point stabilization, i.e., to reach a specific point with a given orientation), ω_T is considered equal to zero and v_T is not necessarily equal to zero.
v_T is then considered as a desired velocity value for the vehicle when it reaches the desired target (x_T, y_T, θ_T).

2.2 Intrinsic characteristics of our vehicle

Our vehicle is capable of :

- Following a specified path based on a waypoint navigation strategy (target navigation instead of trajectory planning) ;
- Respect a certain security distance when following another vehicle in traffic ;
- Avoiding static obstacles.

3 Waypoint navigation method

3.1 Explanation of the waypoint navigation

Waypoint selection consists of obtaining the minimum number of points (waypoints) on the road to be successively reached by the vehicle to perform safe navigation.

These waypoints are selected considering a safe position on the road (as far as possible from the road limits) and the reliability of the obtained vehicle trajectory (smooth changes between the successive points).

To provide a complete framework to achieve the navigation strategy, we will consider the existence of a defined trajectory (infinite number of points) ; the method aims to select an appropriate number of points (waypoints).

The reference path can in our context be obtained using a recorded vehicle trajectory handed to us by the Global Planning (TX3). Different criteria can be considered to obtain the minimum number of straight lines that closely fit the reference path. Criteria such as the Euclidean or curvilinear distance, orientation, or radius of curvature between waypoints can be used to fix the desired waypoints on the path.

The discretized reference path r is composed of sorted position $r_i = (x_{\text{ri}}\ ,\ y_{\text{ri}}\ )$ and its tangent orientation θ_{ri}.

The minimum number of straight line segments over the defined path is then computed while considering a constant threshold $\Delta_{\alpha\text{max}}$ for the orientation variation of the path Δ_α (cf. Algorithm 2).

Algorithm 2 Waypoint selection based on existing reference path

Require: Reference path $\mathbf{r} = (\mathbf{x}_r, \mathbf{y}_r)$ and $\Delta_{\alpha max} \in \mathbb{R}^+$
Ensure: Set of waypoints S_p
Init $j = 0$, $r_{w_j} = r_0$ (initial position of **r**) and $\theta_{w_j} = \theta_{r_0}$ (tangent of the point along trajectory **r**)
for $r_i \in \mathbf{r}$ (sorted set of trajectory points) **do**
  Compute $\Delta_\alpha = |\theta_{r_i} - \theta_{w_j}|$
  if $\Delta_\alpha \geq \Delta_{\alpha max}$ **then**
    j = j + 1
    Set $r_{w_j} = r_i$ and $\theta_{w_j} = \theta_{r_i}$
    Add $w_j(r_{w_j}, \theta_{w_j})$ to S_p
  end if
end for

FIGURE 2 – Waypoint selection based on existing reference path - Extract from article [2].

Figure 3 shows one vehicle trajectory and the obtained waypoints using Algorithm 2 with $\Delta_{\alpha\max} = 5°$, $15°$ and $30°$ respectively. The switch between waypoints is smoother, with a small value of $\Delta_{\alpha\max}$.

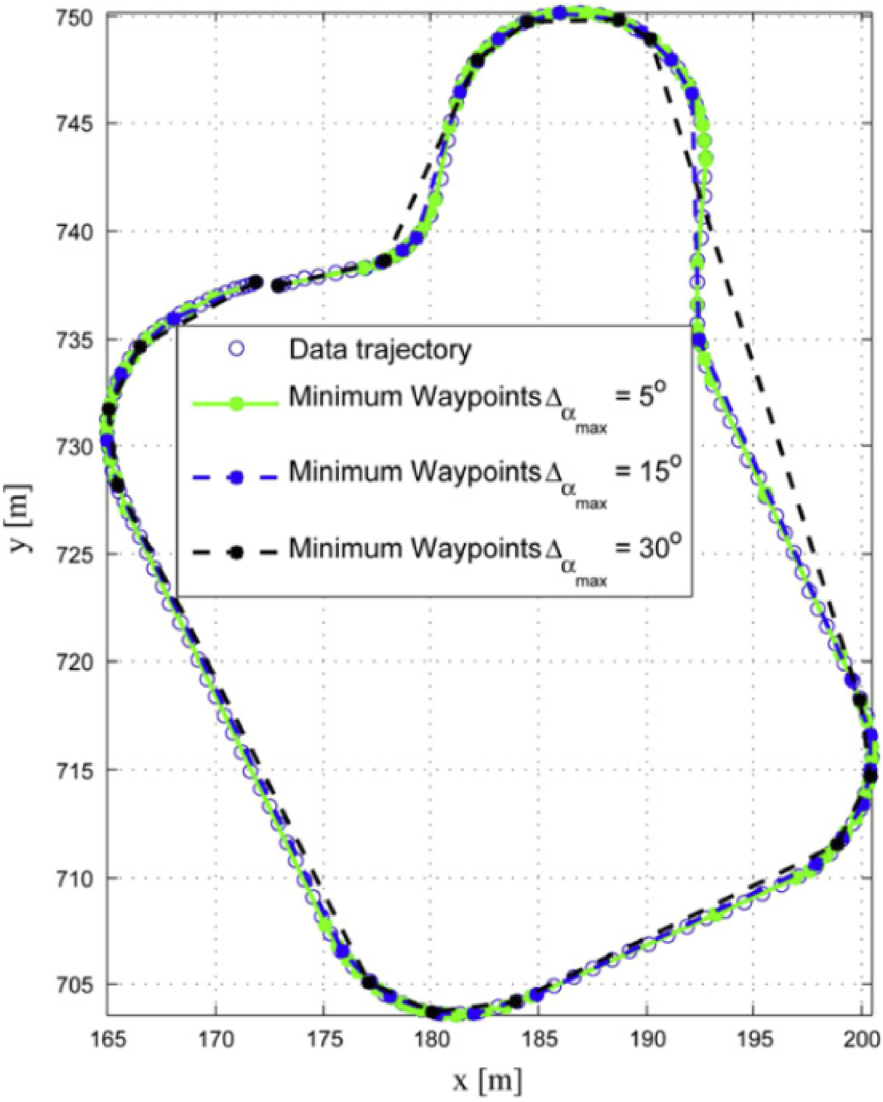


FIGURE 3 – Waypoint selection based on existing reference path.Waypoint selection based on existing reference path - Extract from article [2].

3.2 Reasons for the choice of this strategy

We went on a navigation strategy based on sequential target (waypoints) assignment instead of trajectory planning (The computation of a time-parametrized path while taking into account different vehicle constraints and environment characteristics is time-consuming).

Different algorithms that compute a safe path (without temporal reference) require less computational time but provide trajectories which do not ensure the safe navigation of the vehicle.) because it offers the following advantages :
Guarantees safe, flexible, robust/reliable and real-time adaptive navigation. With this strategy, we can :

- Achieve static (in the case of a predefined trajectory to follow) and dynamic (in the case where there is a leader and follower vehicle as it is when we implement an adaptive cruise control for our autonomous vehicle to follow another vehicle with a certain longitudinal security distance and speed) target reaching.
- Avoid static obstacles : to exhibit the flexibility of the proposed navigation strategy, a scenario with the presence of an obstacle is presented (cf. Fig. 4). An obstacle is placed between the waypoints. The proposed strategy can easily integrate obstacle avoidance behavior. Therefore, the vehicle can perform different maneuvers between waypoints, in this case, obstacle avoidance, without the use of any trajectory replanning method. The obstacle avoidance method is based on limit-cycles. It was selected because it is a stable and robust method which could use only local information from range sensors.

Let us briefly inspect this method. A limit cycle is a reactive, safe trajectory which encloses the hinder obstacle. According to that, the vehicle avoids the obstacle while tracking the direction of the limit-cycle trajectories. The obstacle avoidance is activated as soon as the vehicle detects at least one obstacle which can hinder the future vehicle movements toward the current assigned waypoint.

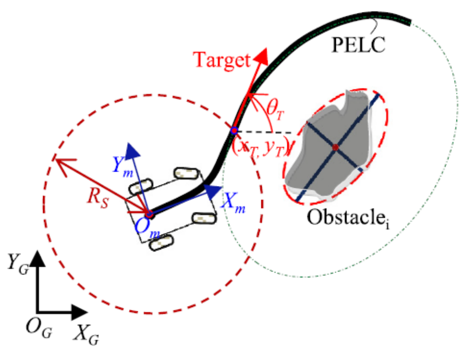


FIGURE 4 – Target set-points definition w.r.t. Euclidean distance.

4 Selection Algorithm

One of the key factors of our strategy is the selection algorithm that allows us to perform our navigation with **homogeneous set-points**. Our inputs are the pre-defined waypoints given by the Global Planning team but also other vehicles and static obstacles to create new waypoints in real-time with the same properties : coordinates, orientation and finally, a linear speed to reach the target.

Target $= (X_t, Y_t, \theta_t, V_t)$

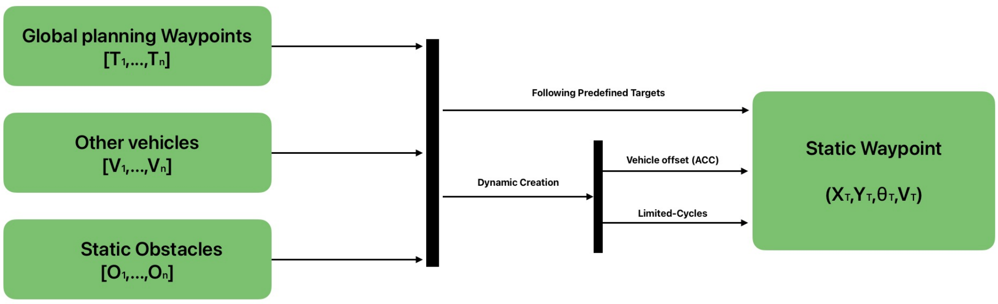


FIGURE 5 – Waypoint Selection.

4.1 Waypoint validation strategy

The error conditions (E_d and E_{angle}) are used to allow switching to the next target when the vehicle position enters a circle with a radius equal to E_{dis} and center $(X_{\text{Tj}}, Y_{\text{Tj}})$. Furthermore, we also use the transformation matrix to know if the target is already overcome ($x^{T_j} \geq 0$). The current target is updated with the following waypoint in the list, and the vehicle starts the movement to reach the new target. If the vehicle does not satisfy the error conditions, then the perpendicular line L_j which connects T_j and $T_{\text{j+1}}$ is used to reach the next target (cf. Fig. 6).

Algorithm 1 Sequential target assignment

Require: Vehicle pose, current target T_j and a set of N sorted waypoints
Ensure: Switch between target set-points
1: **if** (($d \leq E_{dis}$ **and** $e_\theta \leq E_{angle}$) **or** ($x^{T_j} \geq 0$)) { x^{T_j} is the coordinate of the vehicle in the local Target frame $X_{T_j} Y_{T_j}$ (cf. Fig. 8) } **then**
2: Switch from the current target T_j to the next sequential waypoint T_{j+1}
3: **end if**

FIGURE 6 – Target validation algorithm - Extract from article [2].

For computing the value of x^{T_j} and manage the activation and deactivation of the other modes, it is essential to define the transformation matrix that transits between different bases. This matrix will enable us to determine the position of our vehicle within the reference frame of a target, another vehicle, or an obstacle.

$$T_{O\to T} = \begin{pmatrix} \cos(\theta_{\text{tar}}) & -\sin(\theta_{\text{tar}}) & x_{\text{tar}} \\ \sin(\theta_{\text{tar}}) & \cos(\theta_{\text{tar}}) & y_{\text{tar}} \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} x_{\text{vehicle\_base\_target}} \\ y_{\text{vehicle\_base\_target}} \\ 1 \end{pmatrix} = T_{O\to T}^{-1} \begin{pmatrix} x_{\text{vehicle}} \\ y_{\text{vehicle}} \\ 1 \end{pmatrix} \tag{3}$$

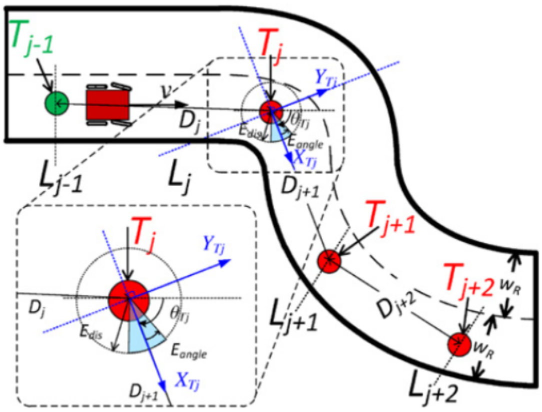


FIGURE 7 – Description of waypoints and target assignment - Extract from article [2].

4.2 Real-time adaptation strategy

1. First, for the Adaptive Cruise Control, we need to check if there is a vehicle ahead of us that is traveling in the same direction. We rely on a transformation matrix to calculate the position of the nearest vehicle in the reference frame of the current vehicle. If the x coordinate is positive, this indicates that the vehicle is ahead. If the vehicle is ahead and near enough, we check if the two vehicles are traveling in the same direction based on the difference in orientation of their next targets.

2. We also need to check for static obstacles to avoid on our trajectory. This mode is activated a few miles before the vehicle enters the circle of influence of the obstacle (this is implicit in obstacle detection). Specifically, this occurs when the radius of the circle of influence of the obstacle plus a smoothing parameter is greater than or equal to the distance from the robot to the obstacle. The smoothing parameter allows for early activation of obstacle avoidance, providing a safety margin. Finally, we deactivate this mode when the obstacle is passed, relying on another transformation matrix to calculate the position of the vehicle in the reference frame of the obstacle.

5 Navigation modes

5.1 ACC - Adaptive Cruise Control

Using this mode, we create a new target based on an offset from the vehicle ahead of us for the x and y coordinates. Additionally, we take the linear speed and orientation of the leading vehicle for other parameters.

1. The angle ϕ between the current vehicle and the leading vehicle is calculated. This can be done using the arctangent of the relative position :

$$\phi = \text{atan2}(y_{\text{lead}} - y_{\text{current}}, x_{\text{lead}} - x_{\text{current}})$$

2. The coordinates of the current vehicle are shifted based on the calculated angle. This involves creating a new target position offset by a certain distance d in the direction of the angle :

$$\begin{cases} x_{\text{target}} = x_{\text{lead}} - d\cos(\phi) \\ y_{\text{target}} = y_{\text{lead}} - d\sin(\phi) \end{cases} \tag{4}$$

3. With these offset coordinates, we are able to create the new waypoint :

$$\text{Target} = (x_{\text{target}}, y_{\text{target}}, \theta_{\text{lead}}, v_{\text{lead}})$$

This approach ensures that the vehicle can dynamically create static targets in real-time, providing a robust mechanism for adaptive cruise control. It enhances the vehicle's ability to follow the leading vehicle safely and efficiently, while also maintaining a safe distance by adapting its speed in real-time using the linear speed set points of the vehicle ahead.

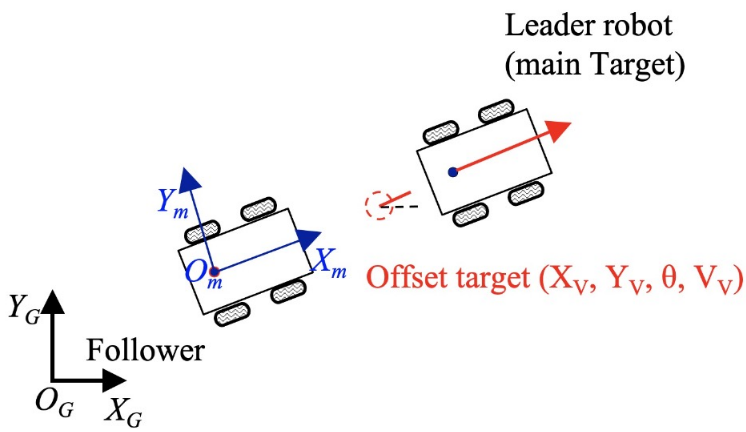


FIGURE 8 – New target based on vehicle offset.

5.2 Static obstacle avoidance

Target creation

To perform the obstacle avoidance behavior, the vehicle must accurately follow limit-cycle vector fields. These vector fields are given by the differential equation :

$$\begin{cases} \dot{X}_p = Y + X * \mu(R_{\text{lim}}^2 - X^2 - Y^2) \\ \dot{Y}_p = -X + Y * \mu(R_{\text{lim}}^2 - X^2 - Y^2) \end{cases} \tag{5}$$

where (X, Y) corresponds to the position of the vehicle according to the center of the convergence circle.
We only apply the clockwise trajectory motion because we drive on the right in France.

Two specific points are selected from the trajectory :

1. **Next Target Coordinates (x, y) :** The first point provides the coordinates of the next target that the vehicle should reach to effectively navigate around the obstacle $(x_{\text{target}}, y_{\text{target}})$.
2. **Orientation (θ) :** The second point is used to determine the orientation (θ) of the next target. This is achieved by calculating the tangent to the trajectory at this point, ensuring the vehicle's heading aligns with the trajectory.

$$\theta_{\text{target}} = atan2(y_{\text{trajectory}}, x_{\text{trajectory}})$$

Speed Determination : The speed to reach the next target is set based on the radius of influence of the obstacle, optimizing for both safety and efficiency.

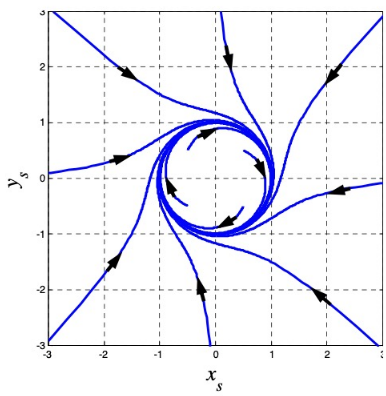


FIGURE 9 – Illustration of limit-cycle vector fields generation surrounding an obstacle.

Target cleaning

When the obstacle avoidance mode is activated, we also need to ensure that the targets previously defined by the Global Planning team are deleted. Otherwise, when the mode is deactivated, the vehicle will attempt to return to validate waypoints that are within the radius of influence of the obstacle.

5.3 Control law

Thanks to our previous algorithm, our set-points are homogeneous, which means we can define and use the same control law for different situations such as navigation, obstacle avoidance, and Adaptive Cruise Control (ACC). Although, thanks to the target speed set-points, our command law provides the ability to adapt the vehicle's linear speed in real time based on different situations. This allows us to stop the vehicle, adjust the speed to maintain a safe distance from the vehicle ahead, and comply with traffic signals and safety regulations.

Details of the control law for adapting linear speed and front wheel orientation can be found in the article [2], which includes all calculations and a demonstration of asymptotic stability according to Lyapunov-based analysis.

$$\begin{aligned} v &= v_T \cos(e_\theta) + v_b \\ \gamma &= \arctan(l_b c_c) \end{aligned} \tag{6}$$

6 Results - Conclusion

Our team has made significant strides in the development of autonomous vehicle navigation in dense urban environments. The implementation of our real-time waypoint selection strategy has shown promising results in preliminary tests, demonstrating a higher degree of flexibility and adaptability compared to traditional trajectory planning methods.

Our Adaptive Cruise Control (ACC) system, based on dynamic target tracking, has also shown its effectiveness in maintaining safe distances from other vehicles and obstacles, contributing to the overall safety of our autonomous vehicle.

Furthermore, our static obstacle avoidance system, also based on dynamic target tracking, has proven to be effective in detecting and avoiding stationary obstacles, further enhancing the safety and reliability of our autonomous vehicle.

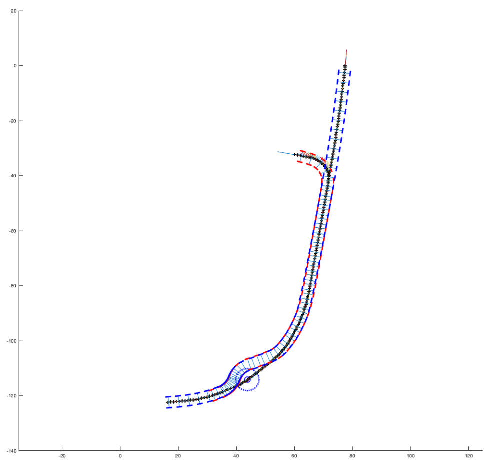


Figure 10 – Illustration of our final results in Matlab (without Carla).

The work of our team represents a significant step forward in the field of autonomous vehicle navigation in dense urban environments. Our innovative solutions not only address the current challenges of urban mobility but also lay the groundwork for the future of autonomous transportation.

By focusing on real-time adaptability and safety, our project contributes to the development of autonomous vehicles that can seamlessly integrate into the urban fabric, providing a reliable and efficient mode of transport for all.

While we have made significant progress, we recognize that there is still much work to be done. We look forward to continuing our research and development efforts, pushing the boundaries of what is possible in autonomous vehicle technology.

Task List

- Theo : Initial Proof of Concept (POC) using PID for linear speed and steering angle
  - First moving vehicle simulation in Matlab using Object-Oriented Programming (OOP)
- Theo : SY28 Transposition using OOP
  - Creating vehicle class for adaptation with Carla, utilizing inheritance
  - Bi-cycle vehicle model with static obstacle avoidance capability and linear corrector for speed and steering angle
- Theo : Final Code Utilizing Homogeneous Waypoints
  - Refactoring the vehicle class to adapt depending on the situation
  - Tricycle vehicle model with new control law for speed and front wheel orientation adaptation
  - Adaptive Cruise Control (ACC) using vehicle offset
  - Transposition of the previous obstacle avoidance trajectory generated by limit-cycles to work through dynamic waypoint creation
- Ruben and Theo : UTAC Presentation
  - Discussing our vision of the project and preparing presentation (50-50 %)
- Ruben and Theo : Final Report
  - 4. - 5. - 6. : Theo
  - 1. - 2. - 3. - 4. : Ruben

Bibliography

[1] M. Aicardi, G. Casalino, A. Bicchi, A. Balestrino, Closed loop steering of unicycle like vehicles via lyapunov techniques, Robotics Automation Magazine, IEEE 2 (1) (1995) 27–35.

[2] José Vilca, Lounis Adouane, Youcef Mezouar, A novel safe and flexible control strategy based on target reaching for the navigation of urban vehicles, Robotics and Autonomous Systems 70 (2015) 215–226.

[3] L. Adouane, “An adaptive multi-controller architecture for mobile robot navigation,” in 10th IAS, Intelligent Autonomous Systems, Baden- Baden, Germany, July 23-25 2008, pp. 342–347.