

Path Following for Autonomous Vehicle Navigation Based on Kinodynamic Control

Kristijan Maček¹, Roland Philippsen² and Roland Siegwart¹

¹Autonomous Systems Lab, Swiss Federal Institute of Technology Zurich, Switzerland

²Robotics and Artificial Intelligence Lab, Stanford University, USA

This paper addresses the problem of path following for mobile robots with particular emphasis on integrating the global path planning, path following and a collision avoidance scheme in a unified framework. Whereas the traditional path following algorithms aim at minimizing an error function with respect to a given path and kinematic and/or dynamic model of the robot, the problem of collision avoidance is often neglected or simply cast to the replanning phase of the global planner which issues the given path. Such approaches that do not check explicitly for collision for the given state of the ego-robot and the environment can easily lead to hazardous situations, in particular if latencies are present in the global path planning phase. In order to address obstacle avoidance directly, a navigation framework is presented here that combines a path following control scheme to attain a global objective with a collision checking scheme that incrementally builds collision-free trajectories, thus ensuring ego-robot safety at all times, with respect to the partially known static environment obstacles and kinodynamic limitations of the ego-robot itself. Two novel path following schemes are presented, namely the Traversability-anchored Dynamic Path Following (TADPF) and a combined TADPF-Sliding Mode Path Following (SMPF), based on a previously developed SMPF technique. The two path following schemes have been verified both in simulation and experimentally on a test Ackermann-like vehicle.

Keywords: path following, autonomous vehicle navigation, kinodynamic control, path planning

1. Introduction

The paper analyzes three distinctive path following strategies for global vehicle navigation. It assumes existence of a navigation function that generates an on-line path towards a global objective and optionally also generates traversability cost estimates for changing environments with limited vehicle's sensor range, such as E^*

([1]) or Field- D^* ([2]). A reference point path following technique by the authors was first presented in [4]. The follow-up TADPF controller was presented in ([3]) and modified here to account for driving comfort based on lateral acceleration constraints of the vehicle, which affects primarily the longitudinal velocity profile of the vehicle. A comparison is made on the path following precision and steering control effort with the SMPF controller presented in [5]. A novel combination of the TADPF controller and the SMPF controller for path following is proposed which enables collision-free navigation along the global path with directly taking into account the dynamic limits of the vehicle.

In Section 2 the overall navigation scheme for autonomous vehicle navigation is introduced. Section 3 describes the TADPF controller followed by a brief description of the SMPF controller in Section 4, whereas the two path following approaches are combined in Section 5. The simulation results and comparison of the path following strategies with references to experimental vehicle runs are given in Section 6, followed by conclusions in Section 7.

2. Navigation Structure

Figure 1 shows the overall scheme of the navigation system for autonomous vehicles. The Goal Manager (GM) handles the global scenario of the vehicle, which is described by a set of goals/waypoints to pass through. The global planner (GP) receives the occupancy information, i.e. occupied areas of the environment, with the map anchor (current vehicle

pose) from the Mapper module (MP) and dynamically replans the global path on-line. The versions of GP which are employed in our system and are able to handle dynamic environment changes are Field- D^* and E^* . The current

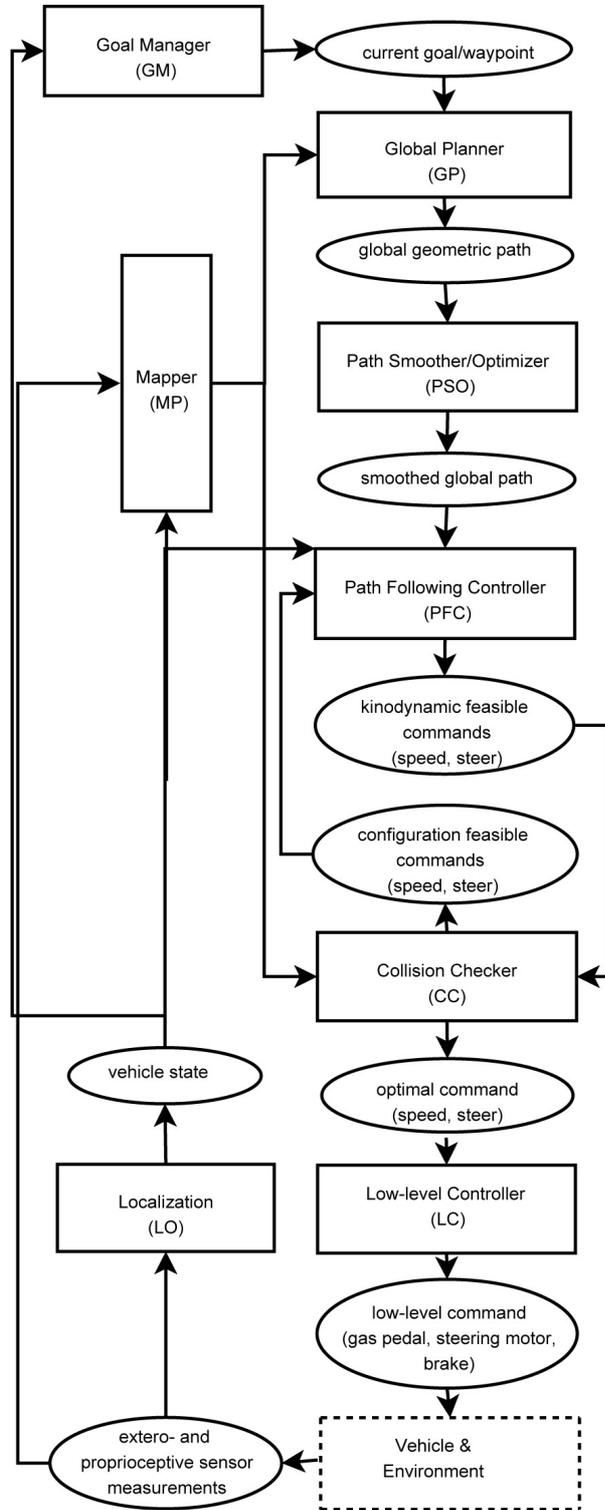


Figure 1. Overall navigation structure.

goal/waypoint is valid until the vehicle reaches a predefined vicinity around the goal, whereupon the next goal is taken and the GP restarted.

The dynamically changing global geometric path is computed at each control cycle and passed on to the Path Smoother/Optimizer (PSO) module, which in its basic version interpolates a smooth continuous path through the geometric path vertices, using cubic splines ([6]). This module can further optimize thus obtained smooth path by minimizing the curvature change (jerk) along the path, which can improve both driving comfort and quality of path following control.

The Path Following Controller (PFC) is employed to follow the smoothed global path. The path to follow can also be a holonomic one. The PFC uses the kinematic model with dynamic limitations of the vehicle, such as longitudinal acceleration and steering rate limits, to generate a set of kinodynamically feasible commands for each state of the vehicle, described by the pose $\{x, y, \theta\}$ and the kinematic state $\{v, \phi\}$, v and ϕ being the longitudinal velocity and the steering angle, respectively. Each kinodynamically feasible command pair within the set $\{v_i, \psi_j; i = 1, \dots, N_v, j = 1, \dots, N_\kappa\}$ represents a possible circular trajectory, if the commands are kept constant from the current control cycle onwards. Therefore, each possible trajectory is checked for collision against obstacles in the Collision Checker (CC) module, i.e. configuration space feasible trajectories. The CC module receives obstacle occupancy data directly from the MP module on-line. This renders the CC module independent of GP and ensures safe navigation even in situations when the GP is in a replanning phase and a collision-free global path is not available. From the set of kinodynamically and configuration space feasible commands/trajectories, the optimal command v^*, ϕ^* taken is the one that ensures path following (steering) and comfort driving by limiting the lateral acceleration (longitudinal velocity profile). The PFC framework which combines the above mentioned features is called TADPF-SMPF (“Traversability-anchored Dynamic Path Following - Sliding Mode Path Following”).

The low-level controller (LC) converts the kinematic level commands v^*, ϕ^* to the actuator signals on the gas pedal, brake (if in deceleration phase) and steering motor reference. The longitudinal velocity controller is based on a Fuzzy

logic controller, which handles also the acceleration/deceleration/cruising phases. The steering column controller is a PID implementation of power steering reference. The underlying vehicle state estimation for vehicle localization is done via Information Filter sensor fusion, available on the vehicle (see [7]).

3. Traversability-anchored Dynamic Path Following (TADPF) Controller

Global free-space connectivity for the vehicle navigation in this navigation scheme is obtained by a global planner. The global navigation function assigns traversability cost values for each free-space position, i.e. some areas are less traversable or even prohibited due to the presence of obstacles. Based on the traversability cost, a measure of cost to a global objective is obtained that takes into account the environment configuration and depends on a specific implementation of the global planner (see further [1] and [2]). By minimizing the cost to goal, the globally optimal geometric path can be obtained.

With the global navigation function available, the aim is to control the vehicle's motion such that it follows the global free-space connectivity based on direct traversability values and/or optimal geometric path computed. The approach of the TADPF controller that was first presented in [3] is recapitulated here, whereas the modifications done in this work are stressed further. The basic concept is in representing the current trajectory of the vehicle, given the low-level control on steering and longitudinal velocity as an arc trajectory, which would be exerted if the commands would be kept constant (see [8] for the first report on this concept). A set of feasible commands given the kinodynamic and configuration (obstacle) space constraints can be therefore represented as a set of feasible arc trajectories. At each control cycle, the optimal arc (command) can be chosen according to different criteria. Since in the presented controller combines both kinodynamically feasible trajectories with the traversability information of the global goal objective it is called "Traversability-Anchored Dynamic Path Following".

3.1. Kinodynamically Feasible Vehicle Trajectories

The description of the car motion is based on the Ackermann kinematic model:

$$\dot{x} = \cos \theta v_l, \quad \dot{y} = \sin \theta v_l, \quad \dot{\theta} = \frac{v_l}{L} \tan \phi, \quad (1)$$

with $\{x, y, \theta\}$ being the robot pose and $\{v_l, \phi\}$ the longitudinal velocity and steering angle as control inputs and L the axes distance of the front and rear wheels.

According to the Ackermann kinematics, the vehicle follows a circular path for a given kinematic level control input $\{v_l, \phi\}$. Therefore a set of arc vehicle trajectories can be defined as:

$$\mathcal{A} = \{a_{i,j} = \{x_{i,j}, y_{i,j}\}; i = 1 \dots N_v, j = 1 \dots N_\kappa\}, \quad (2)$$

where N_{v_l} denotes the number of arc sets due to longitudinal velocity $v_{l,i}$ discretization and N_κ the number of arcs due to curvature κ_j discretization, which corresponds to a steering angle ϕ_j .

At each control cycle a trajectory $a_{i,j}$ is chosen, corresponding to a control input $(v_{l,i}, \phi_j)$ that is feasible with respect to the environment constraints, e.g. obstacles and goal direction, but also according to the limitations on the vehicle motion itself. The kinematic limitations on the vehicle motion are the maximum longitudinal velocity $v_{l,max}$, the minimum allowed¹ vehicle speed $v_{l,min}$, and the maximum steering angle ϕ_{max} . The dynamic limitations are the maximum longitudinal acceleration $\dot{v}_{l,max}$ and the maximum steering rate $\dot{\phi}_{max}$. The aim here is to define a minimum set of arcs necessary to take into account the dynamic limitations of the vehicle at each time instant. From Eq. 1 it follows that:

$$\ddot{\theta} = \frac{v_l}{L \cos^2 \phi} \dot{\phi} + \frac{\dot{v}_l}{L} \tan \phi. \quad (3)$$

Typically, the low-level steering control loop (e.g. power steering) is faster than the longitudinal velocity control loop, thus for small time increments the longitudinal velocity can be considered constant with respect to the angular rate of the vehicle. Therefore, the second term can

¹ unless the goal is reached or during an emergency brake

be neglected in Eq.3. Given that the vehicle is currently on a trajectory defined by $\{v_l, \phi\}$, and the steering rate is at its maximum $\dot{\phi} = \dot{\phi}_{max}$, the curvature change within the kinematic level control sample time-step T_s can be expressed as:

$$\Delta\kappa(\phi) = \frac{v_l}{L \cos^2 \phi} \dot{\phi}_{max} T_s. \quad (4)$$

Taking the smallest curvature change within a control cycle T_s such that switching between neighboring arcs is feasible according to steering limitations leads to the a-priori number of arcs due to the curvature discretization:

$$N_\kappa = 2 \cdot \left\lceil \frac{\tan \phi_{max}}{L \Delta\kappa(\phi_{max})} \right\rceil + 1, \quad (5)$$

taking into account the central $\kappa(\phi = 0) = 0$ separately. Moreover, assuming the vehicle drives at its full longitudinal acceleration capability, the number of arc sets due to longitudinal velocity discretization is obtained as:

$$N_{v_l} = \left\lceil \frac{v_{l,max} - v_{l,min}}{v_{l,max} T_s} \right\rceil. \quad (6)$$

3.2. Configuration Space Feasible Vehicle Trajectories – Traversability and Obstacle Cost Criteria (Γ_t, Γ_o)

In order to chose an optimal vehicle trajectory-arc at each cycle T_s , an arc that is dynamically feasible is checked for potential collision with obstacles. The global navigation function provides the configuration space obstacle regions. If a prohibited node is encountered along an arc $a_{i,j}$ that is less than time $T_{b,i} = \frac{v_{l,i}}{v_{l,max}}$ away from the starting vehicle position, the arc is banned. For a prediction horizon T_h of vehicle motion along an arc $a_{i,j}$ with length $l(a_{i,j}) = v_{l,i} T_h$, the traversability cost $\Gamma_t^{(i,j)}$ and cost to obstacles $\Gamma_o^{(i,j)}$ are given as:

$$\Gamma_t^{(i,j)} = \sum_{t=1}^{N_{t,t}} r(a_{i,j}^{(t)}), \quad \Gamma_o^{(i,j)} = l(a_{i,j}) - l(a_{i,j,o}) \quad (7)$$

where $a_{i,j}^t$ corresponds to a sampled point on the arc $a_{i,j}$, $r(\cdot)$ the increment in the navigation function value (traversability and goal directedness cost at that point) and $N_{t,t}$ being the number

of discrete points up to the traversability prediction horizon $T_t \leq T_h$. The $l(a_{i,j,o})$ represents the length of the arc up to the first occurrence of an obstacle along it, with the condition being $l(a_{i,j,o}) \geq T_{b,i}$, i.e. the arc is still safe according to the dynamic breaking limitations of the vehicle. If an arc is completely void of obstacles, then $\Gamma_o^{(i,j)} = 0$. Traversability cost Γ_t gives essentially the global direction of the vehicle to steer to and the obstacle cost slows down the vehicle in presence of close obstacles.

3.3. Path Orientation Cost Criterion (Γ_r)

In Section 3.2, there was no geometric path to the goal position needed. However, by explicitly computing the path, additional information can be exploited. By following the negative gradient of the global navigation function from the current vehicle position $\{x, y\}$ to the goal $\{g_x, g_y\}$, a global reference path can be constructed that is a set of points $\{c_{r,d} = c_k; k = 1, \dots, N_g, c_1 = \{x, y\}, c_{N_g} = \{g_x, g_y\}\}$. This path is further smoothed by a spline technique to give a reference path $C_r(s)$ which is described with the curvilinear parameter s and curve gradient $\|C_r'(s)\| = \sqrt{p'^2(s) + q'^2(s)} \neq 0, \forall s \in [0, s_f]$, $p(s)$ and $q(s)$ denoting the x - and y -component of $C_r(s)$, respectively.

The kinematic level control objective in this case is to find a longitudinal velocity v_l and steering angle ϕ of the vehicle to follow the reference path C_r given by the global planner. In particular, a desired reference point is defined on $C_r(s_d)$ as:

$$x_d = p(s_d), \quad y_d = q(s_d), \quad (0 \leq s_d \leq s_f). \quad (8)$$

Determining the suitable position of the reference point is important for the path orientation cost of each arc. Here, it is proposed to set the curvilinear length of the reference point proportional to the current longitudinal velocity v_l of the robot with a time prediction horizon T_r (compare with [3]):

$$s_d = T_r v_l \quad (9)$$

and the orientation cost of each arc as:

$$\Gamma_r^{(i,j)} = \frac{\sum_{t=1}^{N_{r,t}} \|\theta_d - \theta_{i,j}(t)\|}{l(a_{i,j,r})} \quad (10)$$

where θ_d is the reference point orientation and $\theta_{i,j}(\iota)$ orientation of the vehicle in the ι -th point along the arc $a_{i,j}$. The cost is scaled to the cumulative length $l(a_{i,j,r}) = v_{l,i}T_r$, with the T_r being the orientation cost prediction horizon.

3.4. Optimal Command Choice with Comfort Criterion

The optimal steering commands ϕ^* chosen at each control cycle minimizes the total weighted sum cost:

$$\phi^* = \underset{\phi_{i^*j}}{\operatorname{argmin}} \left\{ \Gamma^{(i^*,j)} = \gamma_t \Gamma_t^{(i^*,j)} + \gamma_o \Gamma_o^{(i^*,j)} + \gamma_r \Gamma_r^{(i^*,j)} \right\}. \quad (11)$$

where each of the costs Γ_t , Γ_o and Γ_r are normalized before weighting. Although the combined cost Γ could be used also for the longitudinal velocity control, it is desired to impose additional constraints on its profile, due to the comfort of the drive, which is related to the maximum lateral acceleration along the path. In the perfect path following case, the vehicle's curvature would be equal to that of the path, so the maximum lateral acceleration for each $v_{l,i}$ is defined as:

$$a_{L,i,max} = \kappa_{max} v_{l,i}^2 \leq a_{L,max} \quad (12)$$

assuming constant movement along the path with curvature κ_{max} on the interval $s \in [0, s_\kappa]$, s_κ being the curvilinear path lookahead. Therefore, the set of feasible longitudinal velocities according to Section 3.1 and Section 3.2 is further constrained to a set of $\tilde{v}_{l,i}$ velocities based on Eq. 12. In order to minimize the travel time, the velocity chosen is:

$$v_l^* = \max \{ \tilde{v}_{l,i} \}. \quad (13)$$

4. Sliding Mode Path Following (SMPF) Controller

In [5] a sliding mode controller was developed for path tracking of Ackermann-like vehicles in order to address the issues such as fast response, good transient and robustness with respect to system uncertainties and external disturbances. The path following errors are described by y_e and θ_e , where y_e is the lateral distance from the vehicle reference point on the middle of the rear axis $\{x_r, y_r\}$, to the closest point $\{x_d, y_d\}$ on the

reference curve $C_r(s)$, denoted as virtual vehicle position. Angular error θ_e is the difference between the vehicle orientation θ_r and the tangent curve angle at the closest point θ_d , therefore the errors can be written as:

$$\begin{bmatrix} y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos \theta_d & \sin \theta_d \\ -\sin \theta_d & \cos \theta_d \end{bmatrix} \cdot \begin{bmatrix} x_r - x_d \\ y_r - y_d \\ \theta_r - \theta_d \end{bmatrix}. \quad (14)$$

Assuming the kinematics of the vehicle being the same as in Eq. 1, the corresponding error dynamics are:

$$\dot{y}_e = v_l \cdot \sin \theta_e, \quad \dot{\theta}_e = \dot{\theta}_r = \frac{v_l}{L} \tan \phi_r, \quad (15)$$

noting that $\dot{\theta}_d = 0$ in the path following problem.

In [5] a sliding surface was proposed which couples together the lateral error y_e and the angular error θ_e :

$$s = \dot{y}_e + k_1 \cdot y_e + k_0 \cdot \operatorname{sgn}(y_e) \cdot \theta_e. \quad (16)$$

The dynamics of the sliding surface is defined as:

$$\dot{s} = -Q \cdot s - P \cdot \operatorname{sgn}(s), \quad (17)$$

where Q and P are scalars.

The Eq. 16 can also be expressed as:

$$\dot{s} = v_l \cdot \dot{\theta}_e \cos \theta_e + k_1 \cdot v_l \cdot \sin \theta_e + k_0 \cdot \operatorname{sgn}(y_e) \cdot \dot{\theta}_e. \quad (18)$$

Combining Eq. 15, 17 and 18 yields the desired steering command:

$$\phi_c = \arctan \left(\frac{L}{v_l} \cdot \frac{-Qs - P \cdot \operatorname{sgn}(s) - k_1 \cdot v_l \cdot \sin \theta_e}{v_l \cdot \theta_e + k_0 \cdot \operatorname{sgn}(y_e)} \right). \quad (19)$$

It was proven in [5] by Lyapunov analysis that it suffices for the control law to be stable that $Q, P \geq 0$. Note also that the longitudinal velocity v_l in this case is not controlled, but is given as an input parameter to the control law of Eq. 19. It may be constant or variable according to a predefined velocity profile along the reference path $C_r(s)$.

5. Combined TADPF-SMPF Controller

Although the sliding mode controller in Section 4 provides a fast transient response and small path following error, it has two basic disadvantages. Firstly, it does not take explicitly into account the constraints on the control inputs, such as maximal steering angle ϕ_{max} , nor the dynamic constraints $v_{l,max}$ and ϕ_{max} as described in Section 3.1.

Secondly, there is no check for collision of the chosen vehicle trajectory arising from a given control input ϕ_c at v_l , as is done for the TADPF controller in Section 3.2. This effectively means that the vehicle may hit an obstacle or end in other potentially hazardous situation if there is a significant deviation from the pre-planned reference path, due to path following error, external disturbances or other non-modeled effects.

In order to resolve both these aspects and render the navigation of the vehicle safe while taking into account the control and dynamic limitations of the vehicle, the proposed new controller combines both TADPF and SMPF control strategies.

The longitudinal velocity v_l^* is given by Eq. 13 along the global path and the steering command ϕ_c is given by the SMPF controller of Eq. 19. In order to enforce the constraints of Section 3.1 and Section 3.2, the steering command at each instant is calculated as:

$$\phi_c^* = \operatorname{argmin}_{\phi_{c,j}} \{ \|\phi_{c,j} - \phi_c\| \} . \quad (20)$$

Thus, the optimal control chosen is defined by the arc of the TADPF controller $\{v_l^*, \phi_c^*\}$ that is also checked to be configuration space feasible and matches the closest the input command given by the SMPF control strategy $\{v_l, \phi_c\}$, resulting in the TADPF-SMPF control strategy.

6. Results

6.1. TADPF Controller

The path following schemes were tested in a simulation environment with polygonal obstacles. The vehicle scenario was to travel through a set of goal waypoints (global objectives) while avoiding any obstacles on the way. A lidar sensor is attached to the vehicle scanning in

the frontal horizontal plane with limited range. As new environment information is available, the global navigation function is recomputed by wavefront expansion from the current goal position. Based on the gradient information from the navigation function, a globally feasible path for the vehicle to follow is available on-line from any position in the free configuration space. Figure 2 shows a situation where the vehicle approaches a goal position and the wayfront is re-expanded in order to account for newly available obstacle information given by the lidar. In Figure 3 the situation is reversed, since the vehicle has all the information on the obstacle configuration available up to the goal position and the wavefront is retracting. In both Figures 2 and 3 right, a zoom-in on the vehicle and currently feasible trajectories (arcs) are given. Free trajectories are marked green, the ones that hit obstacles, but are still valid according to the dynamic breaking distance of the vehicle (shown on each arc with corresponding color points) are given in red and the trajectories along the re-propagating wayfront are given in magenta. In the latter case, the global gradient is not available for the re-propagation period, however, the traversability information (obstacles) is updated at all times, ensuring safety. The reference point position on the path is given in green.

Figures 4, 5, 6 show the steering comparison between the TADPF controllers where only traversability-obstacle cost is taken into account (Γ_t, Γ_o) versus reference path orientation cost controller (Γ_r) and the full controller with all three costs, when the vehicle is driven at a constant speed of $v_l = 20\text{km/h}$. As can be seen, both separate cost options give similar steering controls. The combined cost pattern depends on the weighting factors of each contribution, where a balanced weighting gives a smoothing effect on the net steering and trajectory.

In the next test drive, the vehicle was allowed to drive the speeds from $v_{l,min} = 10\text{km/h}$ up to $v_{l,max} = 30\text{km/h}$. When analyzing the longitudinal velocity of the traversability-obstacle cost in Figure 7 it can be seen that the speed is maximized due to the max. traversability change criteria (Eq. 7) with the longest arc distance. Slowing down is performed at times only due to the obstacle cost component. Although the safety criteria are thus satisfied, the

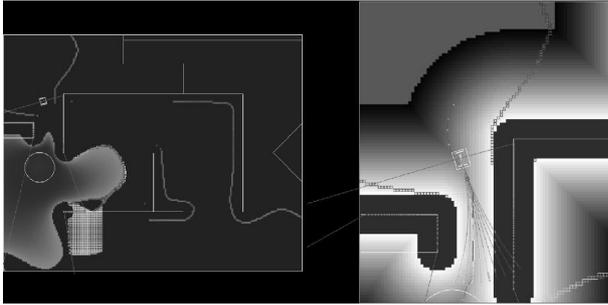


Figure 2. Global environment snapshot with local trajectory view (1).

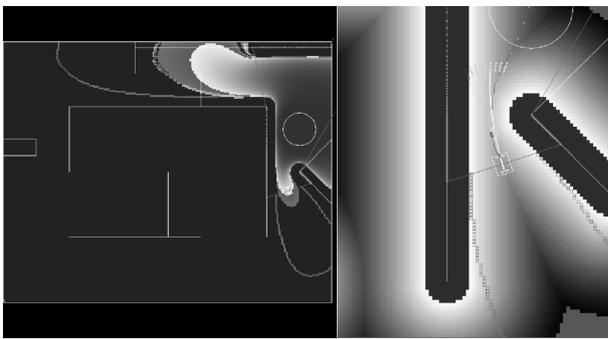


Figure 3. Global environment snapshot with local trajectory view (2).

Figure 10 shows extremely high lateral accelerations up to $a_L = 12m/s^2$, which is clearly unacceptable. By taking the lateral acceleration constraints into account from Eq. 12, the maximal v_l along the path can be limited as is shown for the case of the orientation cost profile, where maximum speeds attained are lower (see Figure 8), but the lateral acceleration a_L is drastically decreased. On average, the a_L stays well within the bounds $a_{L,max} = 1m/s^2$ for fairly comfortable driving ([9]) as in Figure 11.

The peak values of up to $a_L = 2.5m/s^2$ are due to abrupt change of path orientation and curvature beyond a goal waypoint, when a completely new path is replanned to the next goal. This situation can be improved by parallel planning of the next goal path, while the current one is still being executed and jointed to the next one smoothly at the current goal position, which is the topic of future work. Figure 9 shows the reference point position ρ which increases proportionally to the longitudinal speed. Figure 12 shows the current vehicle curvature and that of the closest point along the path for the vehicle and the reference point itself. In the perfect tracking case, the vehicle should have the same curvature as the reference point with

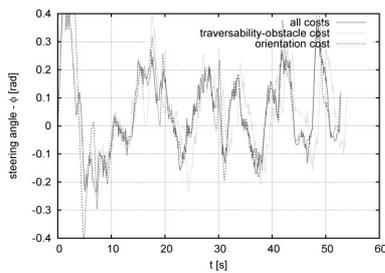


Figure 4. Steering angle ϕ comparison for TADPF.

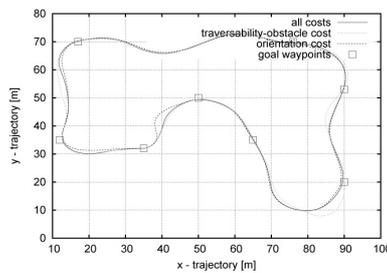


Figure 5. Trajectory x-y comparison for TADPF.

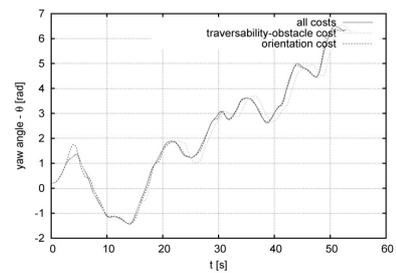


Figure 6. Orientation θ comparison for TADPF.

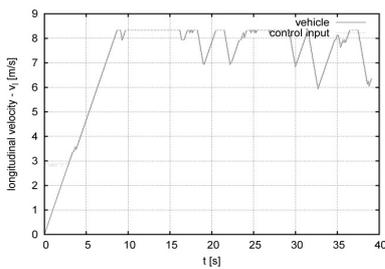


Figure 7. Velocity v_l for TADPF with Γ_l and Γ_ρ .

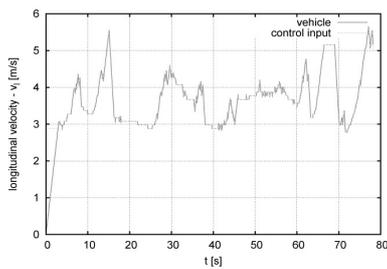


Figure 8. Velocity v_l for TADPF with Γ_r .

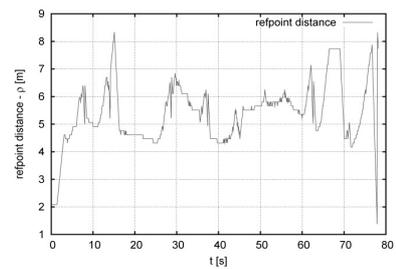


Figure 9. Reppoint distance ρ for TADPF with Γ_r .

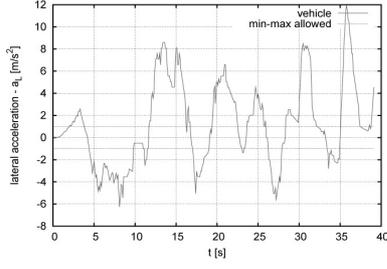


Figure 10. Lateral acceleration a_L for TADPF with Γ_l and Γ_o .

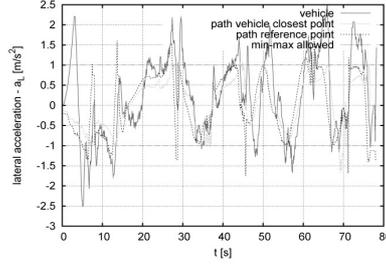


Figure 11. Lateral acceleration a_L for TADPF with Γ_r .

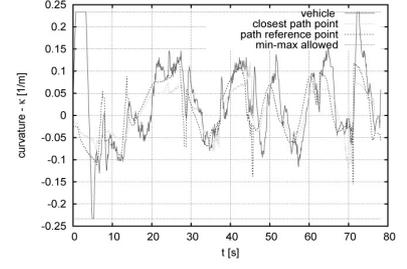


Figure 12. Curvature κ for TADPF with Γ_r .

a time delay related to the vehicle speed. Main differences can be observed due to initial misalignment of the vehicle orientation with respect to the path and the holonomicity of the path itself. Since the path is generated on-line and the vehicle motion is guaranteed to be safe due to the collision checks along currently chosen command/trajectory, small path following errors are acceptable. The velocity profile of Figure 8 has a quality of a time-optimal profile, in the sense that the vehicle is always driving with the maximum allowed speed according to the safety and comfort measure on lateral acceleration. A further improvement is foreseen in generating a smoother profile where the vehicle might take a longer time to traverse a given path, but with less acceleration-deceleration phases.

6.2. TADPF-SMPF Controller

This section deals with comparison of the TADPF and the SMPF controller and their combined version as given in Figure 13 to Figure 18. Since the SMPF control scheme does not provide longitudinal velocity commands, the lateral acceleration-based constraints along the path of the TADPF control scheme (Eq. 12 and Eq. 13) were used to generate the longitudinal velocity control inputs in all three cases (Figure 13). When analyzing the steering angle, curvature and lateral acceleration of the vehicle in Figures 14, 16 and 17, respectively, it can be seen that the peak values of the TADPF method are smaller, due to the fact that a reference

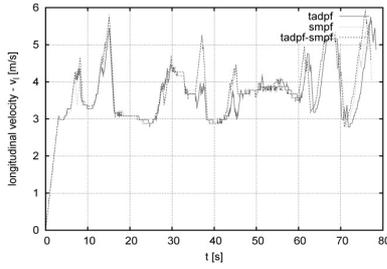


Figure 13. Velocity v_l comparison for all controllers.

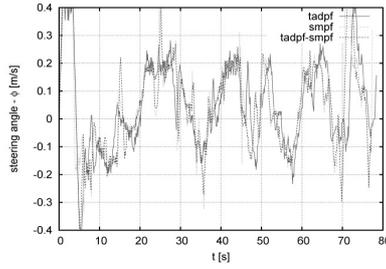


Figure 14. Steering angle ϕ comparison for all controllers.

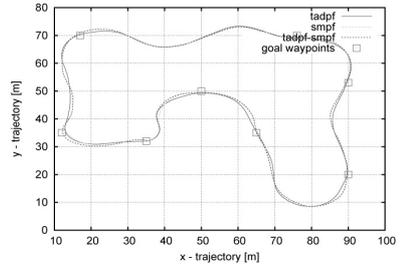


Figure 15. Trajectory x-y comparison for all controllers.

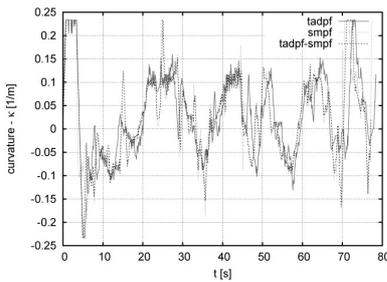


Figure 16. Curvature κ comparison for all controllers.

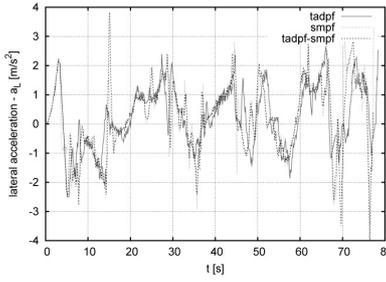


Figure 17. Lateral acceleration a_L comparison for all controllers.

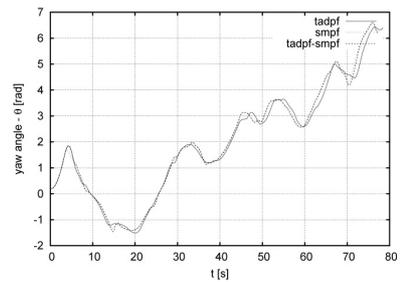


Figure 18. Orientation θ comparison for all controllers.

point look-ahead distance is used, combined with the global navigation function in the configuration space. On the other hand, the SMPF and TADPF-SMPF rely exclusively on the vehicle reference point, which is effectively the closest point on the path with respect to the vehicle rear-axes middle point. As a consequence, abrupt changes in path curvature and tangent direction may cause steering angle control peaks. However, the TADPF-SMPF method provides smaller path following errors with respect to the path generated by the global planner and can be used efficiently and with less control effort if the global path is smooth enough. In comparison, the TADPF method alone tends to smoothen the vehicle trajectory through the free space because it's based on both global navigation function and current trajectory as can be seen in Figures 15 and 18.

If the smoothness and non-holonomicity of the on-line generated global path is not guaranteed, the TADPF method alone is preferred, since it produces less steering signal effort, while still following the global path direction. In contrast, if a global path is smooth, possibly optimized for lesser curvature change along the whole path, the preferred method is TADPF-SMPF that provides smaller path following errors and is invariant on the choice of the reference point look-ahead distance, since it bound to the closest point on the path. Note that the SMPF controller alone is not suitable for real application, since it provides no guarantees on safety with comparison to TADPF or TADPF-SMPF method, which include collision checks at each cycle. Moreover, the dynamic constraints are not taken into account, which can cause larger delays or even instabilities in the cases of limited actuators, in this case particularly the steering rate $\dot{\phi}$.

The newly proposed path following schemes TADPF and TADPF-SMPF and the overall navigation framework in Section 2. have been tested extensively experimentally on the Smart autonomous test vehicle at constant longitudinal speeds of up to 20km/h , in a-priori unknown environments where also moving obstacles such as pedestrians where present, see [10] and [11], respectively.

7. Conclusion and Outlook

In this work, three different control strategies to path following for autonomous navigation of vehicles have been analyzed and compared. Firstly, the TADPF controller is based on a set of feasible arc trajectories according to kinodynamic and configuration space constraints, where the optimal command is chosen according to the global objective along the path. Longitudinal velocity profile was determined in this paper according to the lateral acceleration constraints for increased comfort of driving. A control theory-based SMPF controller was analyzed and enhanced with the obstacle collision, kinodynamic and comfort limits of the vehicle at each control cycle, resulting in the TADPF-SMPF control scheme. Thus, inherent safety and account for vehicle dynamic limits was achieved, regardless of the specific path following scheme used. The TADPF control scheme provided less steering control effort due to the usage of the global navigation function smoothing of free-space, whereas TADPF-SMPF control scheme provided less path following error with the expense of increased steering effort. In future work, both TADPF and TADPF-SMPF controllers will be tested also on a global path that will be further optimized after the initial smoothed geometric path of the global navigation function is obtained.

It is expected that decreasing the curvature changes along the global path will also decrease the steering control effort of the vehicle. Furthermore, a longitudinal velocity profile could be introduced with less acceleration-deceleration phases based on multi-step velocity prediction along the path compared to the one-step velocity prediction in the current scheme. Moreover, the presented navigation scheme is based on the "frozen-world" assumption, thus at the instance of replanning phase the environment obstacles are assumed static, even though possibly not known a-priori. Under the assumption of navigation in more general dynamic environments, the completeness of the global planning scheme cannot be guaranteed, whereas the collision avoidance scheme as presented here can still be applied, if the obstacle trajectories are known sufficiently far in the future and the multi-step collision checking is performed. Nevertheless, the currently proposed navigation

scheme was successfully applied in experimental environments with slowly moving obstacles, such as pedestrians, if the replanning rate was sufficiently high (in our implementation, rated at 10Hz).

8. Acknowledgment

This work has partly been supported by the EC under contract number FP6-IST-027140-BACS.

References

- [1] PHILIPPSSEN, R. AND SIEGWART, R., An Interpolated Dynamic Navigation Function, *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2005.
- [2] FERGUSON, D. AND STENTZ, A., Field D*: An Interpolation-based Path Planner and Replanner, *Proceedings of International Symposium on Robotics Research (ISRR)*, 2005.
- [3] PHILIPPSSEN, R., KOLSKI S., MACEK, K. AND SIEGWART, R., Path Planning, Replanning and Execution for Autonomous Driving in Urban and Offroad Environments, In *ICRA 2007 Workshop on Planning, Perception and Navigation of Intelligent Vehicles (PPNIV)*, 2007.
- [4] MACEK, K., PETROVIC, I. AND SIEGWART, R., A Control Method for Stable and Smooth Path Following of Mobile Robots, In *Proceedings of European Conference on Mobile Robots*, 2005.
- [5] SOLEA, R. AND NUNES, U., Trajectory Planning with Velocity Planner for Fully-automated Passenger Vehicles, *Proceedings of the Intelligent Transportation Systems Conference (ITSC)*, 2006.
- [6] BARTELS, R. H., BEATTY, J. C. AND BARSKY, B. A., An Introduction to Splines for Use in Computer Graphics and Geometric Modeling, Morgan Kaufmann Publishers, Inc., Los Altos, CA 94022, 1987.
- [7] LAMON, P., KOLSKI, S. AND SIEGWART, R., The SmartTer – a Vehicle for Fully Autonomous Navigation and Mapping in Outdoor Environments, In *Proceedings of CLAWAR 2006, Brussels, Belgium*, 2006.
- [8] FOX, D., BURGARD, W. AND THRUN, S., Controlling synchro-drive robots with the dynamic window approach to collision avoidance, In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.
- [9] ISO, Mechanical vibration and shock – Evaluation of human exposure to whole body vibrations – Part 1: General requirements, ISO 2631-1, 1997.
- [10] <https://svn.asl.ethz.ch/kristijan/public/videos/tadpf.mpg>
- [11] <https://svn.asl.ethz.ch/kristijan/public/videos/tadpf-smpf.mpg>

Received: November, 2007

Revised: July, 2008

Accepted: August, 2008

Contact addresses:

Kristijan Maček
Autonomous Systems Lab
Swiss Federal Institute of Technology
Zurich, Switzerland
e-mail: kristijan.macek@mavt.ethz.ch

Roland Siegwart
Autonomous Systems Lab
Swiss Federal Institute of Technology
Zurich, Switzerland
e-mail: rsiegwart@ethz.ch

Roland Philippsen
Robotics and Artificial Intelligence Lab
Stanford University, USA
e-mail: roland.philippsen@gmx.net

KRISTIJAN MAČEK obtained his B.S. and M.S. degree in electrical engineering from the Faculty of Electrical Engineering and Computing, Zagreb, in 1999 and 2004, respectively. He is currently a PhD student at the Autonomous Systems Lab, Swiss Federal Institute of Technology, Zurich. His main research topic is secure navigation of autonomous vehicles in dynamic environments, in particular hierarchical on-line planning schemes and dynamic scene interpretation.

ROLAND PHILIPPSSEN is a visiting scholar in the Stanford Robotics and AI Lab. He received his Ph.D. in mobile robotics from EPFL, Switzerland, in 2005. Dr. Philippsen is working on path planning and obstacle avoidance as well as on interweaving reasoning, planning, and control. He has successfully contributed to real-world robotics projects such as the tour-guide robots at the Swiss Expo.02, the EPFL student robot contest during 2001-2004, robotic actors for a theater play in 2005, and the European Cognitive Robot Companion in 2006. Currently, he is dividing his time between whole-body task-oriented motion planning for human-centered robotics at Stanford and providing strategic and technical support at Willow Garage in Menlo Park, California. He believes autonomous robots can improve the lives of all humanity, provided we openly share knowledge and implementations.

PROFESSOR DOCTOR ROLAND SIEGWART (1959) has worked as full professor of autonomous systems at ETH Zurich since July 2006. He holds a Diploma in mechanical engineering (1983) and PhD in mechatronics (1989) from ETH Zurich. In 1989/90 he spent one year as postdoctoral fellow at Stanford University. After that, he worked part time as R&D director at MECOS Traxler AG and as lecturer and deputy head at the Institute of Robotics, ETH Zurich. In 1996 he was appointed associate and later full professor of autonomous micro-systems and robots at the Ecole Polytechnique Federale de Lausanne (EPFL). During his period at EPFL he was co-initiator and founding Chairman of Space Center EPFL and Vice Dean of the School of Engineering. In 2005 he held a visiting position at NASA Ames and Stanford University. Roland Siegwart is a member of the Swiss Academy of Engineering Sciences and a board member of the European Network of Robotics (EURON). He served as Vice President for Technical Activities (2004/05) and is currently a Distinguished Lecturer (2006/07) and an AdCom Member (2007-2009) of the IEEE Robotics and Automation Society. He is a member of the "Bevolligungsausschuss Exzellenzinitiative" of the "Deutsche Forschungsgemeinschaft (DFG)", coordinator of two European projects and a co-founder of several spin-off companies.
