

Document downloaded from:

<http://hdl.handle.net/10251/62883>

This paper must be cited as:

Girbés, V.; Armesto Ángel, L.; Tornero Montserrat, J. (2014). Path following hybrid control for vehicle stability applied to industrial forklifts. *Robotics and Autonomous Systems*. 62(6):910-922. doi:10.1016/j.robot.2014.01.004.



The final publication is available at

<http://dx.doi.org/10.1016/j.robot.2014.01.004>

Copyright Elsevier

Additional Information

Path Following Hybrid Control for Vehicle Stability Applied to Industrial Forklifts

Vicent Girbés**, Leopoldo Armesto, and Josep Tornero

Institut de Disseny i Fabricació (IDF), Universitat Politècnica de València,
Camí de Vera s/n, 46022 (València, Spain),
{vgirbes, larmesto, jtornero}@idf.upv.es

Abstract. The paper focuses on a closed-loop hybrid controller (kinematic and dynamic) for path following approaches with industrial forklifts carrying heavy loads at high speeds, where aspects such as vehicle stability, safety, slippage and comfort are considered. The paper first describes a method for generating Double Continuous Curvature (DCC) paths for non-holonomic wheeled mobile robots, which is the basis of the proposed kinematic controller. The kinematic controller generates a speed profile, based on “slow-in” and “fast-out” policy, and a curvature profile recomputing DCC paths in closed-loop. The dynamic controller determines maximum values for decelerations and curvatures, as well as bounded sharpness so that instantaneous vehicle stability conditions can be guaranteed against lateral and frontal tip-overs. One of the advantages of the proposed method, with respect to full dynamic controllers, is that it does not require dynamic parameters to be estimated for modelling, which in general can be a difficult task. The proposed kinematic-dynamic controller is afterwards compared with a classic kinematic controller like Pure-Pursuit. For that purpose, in our hybrid control structure we have just replaced the proposed kinematic controller with Pure-Pursuit. Several metrics, such as settling time, overshoot, safety and comfort have been analysed.

Keywords: Path Following, Vehicle Stability, Continuous Curvature Paths, Kinematic and Dynamic Control, Hybrid Control.

1 Introduction

Safe and comfortable driving implies taking into account aspects such as normal and tangential accelerations and their corresponding derivatives, the jerks. It is well known that safety and comfort increase when generating continuous curvature paths. These kind of solutions can be applied on the industry in situations such as transportation of goods and materials in hostile environments in the safest possible manner. This is the case where the kind of material that is being transported must be handled with caution, because it is fragile, hazardous or explosive or because the area where the vehicle moves contains potentially danger aspects such as explosive area or gas pipelines.

** Corresponding Author

Path following problems have been studied intensively in the past and can be applied on different approaches, covering a wide spectrum of applications such as vision-based line following, path generation for overtakes, lateral tracking, parking, etc. One classic approach is to generate a path that converges to a ground painted line, based on direct data from a vision system (vision-based control). Another classic application is to estimate the road profile and to provide a curvature control law that keeps the vehicle within the lane bounds.

The main contribution of this paper is to provide a new closed-loop hybrid controller (kinematic and dynamic), where aspects such as vehicle stability, safety, slippage and comfort are considered. In particular, the kinematic controller generates generic continuous-curvature paths, coined as Double Continuous-Curvature paths (DCC), used in path following problems for non-holonomic vehicles. The controller applies a curvature profile to steer the vehicle to converge to a path. Through the paper, it will be shown that the proposed continuous-curvature method uses combinations of clothoids, line segments and arcs to cope with generic curvature profiles. The method can generate paths between two arbitrary configurations (current pose and target) composed by positions, orientations and curvatures. The type of provided solutions gets benefit from higher comfort and safety and constitutes a set of “natural” paths with the shortest possible length. In addition, a speed profile is also proposed to cope with human-like driving based on “slow-in” and “fast-out” policy. In addition, the dynamic controller guarantees dynamic stability conditions in order to avoid lateral and frontal tip-overs using only odometry and inertial data. For that purpose, lower and upper bounds on sharpness and maximum curvature constrain generated trajectories. Compared to standard dynamic controllers, one of the advantages of the proposed method is that it does not require to estimate complex parameters such as inertial, torques and frictions. In this work we use a clothoid-based approach because they have an explicit relation with jerks (at constant velocity), so designing clothoid-based paths implies planning paths with limited jerks, which also has direct impact on comfort and safety. Other curves such as Bézier or Splines can also provide smooth curvature profiles, but their derivatives are not limited, unless explicitly stated.

In Girbés et al. (2011a,b), the authors already introduced the DCC path generation method which constitutes the mathematical background for this paper. The new contribution of the paper with respect to our previous work (Girbés et al., 2011a,b) is to provide necessary conditions for vehicle stability with an industrial forklift carrying a heavy load at high speeds and to provide an exhaustive analysis to evaluate the performance of the new method with respect to classic ones. Moreover, we provide several examples and videos showing the advantages of the proposed method.

1.1 Related work

During the last years, path following problems, whose goal is to generate a path and follow it with a kinematic control law, have been studied intensively because they can be applied in vehicle applications such as: parking (Laumond et al., 1994; Jiang et al., 1999), overtaking and lane changing (Papadimitriou and Tomizuka, 2003; Montés et al., 2007; Wilde, 2009), vision-based line following (Manz et al., 2010; Girbés et al., 2010), etc. In that sense, the well-known Pure-Pursuit method determines appropriate vehicle’s

curvature and velocity that guarantee convergence to a specific path or trajectory based on current robot pose (Wallace et al., 1985; Ollero and Heredia, 1995; Ollero, 2001). However, applications such as path following or kinematic control differ from motion planning and obstacle avoidance methods, since they do not generally take obstacles into account, neither solve the global path planning problem. For a complete reference on motion planning and obstacle avoidance methods see Minguez et al. (2008).

Most kinematic controllers are based on current robot and target poses, but they do not take into account curvature continuity, which might affect seriously to comfort when transporting people, and safety when transporting dangerous goods. These aspects can affect load stability in transportation systems and they can even affect wheels slippage and therefore odometry errors. In order to generate continuous-curvature paths in mobile robotics, some researchers use clothoids to generate paths in navigation problems (Fraichard and Mermond, 1998; Scheuer and Xie, 1999; Yang and Sukkarieh, 2008; Labakhua et al., 2008) because of their “nice” geometric properties including a close relation between physical phenomena (normal acceleration and jerks) with the clothoid scaling parameter. In Scheuer and Fraichard (1996), Elementary paths were first introduced, a combination of two symmetrical clothoids with the same homothety factor. These ideas were extended in Scheuer and Fraichard (1997a), by introducing the concept of Bi-Elementary paths, combinations of two Elementary paths. In Bi-Elementary paths the initial and final configurations are not necessary symmetric, but the loci of the intermediate configuration is restricted to a circle with specific orientations to ensure that each Elementary path contains symmetrical clothoids. Obviously, the solution space is significantly limited in those cases and Elementary and Bi-Elementary paths might not be appropriate to solve specific problems, specially the obstacle avoidance problem or the line following problem with bounded sharpness and curvature. Dubins’ curves (Dubins, 1957) were the inspiration in Scheuer and Fraichard (1997b) to create the SCC-paths (Simple Continuous-Curvature paths) and thus simplify the problem of finding optimal paths for vehicles that can go only forward, while keeping curvature continuity. They replaced the circular arcs of Dubins’ paths to the called CC-turns, in order to perform paths defined as a combination of clothoids, circular arcs and line segments. The authors of Fraichard and Scheuer (2004) used RS-paths (Reeds and Shepp, 1990) to extend SCC-paths by creating continuous-curvature paths that ensure continuity for vehicles moving both forward and backward.

In order to increase driving safety and comfort, many studies have been done to determine appropriate values for super-elevation and side friction factor for horizontal road alignment (Krammes and Garnham, 1998; Marchionna and Perco, 2007). These studies also establish appropriate values for clothoid sharpness in transition curves. Moreover, in mobile robotics some efforts have been done to improve stability and to avoid robot tip-over by providing the ability of load reconfiguration for robots with manipulators carrying loads. For instance, there are some analysis of the stability of vehicles carrying heavy loads, establishing dynamic models and guidelines to follow in order to avoid accidents (Korayem et al., 2004, 2010; Xinye et al., 2010). Other works use mobile manipulators so that the center of gravity can be repositioned to avoid roll-over when travelling on slopes or on uneven terrains (Beck et al., 2009; Valls et al., 2010; Liu and Liu, 2010).

2 Smooth Curvature Path Generation

2.1 Problem Statement

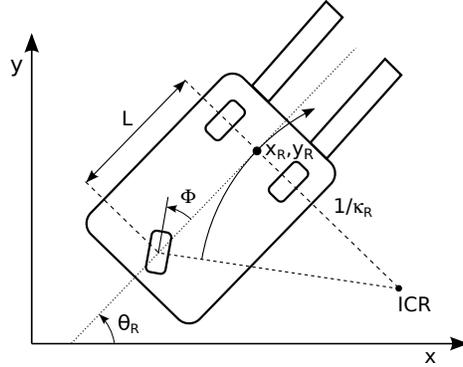


Fig. 1. Vehicle \mathcal{R} in tricycle configuration with back steering wheel.

Let \mathcal{R} be a non-holonomic wheeled robot moving on a 2D plane with extended state space $\mathbf{q}_R(t) = (x_R(t), y_R(t), \theta_R(t), \kappa_R(t))^T \in \mathfrak{R}^2 \times \mathcal{S} \times \mathfrak{R}$ containing the robot Cartesian positions $x_R(t)$ and $y_R(t)$, the robot orientation $\theta_R(t)$ and the curvature $\kappa_R(t)$, which is the inverse of the radius of the robot instantaneous center of rotation (see Figure 1).

The kinematic model for \mathcal{R} is:

$$\dot{\mathbf{q}}_R(t) = \begin{bmatrix} \dot{x}_R(t) \\ \dot{y}_R(t) \\ \dot{\theta}_R(t) \\ \dot{\kappa}_R(t) \end{bmatrix} = \begin{bmatrix} v_R(t) \cos \theta_R(t) \\ v_R(t) \sin \theta_R(t) \\ v_R(t) \kappa_R(t) \\ v_R(t) \sigma(t) \end{bmatrix} \quad (1)$$

being $v_R(t)$ the robot linear velocity and $\sigma(t)$ the sharpness of the path.

We assume that \mathcal{R} is moving with bounded curvature $\kappa_R(t) \in [-\kappa_{max}, \kappa_{max}]$ and sharpness $\sigma(t) \in [-\sigma_{max}, \sigma_{max}]$. Curvature bounds are due to mechanical constraints, where for a forklift with tricycle configuration and back orientable wheels the following expression stands $\kappa_{max} = \tan \Phi_{max} / L$, being L the distance between front and rear wheels and Φ_{max} the maximum angle of steering wheels. On the other hand, sharpness upper and lower bounds are introduced based on the ideas suggested by Marchionna and Perco (2007) to increase safety and satisfy comfort limits, improve road appearance, etc.

The goal is to generate a continuous-curvature path \mathcal{Q} connecting the current robot pose $\mathbf{q}_R(t)$ to target configuration $\mathbf{q}_T(t) = [x_T(t), y_T(t), \theta_T(t), \kappa_T(t)]^T$, while taking curvature and sharpness upper-bounds into account. Notice that κ_R and κ_T can be any arbitrary value in the range $\pm \kappa_{max}$, unlike SCC-paths in Scheuer and Fraichard (1997b) or CC-paths in Fraichard and Scheuer (2004) that connect two configurations in the plane with null curvature only.

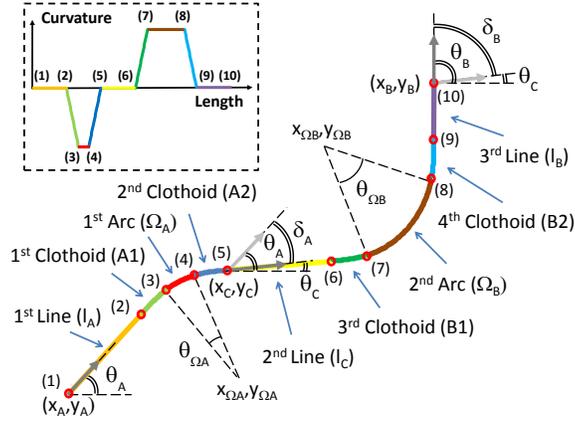


Fig. 2. Standard Double Continuous-Curvature Path definition and its curvature profile.

2.2 Double Continuous-Curvature Paths

A Double Continuous-Curvature (DCC) path is composed by clothoids, circular arcs and line segments, providing a set of general solutions, that guarantee continuity in curvature and a bounded curvature derivative. DCC paths were introduced in Girbés et al. (2011a), and they are based on the following sequence: line-clothoid-arc-clothoid-line-clothoid-arc-clothoid-line. DCC are indeed based on two Continuous-Curvature turns (like CC-turns in Scheuer and Fraichard (1997b) and Fraichard and Scheuer (2004)) connected with straight line segments. The first line-segment and CC-turn path (noted with subscript A) start at configuration $\mathbf{q}_A = [x_A, y_A, \theta_A, \kappa_A]^T$, with $\kappa_A = 0$, while the second CC-turn and line segment (noted with subscript B) finish at posture $\mathbf{q}_B = [x_B, y_B, \theta_B, \kappa_B]^T$, with $\kappa_B = 0$. The configuration joining both CC-turns is $\mathbf{q}_C = [x_C, y_C, \theta_C, \kappa_C]^T$, with $\kappa_C = 0$ at the beginning of the common straight line segment.

Figure 2 shows an example of a DCC path together with its curvature profile. There are four clothoids named as **A1**, **A2**, **B1** and **B2** with lengths l_{A1} , l_{A2} , l_{B1} and l_{B2} , respectively. There are also two circular segments Ω_A and Ω_B with lengths l_{Ω_A} and l_{Ω_B} , and three line segments with lengths l_A , l_C and l_B . Particular solutions can be derived using only four clothoids, obtaining similar solutions to Bi-Elementary paths in Scheuer and Fraichard (1997a). In this sense, it is interesting to remark that, the proposed formulation is a generalization of other continuous curvature paths given in the literature based on clothoids or circles. In addition, our proposed method is not forcing the same sharpness for the different clothoids namely $\sigma_{A1}, \sigma_{A2}, \sigma_{B1}, \sigma_{B2} \in [\sigma_{min}, \sigma_{max}]$, opening a wider spectrum of possibilities, being $s_A = \text{sign}(\sigma_{A1}) = -\text{sign}(\sigma_{A2})$ and $s_B = \text{sign}(\sigma_{B1}) = -\text{sign}(\sigma_{B2})$ curvature's signs of each CC-turn, with opposite signs for clothoids belonging to the CC-turn. The attached Video 1 in Table 1 in Appendix C demonstrates the type of solutions provided with the proposed method.

Let $\mathcal{Q} \in \mathbb{R}^2 \times \mathcal{S} \times \mathbb{R}$ be a standard DCC path defined by a set of parameters as follows:

$$\mathcal{Q}(\lambda, \theta_C, l_A, l_{A1}, l_{\Omega_A}, l_{A2}, l_C, l_{B1}, l_{\Omega_B}, l_{B2}, l_B, \sigma_{A1}, \sigma_{A2}, \sigma_{B1}, \sigma_{B2}, s_A, s_B) \quad (2)$$

where $\lambda \in [0, \lambda_B]$ is the free variable describing all possible configurations from $\mathbf{q}_A = \mathcal{Q}(0, \dots)$ to $\mathbf{q}_B = \mathcal{Q}(\lambda_B, \dots) + \mathbf{q}_A$ with $\lambda_B = l_A + l_{A1} + l_{\Omega_A} + l_{A2} + l_C + l_{B1} + l_{\Omega_B} + l_{B2} + l_B$.

In Appendix B one can find specific details on how to compute DCC paths as a composition of all its parameters. Since $\sigma_{A1}, \sigma_{A2}, \sigma_{B1}, \sigma_{B2}, s_A$ and s_B are design parameters, it is easy to see that, in order to satisfy curvature continuity and first derivative constraints, most of the parameters of a DCC path are dependent and therefore \mathcal{Q} can be re-defined as:

$$\mathcal{Q}(\lambda, \theta_C, l_A, l_B, l_C) \quad (3)$$

where,

$$l_{A1} = |\kappa_A \sigma_{A1}^{-1}|, l_{A2} = |\kappa_A \sigma_{A2}^{-1}|, l_{\Omega_A} = |\theta_{\Omega_A} \kappa_A^{-1}| \quad (4)$$

$$l_{B1} = |\kappa_B \sigma_{B1}^{-1}|, l_{B2} = |\kappa_B \sigma_{B2}^{-1}|, l_{\Omega_B} = |\theta_{\Omega_B} \kappa_B^{-1}| \quad (5)$$

with $\theta_{\Omega_A} = \delta_A - \theta_{A1} - \theta_{A2}$, $\theta_{A1} = \frac{\kappa_A^2}{2\sigma_{A1}}$, $\theta_{A2} = \frac{\kappa_A^2}{2\sigma_{A2}}$ and $\delta_A = s_A(\theta_C - \theta_A)$. Similarly, $\theta_{\Omega_B} = \delta_B - \theta_{B1} - \theta_{B2}$, $\theta_{B1} = \frac{\kappa_B^2}{2\sigma_{B1}}$ and $\theta_{B2} = \frac{\kappa_B^2}{2\sigma_{B2}}$ and $\delta_B = s_B(\theta_B - \theta_C)$. Curvatures of arc segments will be selected upon the following criteria:

$$\kappa_A = \min(\sqrt{\sigma_{A1}\delta_A}, \sqrt{\sigma_{A2}\delta_A}, \kappa_{max}) \quad (6)$$

$$\kappa_B = \min(\sqrt{\sigma_{B1}\delta_B}, \sqrt{\sigma_{B2}\delta_B}, \kappa_{max}) \quad (7)$$

It can be shown that the selected values for $\theta_{A1}, \theta_{\Omega_A}, \theta_{A2}, \theta_{B1}, \theta_{\Omega_B}$ and θ_{B2} guarantee appropriate changes on the orientation and curvature. Indeed, our approach selects the minimum admissible arc angle so if $\delta_A > \delta_{A,min} \equiv \min(|\frac{\kappa_{max}^2}{\sigma_{A1}}|, |\frac{\kappa_{max}^2}{\sigma_{A2}}|)$ or $\delta_B > \delta_{B,min} \equiv \min(|\frac{\kappa_{max}^2}{\sigma_{B1}}|, |\frac{\kappa_{max}^2}{\sigma_{B2}}|)$ then the corresponding arc segments will cover the remainder angle. Otherwise, the DCC path will not include arc segments and the clothoid-pairs segments will compensate such angles with appropriate curvature values at a maximum sharpness. In Section 2.4 we discuss that this is one of the most remarkable aspects of the proposed method compared to SCC-paths. However, this implies to solve an optimization problem, where the aim is to find \mathcal{Q}^* such as:

$$\mathcal{Q}^* = \arg \min_{\mathcal{Q}} |l_A| + |l_{A1}| + |l_{\Omega_A}| + |l_{A2}| + |l_C| + |l_{B1}| + |l_{\Omega_B}| + |l_{B2}| + |l_B| \quad (8)$$

with,

$$\mathcal{Q}(\lambda, \theta_C, l_A, l_B, l_C) = \left\{ \begin{array}{l} \theta_C \in \mathcal{S}, \\ l_A, l_B, l_C \in \mathbb{R} \end{array} \middle| \exists \lambda_B, \mathbf{q}_B = \mathcal{Q}(\lambda_B, \theta_C, l_A, l_B, l_C) + \mathbf{q}_A \right\} \quad (9)$$

which can be solved using Simplex (Nelder-Mead) method as optimization method and the initialization of the starting value for θ_C is described in Appendix B.

For a given θ_C , lengths of the line segments appear as a linear combination on the Cartesian elements of (9):

$$\begin{bmatrix} x_B \\ y_B \end{bmatrix} = \begin{bmatrix} \mathcal{P}'_x \\ \mathcal{P}'_y \end{bmatrix} + \begin{bmatrix} \cos \theta_A & \cos \theta_B & \cos \theta_C \\ \sin \theta_A & \sin \theta_B & \sin \theta_C \end{bmatrix} \cdot \begin{bmatrix} l_A \\ l_B \\ l_C \end{bmatrix} + \begin{bmatrix} x_A \\ y_A \end{bmatrix}, l_A \geq 0, l_C \geq 0 \quad (10)$$

where \mathcal{P}'_x and \mathcal{P}'_y represent the Cartesian coordinates of the remainder of terms that do not depend on line segment lengths and are known for a given θ_C . In order to solve (10), one can find many feasible solutions. Our heuristic criterion is to consider $l_A = 0$, since it is not needed in path following problems but formally introduced for path generation problems. Therefore, computation of l_C and l_B is straight forward once θ_C is given:

$$\begin{bmatrix} l_B \\ l_C \end{bmatrix} = \begin{bmatrix} \cos \theta_B & \cos \theta_C \\ \sin \theta_B & \sin \theta_C \end{bmatrix}^{-1} \cdot \begin{bmatrix} x_B - x_A - \mathcal{P}'_x \\ y_B - y_A - \mathcal{P}'_y \end{bmatrix} \quad (11)$$

2.3 DCCs with non-zero curvature profiles

A **standard** DCC path is defined for $\kappa_A = 0$ and $\kappa_B = 0$, however in a navigation problem may be situations in which $\kappa_R \neq 0$ and $\kappa_T \neq 0$, as shown in Figure 3. In these cases $\mathbf{q}_R \neq \mathbf{q}_A$ and $\mathbf{q}_T \neq \mathbf{q}_B$ to satisfy curvature continuity in coherence with the actual robot curvature. From now on, we will focus on the case where $\kappa_T = 0$, although with a similar procedure, we can also target configurations with non null curvatures. Without loss of generality, we assume that the robot is located within clothoid **A1** segment and therefore the start configuration of a DCC path is *a priori* unknown (see Figure 3). In that sense, the robot configuration can be expressed as:

$$\mathbf{q}_R = \Delta \mathbf{q}_{A1,R} \oplus \mathbf{q}_A \quad (12)$$

where,

$$\Delta \mathbf{q}_{A1,R} = \left[\mathbf{R}(\theta_A) \begin{bmatrix} x_{A1,R} \\ s_A \cdot y_{A1,R} \end{bmatrix} \ s_A \cdot \theta_{A1,R} \ \kappa_{A1,R} \right]^T \quad (13)$$

where all possible cases of $x_{A1,R}$, $y_{A1,R}$ and $\theta_{A1,R}$ can be computed from Fresnel integrals as follows (see Appendix A).

Henceforth, we will use notation $\Delta \mathbf{q}_{A1,R}$ to refer to the clothoid segment at the robot configuration, while $\Delta \mathbf{q}_{A1}$ to refer to the complete clothoid segment. Figures 3(a), 3(c) and 3(e) depict a case in which robot and target's signs of curvature ($s_R = \text{sign}(\kappa_R)$ and $s_T = \text{sign}(\kappa_T)$) are the same as the signs of the generated standard DCC path (s_A and s_B), while Figures 3(b), 3(d) and 3(f) show a situation where they are different. These examples are for a given pair of configurations \mathbf{q}_R and \mathbf{q}_T , which give $s_A > 0$ and $s_B < 0$, although other combinations are possible and can be handled using the same policy.

By marginalising out \mathbf{q}_A from Eq. (12) we can obtain the appropriate starting point of the DCC path $\mathbf{q}_A = \mathbf{q}_R \ominus \Delta \mathbf{q}_{A1,R}$.

2.4 Comparing DCCs with SCCs and Dubins' paths

The proposed DCC paths might be identical to SCC paths in Scheuer and Fraichard (1997b) if the solution implies arc segments with large deflection angles. However,

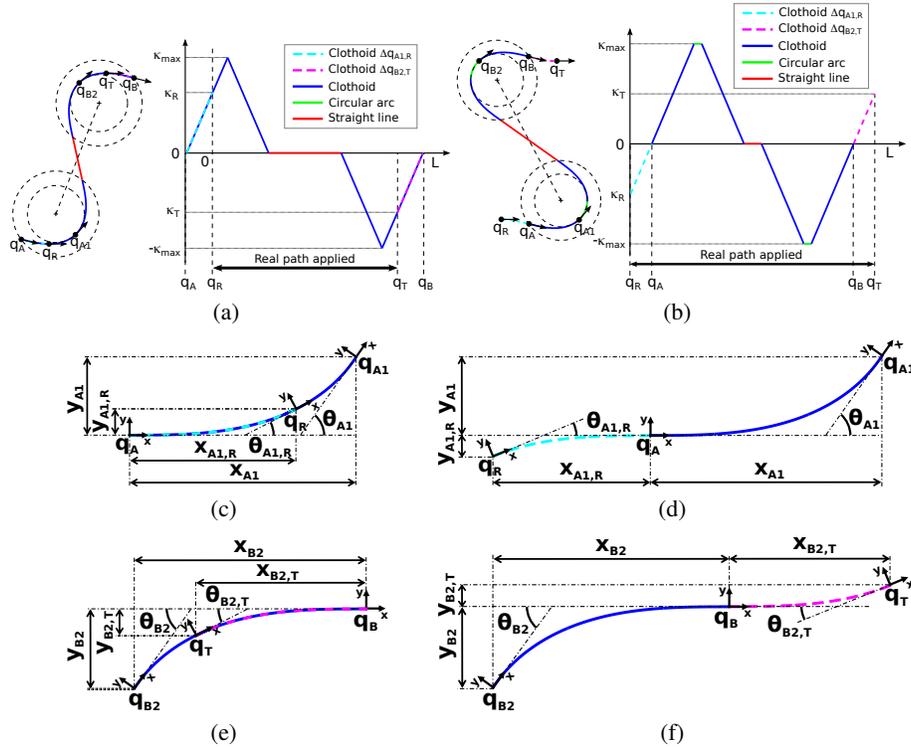


Fig. 3. DCC paths and curvature profiles for two given robot and target configurations, (a) when $s_R = s_A$ and (b) when $s_R \neq s_A$. Robot configuration \mathbf{q}_R along clothoid **A1**, (c) for DCC in (a), and (d) for DCC in (b). Target configuration \mathbf{q}_T along clothoid **B2**, (e) for DCC in (a), and (f) for DCC in (b).

DCCs are better suited for the path following problem as depicted in the blue line of Figure 4(a), where DCC paths can provide solutions where the $l_B < 0$. We name this case as *relaxed* solution, because the final negative length segment is never applied (the path is recomputed on next update as described in Section 3). It can be shown that this kind of converge-to-line solutions provide less oscillatory behaviour in path following problems than the solutions proposed by SCCs (see green dash-dotted line of Figure 4(a)) or even Dubins' paths, which applying maximum curvature are still not able to reach the goal without making a big turn (red dashed line in Figure 4(a)).

Another advantage of DCC paths with respect to SCCs is that the proposed solutions can provide even shorter paths for cases with small deflection angles. That is, cases where the optimization procedure of DCC provides a solution with the maximum admissible sharpness, while the solution provided by SCCs adapts the sharpness so that deflection angle is satisfied. This case is depicted in Figure 4(b), where it can be appreciated that the DCC method finds a solution with the maximum sharpness and because of that the overall path length of DCC is shorter.

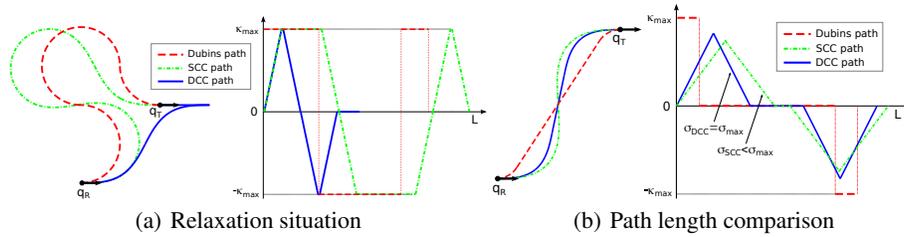


Fig. 4. Comparison of Dubins (red dashed), SCC (green dash-dotted) and DCC (blue) paths.

One important drawback of SCCs in the context of path following problems is that they can not guarantee smooth curvature changes if $\kappa_R \neq 0$. The problem arises from the fact that the sharpness that will be used in their clothoid segments can not be known *a priori*, so they can not either compute the starting point of the clothoid unless $\kappa_R = 0$. For instance, if we apply the method described in Section 2.3, the SCC method would fail to pass even through the robot configuration q_R or even to have the appropriate curvature for $L = 0$, as shown in Figure 5.

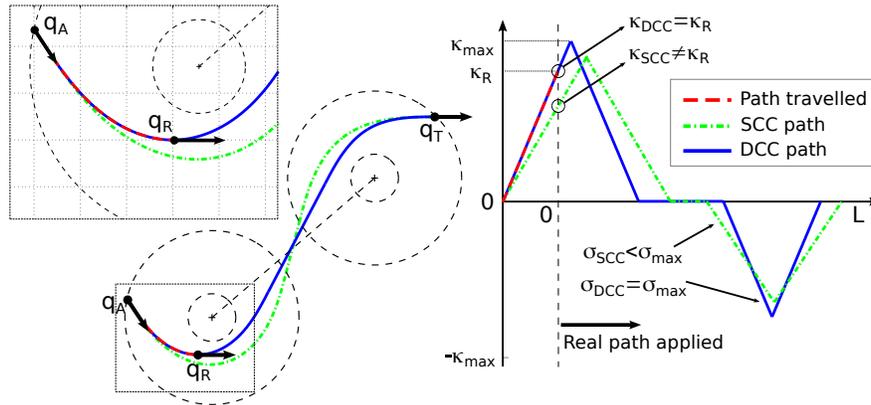


Fig. 5. Example of path following problem in which SCCs cannot guarantee curvature continuity, while DCC paths always do.

Regarding the computational cost, SCC and DCC paths are equivalent when optimization procedure is not necessary, being the average time in such cases around $20\mu s$ in our implementation of the methods. However, when an optimization is performed, DCC's processing time depends on the number of iterations required, although it has good convergence rates, being at most five times slower than SCC, i.e. at most $100\mu s$. These results have been obtained in our implementation with ROS Fuerte in Ubuntu 12.04 (precise), using a computer with processor Intel Core i7-2670QM 2.20GHz and 8GB RAM.

3 Smooth Curvature Hybrid Controller: Dynamic Stabilization of Industrial Forklifts

It is well-known that closed-loop structures increase control robustness to achieve a specific target even with noises, unmodelled dynamics and disturbances. In this sense, this section describes the application of DCC paths to the kinematic control problem with dynamic target selection, which is the case of pure-pursuit framework or car-like planners. This situation is highlighted in Figure 6, where a vehicle is following a meta-path (grey dashed path proposed, let's say, by a global planner) build up from a set of way-points (blue points). It first follows a preliminary path (red path) during a certain amount of time so it tries to converge to the meta-path based on a Look-Ahead (LA) distance criterion. On next position update, the robot might be shifted from the original path due to unmodelled dynamics or disturbances (in Figure 6 the update period is extremely large on purpose to highlight errors), so we need to recompute a new path based on the actual robot position (green path). Again, on next update, we need to recompute the path in order to provide a third path based on new position update (blue path). In this section we make explicit time t dependency for dynamic variables of such robot configuration (including its speed and curvature), target selection, etc.

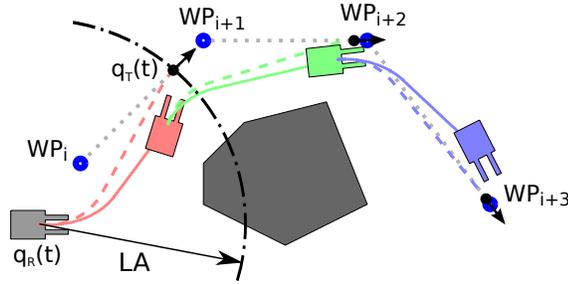


Fig. 6. Target re-computation based on Look-Ahead distance (Pure-Pursuit framework) with odometry errors, where dashed line is the planned path and continuous line is the real path.

At this point, we want to remark the differences between target configurations, way-points and goals within the context of this paper. Target configurations $\mathbf{q}_T(t)$ are obtained from the set of way-points \mathcal{W} as explained herein after. On the contrary, way-points constitute the global path to be followed based, for instance, on a global planner method such as “wavefront planner” to lead the robot to the goal configuration \mathbf{q}_G . Therefore, $\mathbf{q}_T(t)$ is obtained from a given sequence of way-points to be reached based on a LA distance. This can be formulated as the configuration within two consecutive waypoints $\mathbf{WP}_i = [x_{\mathbf{WP}_i}, y_{\mathbf{WP}_i}, \theta_{\mathbf{WP}_i}]^T$ and $\mathbf{WP}_{i+1} = [x_{\mathbf{WP}_{i+1}}, y_{\mathbf{WP}_{i+1}}, \theta_{\mathbf{WP}_{i+1}}]^T$ with $\theta_{\mathbf{WP}_i} = \arctan\left(\frac{y_{\mathbf{WP}_{i+1}} - y_{\mathbf{WP}_i}}{x_{\mathbf{WP}_{i+1}} - x_{\mathbf{WP}_i}}\right)$:

$$\mathbf{q}_T(t) = [\lambda^*(\mathbf{WP}_{i+1} - \mathbf{WP}_i) + \mathbf{WP}_i, \theta_{\mathbf{WP}_i}, 0]^T \quad (14)$$

$$\lambda^* = \arg \min_{\lambda \geq \lambda_{\perp}} \left\| \lambda(\mathbf{WP}_{i+1} - \mathbf{WP}_i) + \mathbf{WP}_i - [x_R, y_R]^T \right\|_2 - LA \quad (15)$$

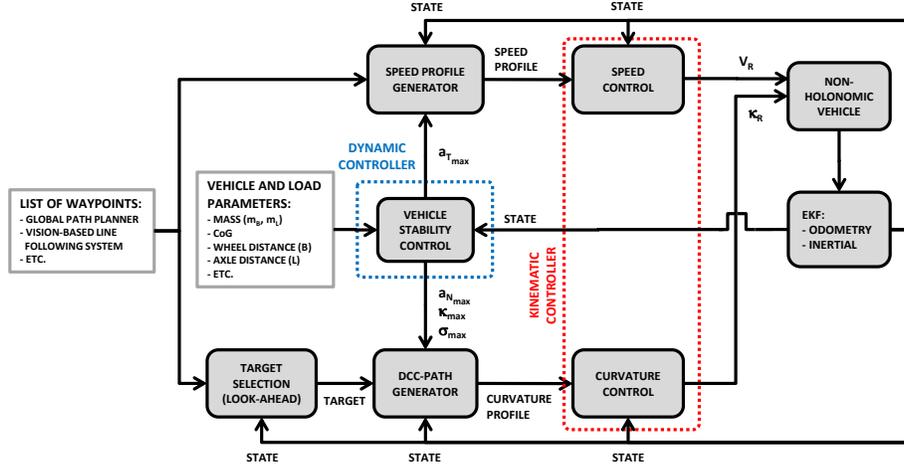


Fig. 7. Closed-loop Hybrid Controller, including a kinematic and a dynamic controller.

$$\lambda_{\perp} = \arg \min_{0 \leq \lambda \leq 1} \|\lambda(\text{WP}_{i+1} - \text{WP}_i) + \text{WP}_i - [x_R, y_R]^T\|_2 \quad (16)$$

Equation (14) defines the target configuration between waypoint segments, with λ^* being the value in which the circle around the robot position with radius LA intersects with the waypoint segment (15). If they do not intersect then the value is selected upon the projection of the robot pose along segment (16).

Our approach to perform a kinematic control of the vehicle is based on a cascade-loop structure, as shown in Figure 7. The controller includes speed and curvature controllers that require speed and curvature profiles obtained from a speed reference generator and DCC path generator, respectively. Speed reference generator simply provides a constant acceleration profile as will be explained later, while the latter has been already described in Section 2. In addition to this, overall closed-loop structure includes a vehicle stability controller derived in Section 3.2 which requires an Extended Kalman Filter (EKF) to estimate the vehicle configuration from inertial and odometry sensors. Target selection is based on LA criteria, as mentioned before. In this paper, we use inertial data not only to feed EKF estimation but for the dynamic stabilization controller (body accelerations) as described in Section 3.2. Here, EKF estimation does not include stereoceptive sensors like ranger finders or cameras for simplicity, and no map is built because these aspects are out of the scope of the paper. Indeed, the implementation of EKF is based on package “robot_pose_ekf” in the ROS Navigation stack . The update frequencies of DCC path and speed profile generators, vehicle stability control and target selection are set to 20 Hz, while the update frequencies of the speed and curvature controllers and EKF are 100 Hz.

3.1 Kinematic Controller

The curvature controller defines a piecewise-function based on the travelled distance along the computed DCC path. We assume that the vehicle is always located on the first clothoid **A1** and it has already travelled l_{A0} from DCC's origin \mathbf{q}_A as a consequence of the method described in Section 2.3. Thus, the distance along the DCC path is computed as $l_R(t) = \int_0^t v_R(t)dt + l_{A0}$. The curvature controller uses the estimated state from EKF to compute the relative travelled distance since DCC path last update. Based on DCC path definition, we define the following piecewise-function:

$$\kappa_R(t) = \begin{cases} 0 & \text{if } 0 \leq l_R(t) \leq l_1 \\ s_A(l_R(t) - l_1)\sigma_A & \text{if } l_1 < l_R(t) \leq l_2 \\ l_{A1}\sigma_A & \text{if } l_2 < l_R(t) \leq l_3 \\ l_{A1}\sigma_A - s_A(l_R(t) - l_3)\sigma_A & \text{if } l_3 < l_R(t) \leq l_4 \\ 0 & \text{if } l_4 < l_R(t) \leq l_5 \\ s_B(l_R(t) - l_5)\sigma_B & \text{if } l_5 < l_R(t) \leq l_6 \\ l_{B1}\sigma_B & \text{if } l_6 < l_R(t) \leq l_7 \\ l_{B1}\sigma_B - s_B(l_R(t) - l_6)\sigma_B & \text{if } l_7 < l_R(t) \leq l_8 \\ 0 & \text{if } l_8 < l_R(t) \end{cases} \quad (17)$$

with $l_1 = l_A$, $l_2 = l_1 + l_{A1}$, $l_3 = l_2 + l_{\Omega_A}$, $l_4 = l_3 + l_{A2}$, $l_5 = l_4 + l_C$, $l_6 = l_5 + l_{B1}$, $l_7 = l_6 + l_{\Omega_B}$, $l_8 = l_7 + l_{B2}$ and $l_9 = l_8 + l_B$. For a demonstration of the kinematic controller with an industrial forklift, see Video 2 of Table 1 in Appendix C.

Speed profile generator is also defined as a piece-wise function considering times for speeding up and down based on path speed and acceleration profiles (including nominal speeds and speeds at turns). When approaching a curve, human use to instinctively apply a “slow-in” and “fast-out” policy, which implies to reduce velocity, i.e. v_{turn} , just before entering into a curve and then progressively increases the velocity up to a nominal value, i.e. v_{path} . The following function defines the proposed speed profile, which includes a constant acceleration a_{path} for speeding up and a constant deceleration a_{brake} for slowing down:

$$v_R(t) = \begin{cases} v_{path} & \text{if } t \leq t_1 \\ v_{turn} \frac{t-t_1}{t_2-t_1} - v_{path} \frac{t-t_2}{t_2-t_1} & \text{if } t_1 \leq t \leq t_2 \\ v_{turn} & \text{if } t_2 \leq t \leq t_3 \\ v_{path} \frac{t-t_3}{t_4-t_3} - v_{turn} \frac{t-t_4}{t_4-t_3} & \text{if } t_3 \leq t \leq t_4 \end{cases} \quad (18)$$

with t_1 the time instant when the vehicle is separated a distance $\Delta x = \frac{v_{path}^2 - v_{turn}^2}{2 \cdot a_{brake}} + LA$ to the next waypoint. $\Delta x - LA$ is indeed the required distance to reduce the velocity from v_{path} to v_{turn} with constant deceleration a_{brake} , which requires total amount of $t_2 - t_1$. In addition to this, t_3 is the time instant when the convergence criteria indicates that the vehicle has converged to the line between current waypoint and next waypoint and t_4 is the required time to accelerate from v_{turn} to v_{path} with constant acceleration a_{path} .

3.2 Dynamic Controller

In order to increase driving safety and comfort and reduce slippage, many studies have been carried out to determine appropriate values for super-elevation and side friction factor in horizontal road alignment (Krammes and Garnham, 1998; Marchionna and

Perco, 2007). These studies also establish appropriate values for clothoid sharpness in transition curves, according to $\kappa_R(t) \in [-\kappa_{max}, \kappa_{max}]$ and $\sigma(t) \in [-\sigma_{max}, -\sigma_{min}] \cup [\sigma_{min}, \sigma_{max}]$, to guarantee specific vehicle dynamic properties. In fact, in order to ensure appropriate curve visibility and comfort, one of the most accepted criteria in clothoid sharpness design (Marchionna and Perco, 2007) is related to the maximum curvature, such $\kappa_{max}^2 \equiv \sigma_{min} < \sigma < \sigma_{max} \equiv 9 \cdot \kappa_{max}^2$. The clothoid sharpness is selected according to the following formula:

$$\sigma = (1 - \alpha) \cdot \sigma_{min} + \alpha \cdot \sigma_{max} \quad (19)$$

where α is a design parameter.

The heuristic criteria used in the computation of α has the purpose of reducing the sharpness when the vehicle is following a straight line with high speeds, while letting it turn by increasing sharpness when necessary. Such α parameter is obtained using the following formula $\alpha = H \cdot \sin \beta_{min} \frac{v_{turn}^2}{v_R LA}$, with H a time horizon (in our case $H = 1$ s) and $\beta_{min} = \min(\beta, \pi/2)$, being $\beta = \arctan(\frac{y_T - y_R}{x_T - x_R}) - \theta_R$. Notice that α depends on v_{turn} and LA , which are included because vehicle's behaviour is sensitive to these parameters.

At this point, several criteria can be considered such as vehicle stability carrying a heavy load, horizontal alignment, slippage avoidance and/or maximum curvatures due to mechanical constraints:

$$\kappa_{max} = \min(\kappa_{max,stab}, \kappa_{max,horiz}, \kappa_{max,slip}, \kappa_{max,mech}, \dots) \quad (20)$$

$$\sigma_{max} = \min(\sigma_{max,stab}, \sigma_{max,horiz}, \sigma_{max,slip}, \sigma_{max,mech}, \dots) \quad (21)$$

$$\sigma_{min} = \max(\sigma_{min,stab}, \sigma_{min,horiz}, \sigma_{min,slip}, \sigma_{min,mech}, \dots) \quad (22)$$

where in our notation subscript name is a short name for the criterion that can be used.

Focusing on stability criterion, Figure 8 defines main geometric parameters involved in lateral and frontal tip-overs. It also shows a picture with the real forklift were values for these parameters have been taken from. Our notation considers sub-index B for body and sub-index L for load.

Let first compute vectors for body and load positions, velocities and accelerations for a given configuration (see Figure 8(d)). Let $\vec{\mathbf{p}}_R = [x_R, y_R, 0]^T \in \mathfrak{R}^3$ be the vehicle Cartesian coordinates where control over its position is to be performed. The position for the body center of mass $\vec{\mathbf{p}}_B$ and load center of mass $\vec{\mathbf{p}}_L$ is:

$$\vec{\mathbf{p}}_B = \vec{\mathbf{p}}_R - L_B \vec{\mathbf{t}} + 0 \vec{\mathbf{n}} + h_B \vec{\mathbf{z}} \quad (23)$$

$$\vec{\mathbf{p}}_L = \vec{\mathbf{p}}_R + L_L \vec{\mathbf{t}} + 0 \vec{\mathbf{n}} + h_L \vec{\mathbf{z}} \quad (24)$$

being $\{\vec{\mathbf{t}}, \vec{\mathbf{n}}, \vec{\mathbf{z}}\}$ a base of orthonormal vector corresponding to the tangential, normal and binormal directions of the vehicle reference frame. Velocities and accelerations can be easily derived:

$$\vec{\mathbf{v}}_B = v_{B_t} \vec{\mathbf{t}} + v_{B_n} \vec{\mathbf{n}} = v_R \cdot \vec{\mathbf{t}} - L_B \cdot v_R \cdot \kappa_R \cdot \vec{\mathbf{n}} \quad (25)$$

$$\vec{\mathbf{v}}_L = v_{L_t} \vec{\mathbf{t}} + v_{L_n} \vec{\mathbf{n}} = v_R \cdot \vec{\mathbf{t}} + L_L \cdot v_R \cdot \kappa_R \cdot \vec{\mathbf{n}} \quad (26)$$

$$\vec{\mathbf{a}}_B = a_{B_t} \vec{\mathbf{t}} + a_{B_n} \vec{\mathbf{n}} = (\dot{v}_R + L_B \cdot v_R^2 \cdot \kappa_R^2) \vec{\mathbf{t}} + (v_R^2 \cdot \kappa_R - L_B (\dot{v}_R \cdot \kappa_R + v_R \cdot \dot{\kappa}_R)) \vec{\mathbf{n}} \quad (27)$$

$$\vec{\mathbf{a}}_L = a_{L_t} \vec{\mathbf{t}} + a_{L_n} \vec{\mathbf{n}} = (\dot{v}_R - L_L \cdot v_R^2 \cdot \kappa_R^2) \vec{\mathbf{t}} + (v_R^2 \cdot \kappa_R + L_L (\dot{v}_R \cdot \kappa_R + v_R \cdot \dot{\kappa}_R)) \vec{\mathbf{n}} \quad (28)$$

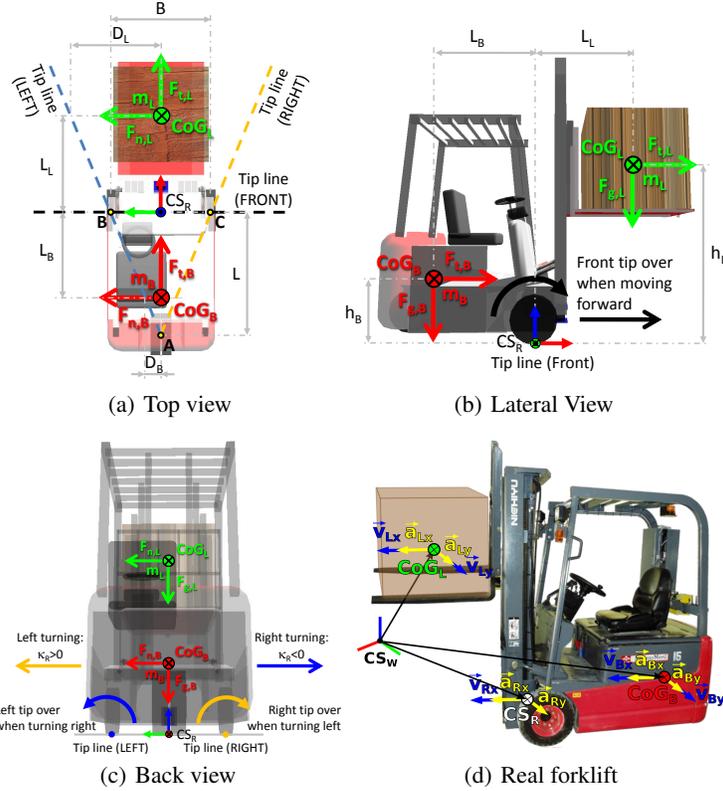


Fig. 8. Different views of the industrial forklift carrying a load. CS stands for Coordinate System, while subscripts B, L, R and W refer to Body, Load, Robot and World respectively.

being $\vec{t} = v_R \cdot \kappa_R \cdot \vec{n}$, $\vec{n} = -v_R \cdot \kappa_R \cdot \vec{t}$ and $\vec{z} = 0$ for a planar motion.

From equilibrium condition, we can get the limit of lateral tip-over stability. As shown in Figure 8(c), the sign criteria for the equilibrium problem is different depending on the turning direction (left and right tip lines), which turns in fact into the following inequalities depending on the case:

$$D_B \cdot m_B \cdot g + D_L \cdot m_L \cdot g - h_L \cdot m_L \cdot a_{L_n} - h_B \cdot m_B \cdot a_{B_n} > 0 \quad \text{if } \kappa_R > 0 m^{-1} \quad (29)$$

$$D_B \cdot m_B \cdot g + D_L \cdot m_L \cdot g + h_L \cdot m_L \cdot a_{L_n} + h_B \cdot m_B \cdot a_{B_n} > 0 \quad \text{if } \kappa_R < 0 m^{-1} \quad (30)$$

with g the gravitational acceleration, m_B and m_L the masses of body and load, $D_B = \frac{B}{2} \cdot \frac{L-L_B}{L}$ and $D_L = \frac{B}{2} \cdot \frac{L+L_L}{L}$ represent distances of the supporting triangle formed with the two front wheels and the rear wheel, L_B and L_L are longitudinal distances from vehicle coordinate system to base and load's centres of gravity respectively, h_B and h_L are heights for base and load's centres of gravity with respect to the floor, B is the front wheel separation and L is the wheel base, i.e. the axle distance.

Assuming that an inertial sensor (IMU) is located in the vehicle body's CoG and from Eq. (28), we can marginalize out κ_R from a_{L_n} component and substitute in the

marginalization of a_{L_n} in Eqs. (29) and (30). Thus, we can find out maximum normal load acceleration and as a consequence it is also possible to compute the maximum allowed curvature that guarantees instantaneous lateral stability:

$$\kappa_{max,stab} = \left| \frac{(D_L \cdot m_L + D_B \cdot m_B) \cdot g - \text{sign}(\kappa_R) \cdot (h_B \cdot m_B \cdot a_{B_n} + v_R \cdot \dot{\kappa}_R \cdot L_L)}{h_L \cdot m_L \cdot (v_R^2 + L_L \cdot \dot{v}_R)} \right| \quad (31)$$

where a_{B_N} is taken from the IMU's measurement.

Similar to horizontal alignment criteria Krammes and Garnham (1998), the maximum sharpness from the lateral stability criteria is obtained by the following expression:

$$\sigma_{min,stab} = \kappa_{max,stab}^2 \text{ and } \sigma_{max,stab} = 9 \cdot \sigma_{min,stab}.$$

In order to avoid frontal tip-overs due to excessive braking, we also analyse conditions for tangential equilibrium equations when braking:

$$L_B \cdot m_B \cdot g - L_L \cdot m_L \cdot g - h_B \cdot m_B \cdot a_{B_t} - h_L \cdot m_L \cdot a_{L_t} > 0 \quad (32)$$

Taking into account Eq. (32) and (28), we can similarly proceed to derive a frontal stability condition which implies to compute maximum allowed deceleration:

$$a_{brake} \leq \frac{(L_B \cdot m_B - L_L \cdot m_L) \cdot g + h_B \cdot m_B \cdot a_{B_t}}{h_L \cdot m_L} + v_R^2 \cdot \kappa_R^2 \cdot L_L \quad (33)$$

4 Simulation Results

In order to validate our proposal, we have simulated a forklift and an environment using ROS-Gazebo with parameter values corresponding to the real forklift shown in Figure 8(d): body mass $m_B = 1500\text{kg}$, load mass $m_L = 1000\text{kg}$, with $L_B = 1.0\text{m}$, $D_B = 0.12\text{m}$, $h_B = 0.42\text{m}$, $L_L = 1.01\text{m}$, $D_L = 0.924\text{m}$, $h_L = 2.75\text{m}$. Simulation step time has been set to $t_{sim} = 10^{-3}\text{s}$ and control period $T_R = 0.01\text{s}$, that is, the sampling time in which DCC paths are recomputed. Additional parameters are: gravity acceleration $g = 9.81\text{m/s}^2$, variables related with joints and constraints for dynamic simulations based on ODE (Open Dynamics Engine), such as constraint force mixing $cfm = 0.0001$ and error reduction parameter $erp = 0.4$ as global parameter settings and damping $damp = 0.1$ and friction $fric = 10.0$ for every joint. On the other hand, a tricycle kinematic model of the vehicle has been used, with a rear steering wheel and two front driving wheels: front wheel separation $B = 1.04\text{m}$, wheel base or axle distance $L = 1.3\text{m}$ and wheel diameter $D = 0.44\text{m}$. From the robot configuration we can set the maximum mechanical curvature as $\kappa_{max,mech} = R_{min}^{-1} = \tan(\Phi_{max})/L = 0.7692\text{m}^{-1}$, where R_{min} is the minimum radius of curvature, L is the wheel base and $\Phi_{max} = \pi/4\text{rad}$ is the maximum steering angle of the rear wheel.

First, we show the advantages of using the proposed hybrid-controller described in previous section. For that purpose, we have implemented components of Figure 7 in ROS and will be separately tested first. The implementation allows activation and deactivation of nodes and re-parametrization based on ‘‘dynamic_reconfigure’’ ROS package. The EKF implementation is based on the ‘‘robot_pose_ekf’’ ROS package, where only odometry and inertial values are considered¹.

¹ We are not concerned about the dead-reckoning problem, since it is out of our scope.

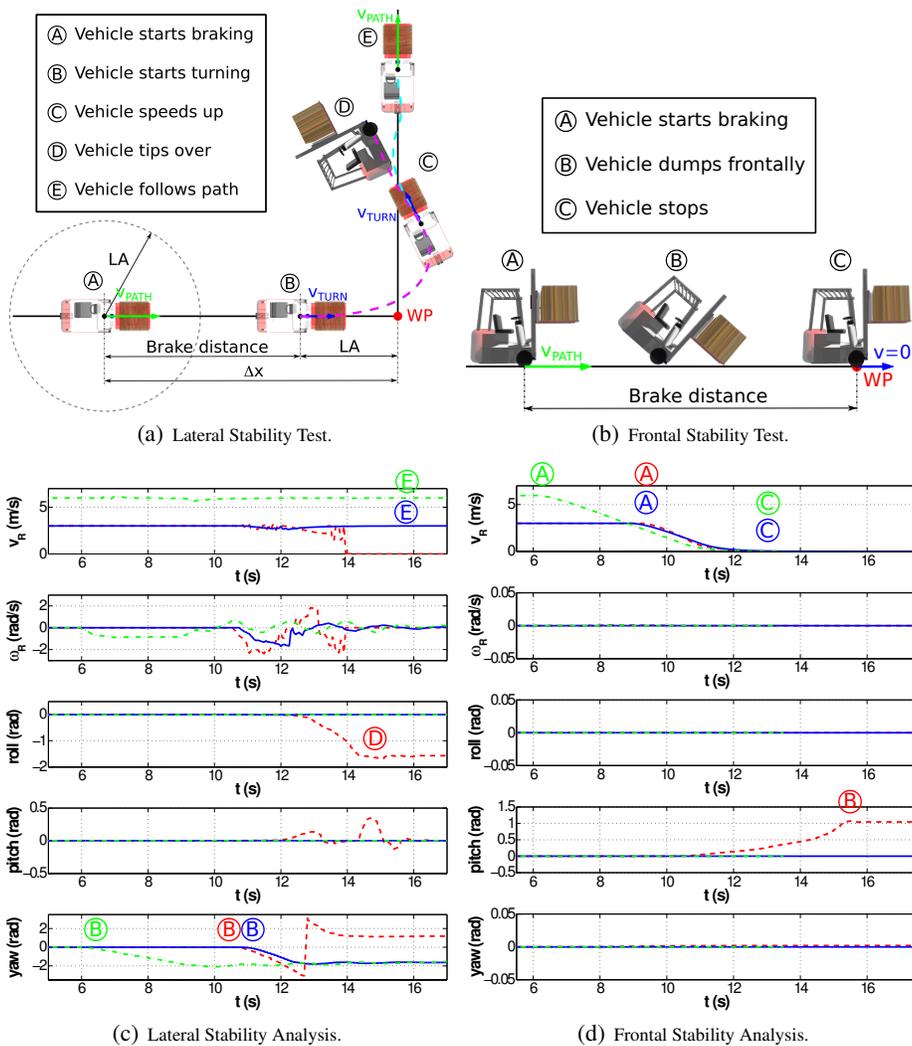


Fig. 9. (Up) Sequences explaining experiments performed in the stability controller analysis, (a) lateral test and (b) frontal test. (Down) Stability controller analysis, (c) lateral stability with $v_{turn} = v_{path}$ and (d) frontal stability. Solid blue line corresponds to stability controller set to ON with $v_{path} = 3\text{m/s}$, dashed red line is for the OFF state (where it starts failing) and dash-dotted green line corresponds to the maximum velocity case $v_{path} = v_{max} = 6\text{m/s}$.

In our first simulation scenario, the vehicle is moving almost at constant velocity $v_{path} = 3\text{m/s}$, but it needs to make a 90° turn (see Figure 9(a)). Thus, the aim of this test is to compare the behaviour of the vehicle when speed profile generator and frontal stability nodes are not active and when they are. If lateral stabilization node is off, that is, minimum and maximum sharpness and maximum curvatures are not recomputed based on IMU measurements, the DCC path generator cannot perform successfully when carrying heavy loads. Indeed, the vehicle starts turning at $t \approx 11\text{s}$ with $v_R = 3\text{m/s}$ and it tips over around $t \approx 14\text{s}$, as shown in Figure 9(c) (dashed red line). It has to be mentioned that $v_{path} = 3\text{m/s}$ and $LA = 3\text{m}$ is the limit situation where the vehicle starts tipping over. However, if we activate the lateral stabilization node, it can be clearly seen that the vehicle does not tip over (blue solid line in Figure 9(c)). This can be proved even at the maximum velocity taken from real vehicle specifications $v_{max} = 6\text{m/s}$, where our stabilization controller still performs successfully, as depicts Figure 9(c) (green dash-dotted line). Full simulations can be seen in Video 4 in Table 1 in Appendix C.

Similarly, we can analyse the effect of enabling or disabling the frontal stabilization module, in which maximum allowed decelerations are recomputed. In this case, the simulation scenario includes the vehicle moving straight forward during a certain amount of time with $v_{path} = 3\text{m/s}$ and then it performs a brake, that is, it sets velocity to zero (see Figure 9(b)). Figure 9(d) shows the case in which a constant deceleration is taken as the maximum vehicle deceleration with $a_{brake} = 2\text{m/s}^2$ for the case where the vehicle is carrying a load, where it is clearly shown that the vehicle destabilized during braking until it dumped frontally at $t \approx 15\text{s}$ (dashed red line). In addition to this, Figure 9(d) also clearly shows that with frontal stabilization on, the vehicle can successfully perform the brake (solid blue line). We can also see in Figure 9(d) that the stabilization controller still performs successfully at the maximum velocity case (dash-dotted green line). Full simulations can be seen in Video 5 in Table 1 in Appendix C.

In addition to this, Figure 10 shows a scenario where the vehicle is following a path with a squared side of 30m . Two different kinematic controllers are tested: blue line corresponds to our DCC path generator method, while dashed-dotted line corresponds to Pure-pursuit (PP) method (Wallace et al., 1985; Ollero and Heredia, 1995; Ollero, 2001) a well known algorithm commonly used as a curvature controller which proposes circular arcs to reach targets. In both cases, lateral and frontal stabilization nodes are active and the unique difference is on how the curvature profile is computed for a given target. What we are interested to highlight is that if the vehicle has not properly converged after performing a turn with the appropriate heading direction, well-known policies such as “slow-in” and “fast-out” can potentially cause oscillation problems. This is the case of PP method in Figure 10, in which the angular rate increases as a consequence of increasing the velocity. This also may cause additional inconvenience such as increased odometry errors due to oscillatory behaviour. This error is computed as the accumulate value of error between the relative ground-truth pose (provided by Gazebo) and the relative estimated pose. Video 3 of Table 1 in Appendix C shows an illustrative example of problems caused due to over-oscillation in an factory environment.

Moreover, we analyse the main characteristics of the proposed method against the PP method. Again, without loss of generality, the following analysis has been deve-

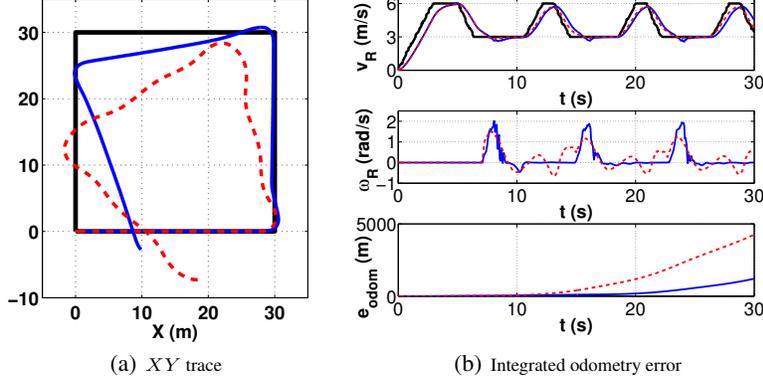


Fig. 10. DCC (blue) vs. PP (dashed red), following a square path of 30m side, with $LA = 4.5\text{m}$, $v_{path} = 6\text{m/s}$ and $v_{turn} = 3\text{m/s}$.

loped for the case in which the initial robot configuration is $\mathbf{q}_R = (0, 0, 0, 0, 0)^T$, with $v_R = 0\text{m/s}$, and the path to follow is a straight line $\rho_l = x \cos \phi_l + y \sin \phi_l$ with $\rho_l = 20\text{m}$ and $\phi_l = 0\text{rad}$ (i.e. line $x = 20\text{m}$), which represents one of the most difficult cases for both algorithms since it represents a 90° turn. Our purpose is to analyse the sensibility of several metrics (described below) for each method against Look-Ahead distance LA parameter and the robot velocity v_R . Initially, the robot accelerates in a straight movement until it reaches the path reference speed and when the robot is at a distance LA to the path begins to twist until converges to the line. All simulations have been carried out with a forklift carrying a heavy load at a height of $h_L = 1.75\text{m}$.

To evaluate the performance, several metrics have been used to characterize the resulting path $\{[x_{R,0}, y_{R,0}]^T, \dots, [x_{R,N}, y_{R,N}]^T\}$, where N is the number of points of the simulation. In particular, we consider the following metrics: overshoot $\delta = \max\{\frac{x_{R,i} - \rho_l}{LA}\}$, settling time t_s such that $|\frac{x_{R,i} - \rho_l}{LA}| < 0.05$ and mean error $\bar{e} = \sum_i |\rho_l - (x_{R,i} \cos \phi_l - y_{R,i} \sin \phi_l)|$, with $x_{R,i}$ and $y_{R,i}$ the i -th position of the simulated path. In addition to this, we also consider additional metrics taking into account comfort and safety, so we evaluate the mean normal acceleration $\bar{a}_N = \sum_i |a_{N,i}|$, the mean curvature $\bar{\kappa} = \sum_i |\kappa_{R,i}|$ and the maximum curvature $\kappa_{max} = \max\{\kappa_{R,i}\}$, being $a_{N,i}$ the normal acceleration along the path measured with the IMU and $\kappa_{R,i}$ the robot curvature along the path obtained from odometry data. Such metrics values are reflected on Figure 11(a), for different values of LA parameter and Figure 11(b) shows path trace for specific values of LA parameter. It can be appreciated that as long as the LA parameter is smaller, the PP method presents an oscillatory behaviour. In all presented cases, the convergence time for the DCC method is smaller than the PP method as well as the mean error. Also the mean normal acceleration and curvature are lower because we have, in general, lower oscillatory effect. Our maximum curvature is always higher, which implies that we are being less conservative than PP method, which implies that our controller is able to bring the vehicle to the limit conditions while taking

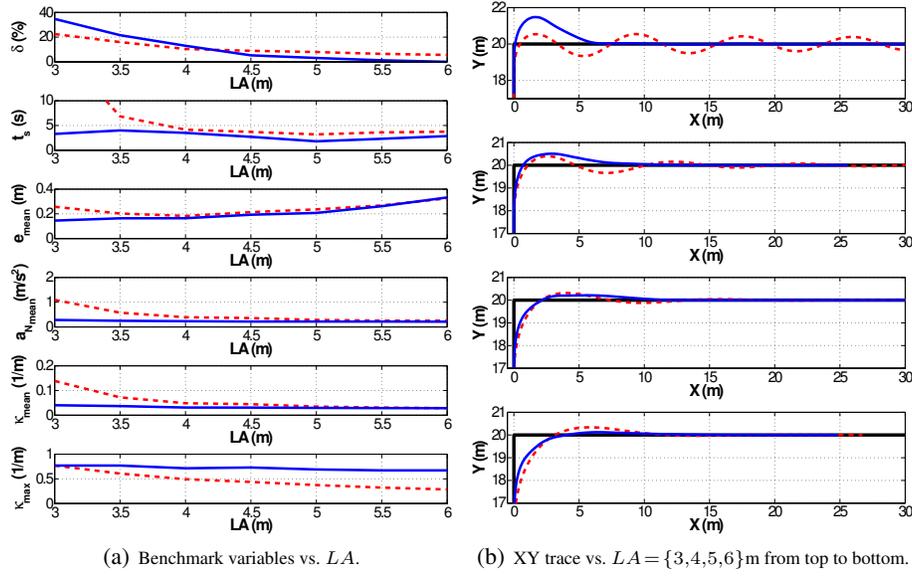


Fig. 11. DCC (blue) vs. PP (dashed red), for different values of LA , with $v_R = 3\text{m/s}$.

into account stabilization conditions. Figure 12 shows the results for different velocity conditions, where similar conclusions can be drawn. Our method performs better in terms of settling time, mean error, mean normal acceleration and curvature and maximum curvature. Here the effect is contrary to the LA parameter, that is, PP method produces higher oscillations with increasing velocity values.

5 Conclusions

The paper has introduced a closed-loop hybrid control for autonomous-guided industrial vehicles (AGVs), where vehicle stability, safety, slippage and comfort have been taken into account. From a theoretical point of view, the paper has described a complete method for generating continuous-curvature paths for non-holonomic wheeled mobile robots, and introduces a hybrid controller composed by both a kinematic and a dynamic controller.

The method computes Double Continuous-Curvature (DCC) paths from any arbitrary starting and target configurations, based on clothoids, as a reference to the kinematic controller. A heuristic criterion has been implemented to find the shortest path joining both configurations. Target configurations are obtained from a sequence of waypoints taken from a global path planner or a vision-based line following system.

The kinematic controller computes trajectories from a combination of DCC paths and speed profiles, based in a “slow-in” and “fast-out” policy. A closed-loop feedback structure is implemented in order to improve robustness, recomputing curvatures and speed profiles, based on sensor data.

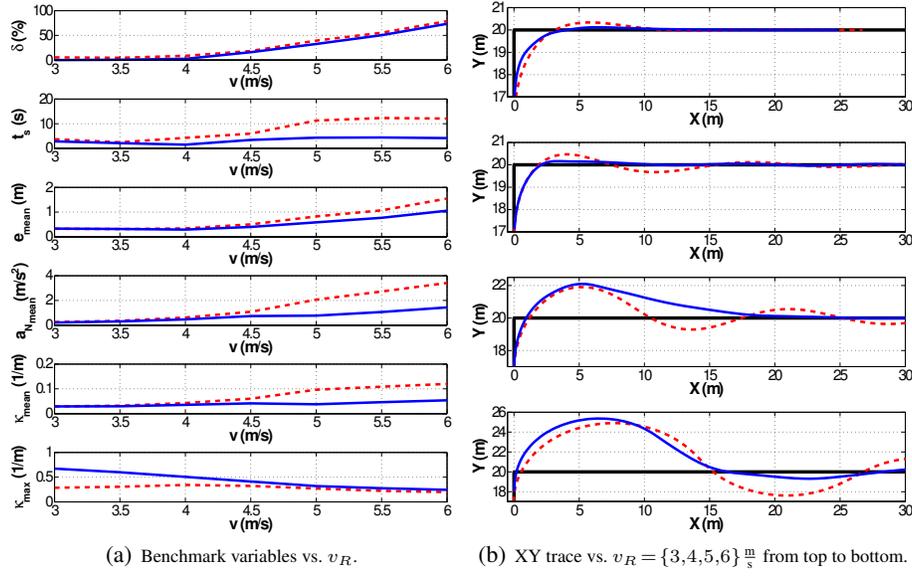


Fig. 12. DCC (blue) vs. PP (dashed red), for different values of v_R , with $LA = 6\text{m}$

Moreover, the proposed control structure includes a dynamic controller to guarantee vehicle stability conditions. The dynamic controller establishes bounds for DCC paths so that vehicle stability, safety and comfort are satisfied. The dynamic controller also includes a closed-loop feedback structure, where odometry and IMU data are used to stabilize the vehicle against tip-overs.

The method has been validated in different simulated scenarios, using an industrial forklift carrying heavy loads at high speeds. In simulations, lateral and frontal tip-overs have been analysed showing that our method is able to control the vehicle carrying a load of 1000kg at 2.75m height and 6m/s speed, the maximum vehicle speed for the considered forklift. Without the dynamic controller the vehicle tips over at 3m/s speed.

The paper also provides an exhaustive analysis to evaluate the performance of the new DCC kinematic method with respect to the conventional Pure-Pursuit (PP) method. It is interesting to remark that the PP method tends to oscillate the closer the target is selected (short Look-Ahead distances). These oscillations may cause instability and eventually frontal or lateral tip-overs, as well as crashing when moving in narrow corridors.

In addition, several metrics have been considered for evaluation of the performance, such as overshoot, settling time, safety and comfort. It is shown that the proposed method achieves better results. The set of attached videos also illustrates the advantages of the proposed method.

6 Acknowledgements

This work was supported by VALi+d and PROMETEO Programs (Conselleria d'Educació, Generalitat Valenciana), DIVISAMOS (DPI-2009-14744-C03-01) and SAFEBUS (IPT-2011-1165-370000): Ministry of Economy and Competitivity.

Bibliography

- C. Beck, J. Valls, and G. Dissanayake. Trajectory optimisation for increased stability of mobile robots operating in uneven terrains. In *IEEE Int. Conf. on Control and Automation*, pages 1913–1919, 2009.
- Lester. E. Dubins. On curves of minimal length with constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79 (3):497–516, July 1957.
- T. Fraichard and R. Mermond. Path planning with uncertainty for car-like robots. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 1, pages 27–32 vol.1, 1998. doi: 10.1109/ROBOT.1998.676244.
- Th. Fraichard and A. Scheuer. From reeds and shepp’s to continuous-curvature paths. In *IEEE Trans. on Robotics and Automation*, number 6, pages 1025–1035 vol. 20, 2004.
- V. Girbés, L. Armesto, and J. Tornero. Pisala project: Intelligent sensorization for line tracking with artificial vision. In *International Symposium on Robotics*, pages 558–563, 2010.
- V. Girbés, L. Armesto, and J. Tornero. On generating continuous-curvature paths for line following problem with curvature and sharpness constraints. In *IEEE Int. Conf. on Robotics and Automation*, pages 6156–6161, May 2011a.
- V. Girbés, L. Armesto, and J. Tornero. Continuous curvature control of mobile robots with constrained kinematics. In *IFAC World Congress*, pages 3503–3508, 2011b.
- K. Jiang, D.Z. Zhang, and L.D. Seneviratne. A parallel parking system for a car-like robot with sensor guidance. *Journal of Mechanical Engineering Science*, 213(6):591–600, 1999. ISSN 0954-4062.
- M.H. Korayem, H. Ghariblu, and A. Basu. Maximum allowable load of mobile manipulators for two given end points of end effector. *Int. J. Adv. Manuf. Technol*, 24:743–751, 2004.
- V. Korayem, M. H.Azimirad, A. Nikoobin, and Z. Boroujeni. Maximum load-carrying capacity of autonomous mobile manipulator in an environment with obstacle considering tip over stability. *Int. J. Adv. Manuf. Technol.*, 46:811–829, 2010.
- R. A. Krammes and M. A. Garnham. Worldwide review of alignment design policies. In *Int. Symp. on Highway Geometric Design Practices*, pages 19:1–17, 1998.
- L. Labakhua, U. Nunes, R. Rodrigues, and F. S. Leite. Smooth trajectory planning for fully automated passengers vehicles. spline and clothoid based methods and its simulation. *Springer: Informatics in Control Automation and Robotics*, 2008.
- J.-P. Laumond, P.E. Jacobs, M. Taix, and R.M. Murray. A motion planner for nonholonomic mobile robots. *Robotics and Automation, IEEE Trans. on*, 10(5):577–593, oct. 1994. ISSN 1042-296X. doi: 10.1109/70.326564.
- Yugang Liu and Guangjun Liu. Interaction analysis and online tip-over avoidance for a reconfigurable tracked mobile modular manipulator negotiating slopes. *IEEE/ASME Transactions on Mechatronics*, 5(4):623 – 635, August 2010.
- M. Manz, F. von Hundelshausen, and H.-J. Wuensche. A hybrid estimation approach for autonomous dirt road following using multiple clothoid segments. In *IEEE Int. Conf. on Robotics and Automation*, pages 2410–2415, 2010.
- A. Marchionna and P. Perco. A proposal to update the clothoid parameter limiting criteria of the italian standard. In *Int. Società Italiana Infrastrutture Viarie Congress*, 2007.
- J. Minguez, F. Lamiroux, and J.P. Laumond. *Handbook of Robotics*, chapter 35. Motion Planning and Obstacle Avoidance. Springer, 2008.
- N. Montés, M.C. Mora, and J. Tornero. Trajectory generation based on rational bezier curves as clothoids. In *IEEE Intelligent Vehicles Symposium*, pages 505–510, 2007.

- A. Ollero. *Robótica. Manipuladores y robots móviles*. Marcombo, S.A., Barcelona, 1 edition, 9 2001.
- A. Ollero and G. Heredia. Stability analysis of mobile robot path tracking. In *Int. Conf. on Intelligent Robots and Systems*, pages 461–466 vol.3, 1995.
- I. Papadimitriou and M. Tomizuka. Fast lane changing computations using polynomials. In *American Control Conf.*, pages 48–53 vol.1, 2003.
- J.A. Reeds and L.A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990.
- A. Scheuer and T. Fraichard. Collision-free and continuous-curvature path planning for car-like robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 867–873 vol.1, 1997a.
- A. Scheuer and Th. Fraichard. Planning continuous-curvature paths for car-like robots. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1304–1311 vol.3, 1996.
- A. Scheuer and Th. Fraichard. Continuous-curvature path planning for car-like vehicles. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 997–1003 vol.2, 1997b.
- A. Scheuer and M. Xie. Continuous-curvature trajectory planning for manoeuvrable non-holonomic robots. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1675–1680 vol.3, 1999.
- Steven H. Schot. Jerk: The time rate of change of acceleration. *American Journal of Physics*, 46: 1090–1094, 1978.
- J. Valls, G. Dumonteil, C. Beck, and G. Dissanayake. A kyno-dynamic metric to plan stable paths over uneven terrain. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 294–299, oct. 2010.
- R. Wallace, A. Stentz, C. E. Thorpe, H. Moravec, W. Whittaker, and T. Kanade. First results in robot road-following. In *Int. Conf. on Artificial Intelligence*, 1985.
- Doran K. Wilde. Computing clothoid segments for trajectory generation. In *Int. Conference on Robotics and Automation*, 2009.
- W.U. Xinye, G.E. Xiaohong, and H. Hongwu. Study on vehicle rollover avoidance. In *IEEE Int. Conf. on Measuring Technology and Mechatronics Automation*, pages 681–684, 2010.
- K. Yang and S. Sukkariéh. Real-time continuous curvature path planning of uavs in cluttered environments. In *Int. Symp. on Mechatronics and its Applications*, pages 1–6, 2008.

A Clothoid-based curve properties

Definition Cornu’s Spiral or Clothoid is defined by the Fresnel integrals in \mathbb{R}^2 as follows:

$$\mathcal{C}(\gamma) = \begin{bmatrix} C_x(\gamma) \\ C_y(\gamma) \end{bmatrix} = K \begin{bmatrix} \int_{\gamma_i}^{\gamma} \cos(\frac{\pi}{2}\xi^2)d\xi \\ \int_{\gamma_i}^{\gamma} \sin(\frac{\pi}{2}\xi^2)d\xi \end{bmatrix} \quad (34)$$

where K is the homothety factor, i.e.: the scale of the spiral, and $\gamma \in [\gamma_i, \gamma_f]$ comprises the integration interval between the initial and final configurations. Unfortunately, there is no closed-form solution to compute Fresnel integrals, however some interesting geometric properties of clothoids can be analytically computed.

Property Let $[\mathcal{C}(\gamma_i), \mathcal{C}(\gamma_f)]$ be a clothoid segment, the so called clothoid parameter A and its homothety factor are related by $K = \sqrt{\pi}A$. For given curvature range $\kappa \in [\kappa_i, \kappa_f]$ and sharpness σ the integration parameter is $\gamma = s_\sigma \kappa K / \pi$, with $s_\sigma = \text{sign}(\sigma)$. The cartesian coordinates are $x = C_x(\gamma)$ and $y = s_\sigma C_y(\gamma)$. The tangent angle τ with

respect to the abscissa axis \mathcal{X} of $\mathcal{C}(\gamma)$ is $\tau = s_\sigma \frac{\pi}{2} \gamma^2$. The curvature κ and length L of the clothoid $\mathcal{C}(\gamma)$ increase proportionally with γ for a given homothety factor, being the expression $\kappa = s_\sigma \pi \frac{\gamma}{K}$ for curvature and $L = K\gamma$ for length. with respect to the origin. It is straight forward to see that both, curvature and length are related by the clothoid parameter as $\kappa = \frac{L}{A^2}$, which implies that constant changes on the curvature are proportional to changes on the length of the curve.

Property Let $\mathcal{C}(\gamma)$ be point on a clothoid curve with a constant velocity v and the sharpness $\sigma \equiv A^{-2}$, the curvature derivative of $\mathcal{C}(\gamma)$ is constant and given by $\dot{\kappa} = v\sigma$. Clothoid derivatives can be analytically computed Schot (1978). The tangential and normal components of acceleration are $a_t = 0$ and $a_n = v^2\kappa$, respectively, while the tangential and normal components of rate of acceleration (jerk) are $j_t = -v^3\kappa^2$ and $j_n = v^2\dot{\kappa} = v^3\sigma$ respectively.

B Clothoid-based Double Continuous Curvature Paths

The end configuration of a DCC path \mathbf{q}_B can be computed as follows:

$$\mathbf{q}_B = \Delta\mathbf{q}_{l_A} \oplus \Delta\mathbf{q}_{A1} \oplus \Delta\mathbf{q}_{\Omega_A} \oplus \Delta\mathbf{q}_{A2} \oplus \Delta\mathbf{q}_{l_C} \oplus \Delta\mathbf{q}_{B1} \oplus \Delta\Omega_B \oplus \Delta\mathbf{q}_{B2} \oplus \Delta\mathbf{q}_{l_B} \oplus \mathbf{q}_A \quad (35)$$

where,

$$\Delta\mathbf{q}_{l_A} = [l_A \cos \theta_A \quad l_A \sin \theta_A \quad 0 \quad 0]^T \quad (36)$$

$$\Delta\mathbf{q}_{A1} = \left[\mathbf{R}(\theta_A) \begin{bmatrix} x_{A1} \\ s_A \cdot y_{A1} \end{bmatrix} \quad s_A \theta_{A1} \quad s_A |\kappa_A| \right]^T \quad (37)$$

$$\Delta\mathbf{q}_{\Omega_A} = \left[\kappa_A^{-1} \mathbf{R}(\theta_A + s_A \theta_{A1}) \begin{bmatrix} \sin \theta_{\Omega_A} \\ s_A (1 - \cos \theta_{\Omega_A}) \end{bmatrix} \quad s_A \theta_{\Omega_A} \quad 0 \right]^T \quad (38)$$

$$\Delta\mathbf{q}_{A2} = \left[\mathbf{R}(\theta_A + s_A (\theta_{A1} + \theta_{\Omega_A})) \begin{bmatrix} x_{A2} \\ s_A \cdot y_{A2} \end{bmatrix} \quad s_A \theta_{A2} \quad -s_A |\kappa_A| \right]^T \quad (39)$$

$$\Delta\mathbf{q}_{l_C} = [l_C \cos \theta_C \quad l_C \sin \theta_C \quad 0 \quad 0]^T \quad (40)$$

$$\Delta\mathbf{q}_{B1} = \left[\mathbf{R}(\theta_C) \begin{bmatrix} x_{B1} \\ s_B \cdot y_{B1} \end{bmatrix} \quad s_B \theta_{B1} \quad s_B |\kappa_B| \right]^T \quad (41)$$

$$\Delta\mathbf{q}_{\Omega_B} = \left[\kappa_B^{-1} \mathbf{R}(\theta_C + s_B \theta_{B1}) \begin{bmatrix} \sin \theta_{\Omega_B} \\ s_B (1 - \cos \theta_{\Omega_B}) \end{bmatrix} \quad s_B \theta_{\Omega_B} \quad 0 \right]^T \quad (42)$$

$$\Delta\mathbf{q}_{B2} = \left[\mathbf{R}(\theta_C + s_B (\theta_{B1} + \theta_{\Omega_B})) \begin{bmatrix} x_{B2} \\ s_B \cdot y_{B2} \end{bmatrix} \quad s_B \theta_{B2} \quad -s_B |\kappa_B| \right]^T \quad (43)$$

$$\Delta\mathbf{q}_{l_B} = [l_B \cos \theta_B \quad l_B \sin \theta_B \quad 0 \quad 0]^T \quad (44)$$

The Cartesian coordinates x_{A1} , y_{A1} , x_{A2} , y_{A2} , x_{B1} , y_{B1} , x_{B2} and y_{B2} are computed from Fresnel's integrals for their corresponding curvature κ_A or κ_B .

Cartesian elements of \mathbf{q}_B can be written as:

$$\mathcal{P}_x + \mathcal{X}_1 \cos \theta_C + \mathcal{X}_2 \sin \theta_C = l_A \cos \theta_A + l_B \cos \theta_B + l_C \cos \theta_C \quad (45)$$

$$\mathcal{P}_y - \mathcal{X}_2 \cos \theta_C + \mathcal{X}_1 \sin \theta_C = l_A \sin \theta_A + l_B \sin \theta_B + l_C \sin \theta_C \quad (46)$$

where,

$$\begin{aligned} \mathcal{P}_x = & x_B - x_A - x_{A1} \cos \theta_A - x_{B2} \cos \theta_B + s_A y_{A1} \sin \theta_A - s_B y_{B2} \sin \theta_B + \\ & + \kappa_A^{-1} \sin(\theta_A + \theta_{A1}) - s_B \kappa_B^{-1} \sin(\theta_B - s_B \theta_{B2}) \end{aligned} \quad (47)$$

$$\begin{aligned} \mathcal{P}_y = & y_B - y_A - s_A y_{A1} \cos \theta_A + s_B y_{B2} \cos \theta_B - x_{A1} \sin \theta_A - x_{B2} \sin \theta_B - \\ & - s_A \kappa_A^{-1} \cos(\theta_A + \theta_{A1}) + s_B \kappa_B^{-1} \cos(\theta_B - s_B \theta_{B2}) \end{aligned} \quad (48)$$

$$\mathcal{X}_1 = -x_{B1} - x_{A2} + \kappa_A^{-1} \sin \theta_{A2} (\cos^2 \theta_A + s_A \sin^2 \theta_A) + \kappa_B^{-1} \sin \theta_{B1} \quad (49)$$

$$\mathcal{X}_2 = s_B y_{B1} - s_A y_{A2} - \kappa_A^{-1} \cos \theta_{A2} (s_A \cos^2 \theta_A + \sin^2 \theta_A) + \kappa_B^{-1} s_B \cos \theta_{B1} \quad (50)$$

There are four unknowns and just two equations in (45) and (46), thus multiple solutions are possible. Since DCC path are applied within the path following problem, we force $l_A = 0$, otherwise convergence in path following problems can not be guaranteed. Therefore, the problem reduces to determine l_B , l_C and θ_C . In that case (45) and (46) can be rewritten as:

$$\mathcal{P}_x + \sqrt{(\mathcal{X}_1 - l_C)^2 + \mathcal{X}_2^2} \cos(\theta_C - \alpha) = l_B \cos \theta_B \quad (51)$$

$$\mathcal{P}_y - \sqrt{(\mathcal{X}_1 - l_C)^2 + \mathcal{X}_2^2} \sin(\theta_C - \alpha) = l_B \sin \theta_B \quad (52)$$

with $\tan \alpha = \frac{\mathcal{X}_2}{\mathcal{X}_1 - l_C}$. Equations (51) and (52) can be merged to provide a family of solutions for θ_C depending on the value l_C :

$$\theta_C = \arcsin \left(\frac{\mathcal{P}_x \sin \theta_B - \mathcal{P}_y \cos \theta_B}{\sqrt{(\mathcal{X}_1 - l_C)^2 + \mathcal{X}_2^2}} \right) + \arctan \left(\frac{\mathcal{X}_2}{\mathcal{X}_1 - l_C} \right) - \theta_B \quad (53)$$

In order to provide the shortest path l_C must satisfy the following:

$$l_C = \begin{cases} \max \left(0, \mathcal{X}_1 + \sqrt{(\mathcal{P}_x \sin \theta_B - \mathcal{P}_y \cos \theta_B)^2 - \mathcal{X}_2^2} \right) & \text{if } |\mathcal{P}_x \sin \theta_B - \mathcal{P}_y \cos \theta_B| < |\mathcal{X}_2| \\ 0 & \text{otherwise} \end{cases} \quad (54)$$

The angle θ_C is the analytic solution for the optimization problem stated in Section 2.2 when the solution implies arc segments. This value is used in any-case as starting point of the optimization procedure because it provides deflection angles which are “close” to the optimal ones.

C Index of Multimedia Data

Item #	Type	File
1	Video	Multimedia #1. DCC Generation.avi
2	Video	Multimedia #2. Line Following Real Forklift.avi
3	Video	Multimedia #3. PPvsDCC - Factory Crash.avi
4	Video	Multimedia #4. PPvsDCC - Lateral Stability Effect.avi
5	Video	Multimedia #5. PPvsDCC - Frontal Stability Effect.avi

Table 1. Multimedia Data files relation.