Continuous Curvature Path Planning for Differential Drive Robots

Mads Frier
Dept. of Electronic Systems
Aalborg University
Aalborg, Denmark
mfrier20@student.aau.dk

Asbjørn Lauritsen
Dept. of Electronic Systems
Aalborg University
Aalborg, Denmark
alau20@student.aau.dk

Frederik Saldern Nielsen
Dept. of Electronic Systems
Aalborg University
Aalborg, Denmark
fsni20@student.aau.dk

Victor Hoffmann Risager
Dept. Of Electronic Systems
Aalborg University
Aalborg Denmark
vrisag20@student.aau.dk

Anye Den Warren Igwacho
Dept. of Electronic Systems
Aalborg University
Aalborg, Denmark
aigwac23@student.aau.dk

***Abstract*—This paper presents a path planning algorithm for differential drive robots that need to follow trajectories precisely for line following tasks. The algorithm improves the geometric continuity of the path from $\mathbf{G^1}$ to $\mathbf{G^2}$ by using clothoids as transition curves between the segments of the Dubins path. Testing of the performance of the proposed algorithm in terms of continuity, completeness, optimality, and wheel acceleration were conducted. The results show the algorithm generates continuous curvature paths that increases the accuracy of the line following robot, but with a cost of longer trajectories. The parameters for the proposed algorithm considers the kinematic and dynamic constraints of the robot to estimate minimum turning radius for arcs and curvature rate for clothoids. Demonstrating the applicability of the algorithm for various line marking templates has been done to compare it with the unmodified Dubins path planner, to evaluate benefits and drawbacks with the developed algorithm.**

***Index Terms*—Robotics, Continuity, Trajectory planning, Line following, Dubins path, Clothoid**

I. Introduction

An increased need for fast but accurate line following has arisen in the realm of line following robots. Existing trajectory generation methods, which are based on Dubins paths for transitions, fail to achieve the desired level of geometric continuity, more specifically, while it may be a continuous path, the curvature of the path is discontinuous. As a result of the discontinuous curvature they fall short in satisfying the constraints of the robot, and it thus experiences undesired and infeasible high wheel acceleration, which results in a long transient responses in the path.

This paper aims to tackle the shortcomings of existing trajectory generation methods by proposing a novel path planning algorithm that considers the kinematic and dynamic constraints of the differential drive robot. More specifically, this is achieved by improving curvature continuity of the trajectory increasing the accuracy of the line following robot. Previous work has been conducted on the topic of path planning for line marking robots and improving this for paths. In the context of line marking robots, I. Hameed [1] proposes the usage of Dubins paths for transitions in a path planning algorithm used to paint a standard football field. This solution is not curvature continuous, thus A. Scheuer and T. Fraichard proposes a path planning algorithm in [2] based on Dubins paths, that uses clothoids as transition curves between the segments of the Dubins' segments to make it curvature continuous. They also propose curvature continuous paths for a Reeds & Shepp paths in [3], but both are not made in the context of line marking robots. Other methods to compute transition curves include Fermat's spirals as M. Candeloro et. al. proposes in [4], bezier curves are used in [5]. Even though the Bezier curves are simple to compute with a closed form solution and achieves the desired continuity, the viable configurations ends up being highly constrained. Another way of conducting path planning for differential drive robots, is by doing Balkcom & Mason curves [6], which is a stop and turn algorithm. The discontinuity of wheel velocities is the reason for disregarding this algorithm.

Throughout this paper, the methods and theory behind the proposed path planning algorithm are defined, with the necessary background knowledge in Section II. Here the contributions in path planning in relation to line marking robots will be enlightened. Subsequently the results are presented in Section III. Lastly the results are discussed in Section IV and concluded upon in Section V.

II. Methods

In this section the continuity of paths will be formally defined, where the exact desired level of continuity will be enlightened. Furthermore the proposed path planner algorithm is described, which is based on the results of the continuity definition.

A. Continuity

A path $\Pi(s)$ where $s(t), s \in [0, 1]$ is continuous of order n if it is connected at all transition points along the path

and differentiable everywhere, n times. This concept is called parametric continuity, and is defined by C^n.

- C^0: Π is connected at every point, but not differentiable.
- C^1: $\frac{\mathrm{d}\Pi}{\mathrm{d}s}$ is continuous and velocity vectors $\mathbf{T}$ at transition points. must be collinear and of *equal* magnitude. In addition C^0 is fulfilled.

While a path can be parametric continuous, it can still be discontinuous in the sense of curvature along the path. This is called geometric continuity of order n and is defined as G^n.

- G^0: Π is connected, but not differentiable.
- G^1: At a transition point between two curves, the tangent vectors $\mathbf{T}$ must be collinear, and with *proportional* magnitudes.
- G^2: G^1 is fulfilled and the curvature κ of Π is continuous.

The different levels of continuity are illustrated in Figure 1.

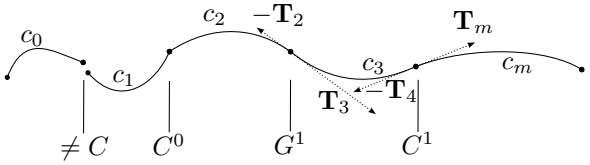


Fig. 1: Curve sequence continuity showing the different rates of continuity.

In a Dubins Path Planning (DPP) algorithm, only C^1 and G^1 continuity is achieved. This implies that the robot's velocity vector along the path is continuous, but the curvature (change of heading) is not. Consequently, the robot would require infinite angular acceleration at transition points, which is infeasible. Therefore the objective of this paper is to improve the geometric order of continuity of the path from G^1 to G^2.

B. Path Planner Algorithm

The goal of a path planner algorithm is to bring the robot from starting pose defined as the triplet $q_s = \langle x_s, y_s, \theta_s \rangle$ to a goal pose $q_g = \langle x_g, y_g, \theta_g \rangle$. In the context of line following robots, path planning is conducted between either disconnected or discontinuous path segments, and are referred to as transitions. Transitions are commonly conducted with the DPP algorithm. A Dubins path is based on three controls: driving straight (S), turning right (R), and turning left (L), from these curve segments the following Dubins configurations can be made: RSL, LSR, RSR, LSL, RLR, and LRL. The six configurations can be split into two classes: Curve-Curve-Curve (CCC) and Curve-Straight-Curve (CSC).

From these configurations, the optimal path for turning constrained vehicles can be found, from an arbitrary start- and goal pose. A. Giese explains in [7] how the optimal configuration is found geometrically. Two circles are created tangentially to the start- and goal pose with radius: $r = {}^1/_{\kappa_{max}}$, where κ_{max} is the maximum turning rate of the vehicle. The four tangentially supporting start- and goal circles are centred in $\mathbf{\Omega}_{sr}$, $\mathbf{\Omega}_{sl}$, $\mathbf{\Omega}_{gr}$, and $\mathbf{\Omega}_{gl}$ respectively.

From these tangentially supporting circles the optimal CSC Dubins configuration can be found, by spanning inner and outer tangents from each support circle to the opposing circles represented as $t1$, $t2$, $t3$, and $t4$ seen on Figure 2.

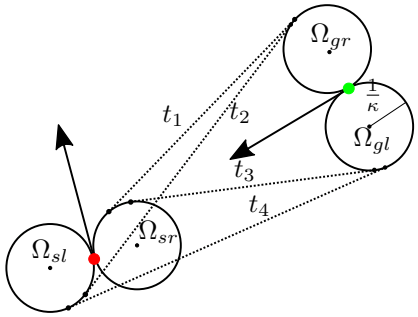


Fig. 2: Four tangents between support circles, the red- and green points are the start- and goal pose respectively.

A. Giese states the shortest of the four CSC configurations spanned by the tangents, guarantee optimality. Except that CCC is introduced if the euclidean distance between any two centres $||\mathbf{\Omega}_s\mathbf{\Omega}_\mathbf{g}|| < 4r$. With CCC configurations, a third circle tangent to both the support circles closer than $4r$ is used, as seen in Figure(3).

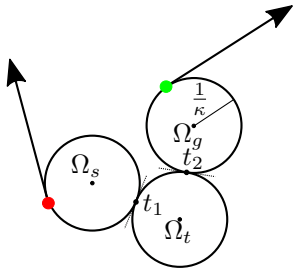
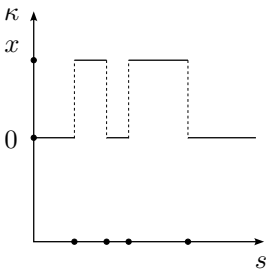
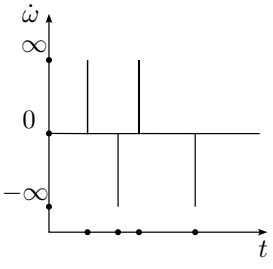


Fig. 3: CCC configuration: RLR path where Ω_t is the tangent circle connecting Ω_s and Ω_g. t_1 and t_2 are the tangent points.

The shortest configuration is then chosen by the shortest total parametric length of all 3 controls, ensuring optimality. As seen on Figure 4a, the Dubins configuration's curvature is G^1 continuous, but infinite wheel acceleration does occur, as seen on Figure 4b.



(a) Curvature graph of RSR trajectory.



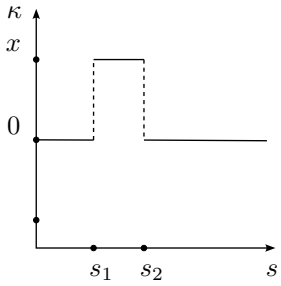
(b) Acceleration graph of the RSR trajectory.

Fig. 4: Figures showing the curvature and acceleration over time of an RSR configuration.

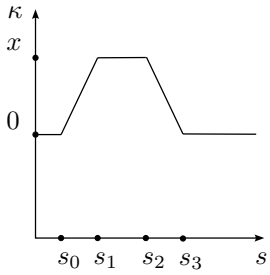
In order to ensure continuous curvature, the changes in κ must evolve continuously along s. Many spirals exists with this property but clothoids changes curvature linearly. Clothoids can assess the continuity problem in the Dubins path. The clothoid is computed for both axes, namely x and y of the Fresnel Integrals which can be seen in (1).

$$C(x) = \int_0^x \cos(\frac{\pi}{2}t^2)\,dt \qquad S(x) = \int_0^x \sin(\frac{\pi}{2}t^2)\,dt \quad (1)$$

When inserted between lines and arcs, the linear property of the clothoid can increase G^1 to G^2 continuity by connecting the curvature profile as seen in Figure 5.



(a) Curvature graph of the trajectory without transition.



(b) Curvature graph of the trajectory with transition.

Fig. 5: Comparison of a curvature with clothoid transition and without.

To implement a clothoid between lines and arcs making a single control continuous, consider the illustration in Figure 6 of a continuous left turn using clothoids.

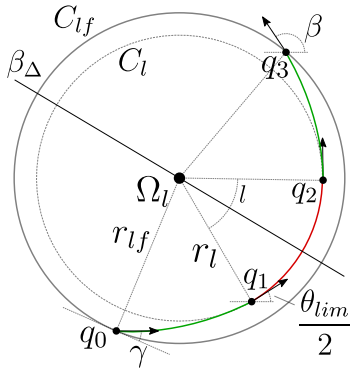


Fig. 6: Continuous left turn using clothoids, adapted from [2].

The left control is made from a null-configuration represented by the quadruplet: $q_0 = \langle x_0, y_0, \theta_0, \kappa_0 \rangle = \mathbf{0}$. From q_0, a clothoid can be made using the Fresnel integrals from (1). The clothoid is generated with sharpness/curvature rate σ_{max} which is computed from the maximum wheel accelerations. This enables the clothoid of length $l = {}^{\kappa_{max}}/_{\sigma_{max}}$ to end at curvature κ_{max}, since $\theta_{lim} = {}^{\kappa^2_{max}}/_{\sigma_{max}}$. q_1 can be found by Fresnel, seen in (2).

$$q_1 = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \\ \kappa_1 \end{bmatrix} = \begin{bmatrix} C\left(\sqrt{\frac{\theta_{lim}}{\pi}}\right)\sqrt{\frac{\pi}{\sigma_{max}}} \\ S\left(\sqrt{\frac{\theta_{lim}}{\pi}}\right)\sqrt{\frac{\pi}{\sigma_{max}}} \\ {}^{\theta_{lim}}/_2 \\ \kappa_{max} \end{bmatrix} \quad (2)$$

The deflection θ_1 is the difference between the start- and end heading of a single clothiod. $\theta_{lim} = 2\theta_1$ for the total turn deflection with zero arc length, which is the smallest possible deflection. The curvature of q_1 is κ_{max}, since the clothoid always end in maximum curvature. The inner circle on Figure 6, has radius given as ${}^1/_{\kappa_{max}}$. From q_1 to q_2 the red arc has a radius ${}^1/_{\kappa_{max}}$ and a centre point defined in (3). The radius of the outer circle where the clothoid poses q_0 and q_3 lie, is called $r_{lf} = \sqrt{x_\Omega^2 + y_\Omega^2}$, with centre point in Ω. The red maximum curvature arc on Figure 6, has a length of: $l = \beta - \theta_{lim}$.

$$\Omega = \begin{bmatrix} x_1 - \frac{sin\theta_1}{\kappa_{max}} \\ y_1 + \frac{cos\theta_1}{\kappa_{max}} \end{bmatrix} \quad (3)$$

The modified continuous controls are completely symmetrical around ${}^\beta/_2$, indicated by the symmetry-line β_Δ on Figure 6. From q_2, the last clothoid pose in the modified control can be found with an inverted clothoid. The clothoid start at $\kappa = \kappa_{max}$ and ends with $\kappa = 0$ with $-\sigma_{max}$, as seen in (4).

$$q_3 = \begin{bmatrix} x_3 \\ y_3 \\ \theta_3 \\ \kappa_3 \end{bmatrix} = \begin{bmatrix} -C\left(\sqrt{\frac{\theta_{lim}}{\pi}}\right)\sqrt{\frac{\pi}{-\sigma_{max}}} + x_2 \\ S(\sqrt{\frac{\theta_{lim}}{\pi}})\sqrt{\frac{\pi}{-\sigma_{max}}} + y_2 \\ \beta \\ 0 \end{bmatrix} \quad (4)$$

Note that inverting σ_{max} to $-\sigma_{max}$ results in an inverted clothoid spanning from q_2, with $\kappa_2 = \kappa_{max}$ in q_2 and $\kappa_3 = 0$ in q_3.

By changing the deflection angle β the continuous heading difference between the critical clothoid poses q_0 and q_3 can be manipulated.

The γ rotation constraint is the angle between the headings of either q_0 or q_3 and the tangent to the C_{lf} circle, seen on Figure 6. γ is always the same for both clothoid points on the control, and can be calculated by (5).

$$\gamma = -tan^{-1}\left(\frac{x_\Omega}{y_\Omega}\right) \quad (5)$$

It is important to note that the entry and exit poses are the only poses of interest, with respect to planning an entire continuous curvature Dubins configuration, as q_1 and q_2 can be found from them. The entry and exit poses for both the start- and goal control, resulting in a total of four clothoid poses, from which the modified clothoids will be spanned from. For

a right turn spanned from the unmodified q_0, a left turn is generated, and q_{1-3} are flipped around the axis spanned by the heading of q_0. The entry and exit poses of a normal Dubins configuration are rotated with $\pm\gamma$ with respect to the tangent line to C_{lf}. This is very important as γ can be used to apply the theory on a discontinuous Dubins planner. [2]

Now that the important property of γ has been revealed, the combination of Dubins path and clothoids can be considered. γ is used to offset the Dubins planner in order for it to take the necessary heading difference between the tangent to C_{lf} and γ into consideration, which is needed for the clothoid to hit the desired curvature of C_{lf}. This can be seen in Figure 7.

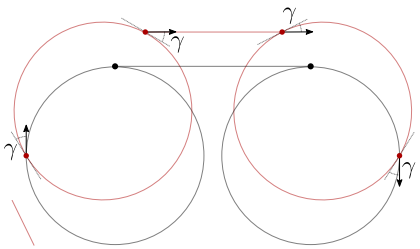


Fig. 7: Showcase of how a Dubins path is modified with γ to allow for clothoids. The red circles are C_{lf} and red poses are the critical poses.

The theory is applied for both entry and exit poses on C_{lf}. Combining the modifications to the Dubins planner will result in a path seen in figure 8. [2]

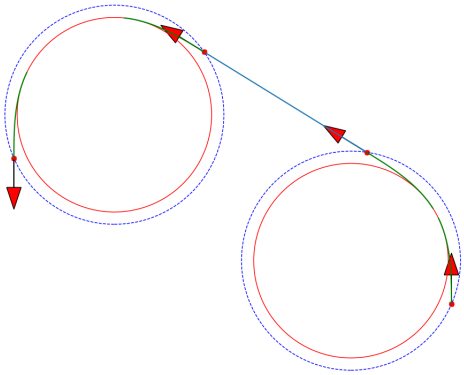


Fig. 8: Modified Dubins Path with continuous curvature using clothoids, with an LSL configuration. The inner circle is the minimum turning radius $1/\kappa_{max}$.

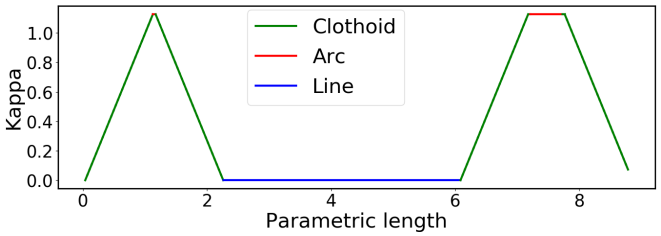
For the four different CSC Dubins configurations, it is necessary to change if γ is added or subtracted to the original Dubins planner. CCC configurations do not use tangents to find entry and exit points like CSC configurations, but at the single common intersection point between the two Dubins circles. Although, when making a continuous CCC configuration, the before mentioned γ discrepancy on how a continuous CSC configuration is found, is applied. It is important to note that the found entry and exit headings, regardless of configurations, are constrained between θ_{lim} and 2π as the path will not connect if this is not complied with. In [2] it is assumed that turns with deflection smaller than θ_{lim} does not exist. This is however based on the constraint that κ_{max} must be reached when making a turn or that σ must be fixed at σ_{max}.

III. Results

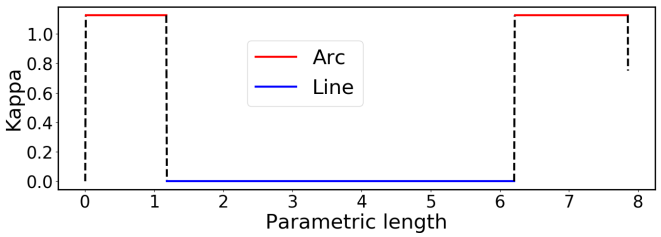
This section will delve into the performance of the new improved path planner for a Differential Drive robot. All testing of the algorithms have been conducted by simulation in Gazebo, with the Robot Operating System (ROS) and the visualisation program RVIZ. In which a simple pure-pursuit velocity PID controlled robot, follows the generated trajectories. Note that this is not the control algorithm running on the real robot, and is not as sophisticated and precise. When testing, the proposed Continuous Curvature Dubins path planner (CCD) algorithm will be tested against the original Dubins Path Planner (DPP) algorithm.

A. Continuity Test

The objective of this test is to verify C^1 and G^2 continuity of the path generated. To ensure this, 100 paths are generated with CCD and 100 with DPP between arbitrary start and goal poses. These poses are uniformly sampled from a 10×10 m area. For each path a curvature plot is generated and is inspected manually to verify G^2 continuity. An example of the curvature plot of a CCD- and DDP path can be seen in Figure 9a, and 9b, respectively.



(a) Curvature of a RSR CCD path.



(b) Curvature of the RSR DPP path.

Fig. 9: Example of curvature plots of the same start and end pose.

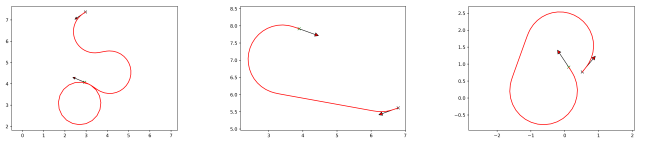
Testing for C^1 continuity was done by generating a full trajectory for a soccer field template and check if each segment is connected. Therefore, if the goal pose for segment Π_i is equal to start pose for segment Π_{i+1} ± 0.001 m, that connection is considered adequate. The results for the continuity tests can be seen in table I.

Continuity Test		
	CCD	DPP
C^1	100%	100%
G^2	100%	0%

TABLE I: Results from the continuity tests.

B. Completeness Test

This test checks whether the CCD is able to go from any start- to any goal pose. The test was conducted with 100 different trials, where the path planner had to calculate a path from a start to a goal pose, where checking if the path was generated successfully has been done manually. The poses are sampled uniformly from a 10×10 m area. Examples of the generated paths can be seen in figure 10.



(a) LRL configuration. (b) RSR configuration. (c) LSL configuration.

Fig. 10: Examples of the generated paths.

In table II, the results after 100 trials can be seen.

Completeness Test	
Complete	100%
Incomplete	0%

TABLE II: Results from the completenes test.

C. Robot Path Optimality Test

With the CCD, it is necessary to see if it improved the path length with respect to the Dubins planner. Table III highlights the measured path length at selected σ based on the plot seen in Figure 11. As a result, the path length depends on σ, where a lower σ increases path length and vice versa.

Optimality Test				
	σ_{max}	Path length m	Difference	Increase %
DPP	-	1653.55	-	0%
CCD	1.0	1915.15	+261.6	≈ 14
CCD	2.1	1781.59	+128.04	≈ 8

TABLE III: Results from the optimality test.

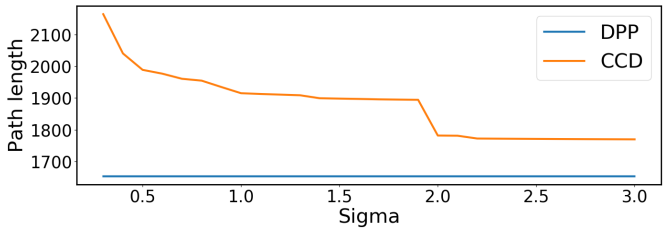


Fig. 11: Path lengths evaluated $\forall \sigma_{max} \in [0.3, 0.4 \ldots 2.9, 3.0]$.

D. Time Optimality Test

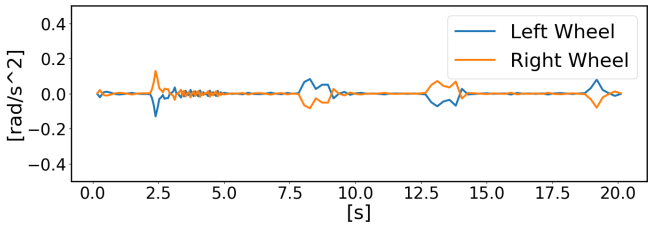
The objective of this test is to see if the the CCD is faster than the existing DPP. The test will be made by comparing the total time it takes for both trajectory planners to complete a template. If the test shows a 5% improvement for the time traversal of the CCD trajectory, the test will be deemed successful. Table IV shows the difference in traversal time of the CCD- and DPP trajectory. The trajectory is generated with $\sigma_{max} = 1.0$ and $v = 1.5$ $^{\text{m}}/_{\text{s}}$.

Time Optimality Test				
	CCD min:s	DPP min:s	Difference min:s	Increase %
Results	21:37	18:41	2:56	12

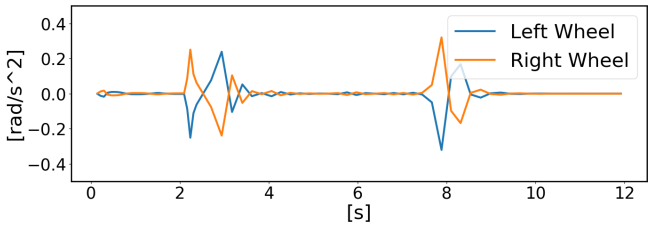
TABLE IV: Results from the time optimality test.

E. Wheel Acceleration Test

This test is conducted by executing a CCD- and a DPP trajectory along a single transition on a right corner transition. The main objective is to register the wheel accelerations experienced by each wheel along the trajectory. The trajectory is executed completely while recording wheel acceleration experienced by each wheel. The trajectory is generated with $\sigma_{max} = 1.0$ and it includes 3 meters in the entry- and exit direction to ensure a steady velocity. The measured wheel accelerations can be seen in Figure 12.



(a) Acceleration plot for CCD.



(b) Acceleration plot for DPP.

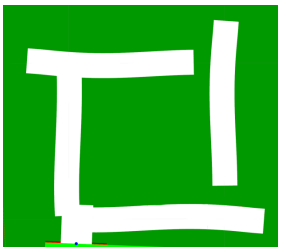
Fig. 12: Wheel accelerations of a right corner transition.

Looking at Figure 12, there are two additional spikes. This is due to additional wrap around in control one and three in the path, which happens when the turn is self intersecting, because the θ_{lim} constraint is violated. This is also a great contributor to the additional execution time seen along the horizontal axis. During the acceleration test, the maximum magnitude of the acceleration was recorded over a span of 10 runs, and the average for each planner is denoted in Table V.

Max Recorded Acceleration				
	CCD $^{rad}/_{s^2}$	DPP $^{rad}/_{s^2}$	Difference $^{rad}/_{s^2}$	Improvement %
Results	0.198	0.480	0.282	≈ 59

TABLE V: Max recorded accelerations and comparison between CCD and DPP planner.

The product of reducing the acceleration for the CCD planner is a lower transient response, where the improvement can be seen below in figure 13a and 13b.



(a) Showcase of transient response for DPP.

(b) Showcase of transient response for CCD.

IV. Discussion

This section will explore various facets of the solution, highlighting both successes and shortcomings observed during testing, and provide potential long term solutions.

A. Test Results

The continuity test showed in Figure 9 the path planner is successfully providing G^2 continuity during transitions while still being C^1 continuous. However, the test did not show that the planner only achieves G^2 continuity when it is transitioning between lines, but during transitions involving arcs, G^2 continuity is lost. This is due to the constraint that transitions must start and end with $\kappa = 0$.

The completeness test showed the CCD algorithm was able to find a path between points with 100% accuracy. Meaning the robot should be applicable for all templates, if there are not any obstacles in its path.

The time and path tests showed correlation between trajectory length and σ, indicating a trade-off in minimising σ without increasing trajectory length too much.

The acceleration test showed a 59% decrease in wheel acceleration during transitions, which contributes to a reduced transient response, when the robot reaches the desired end position. From the results of acceleration test 13b, the CCD algorithm significantly improves a differential drive robot's ability to follow a path. The algorithm could be ideal for applications where precise path following is crucial, such as line-marking robots.

B. Future Work

The current path planner, assumes constant velocity. Meaning potentially faster paths found by slowing down to turn faster are disregarded.

Football fields are surrounded by obstacles, thereby restricting the robots workspace. Therefore an addition of an obstacle avoidance algorithm could prove useful.

Using balkcom-Mason [6] curves can possibly reduce path length by removing the upper bound on κ to allow the robot to turn on the spot.

The planner has only been tested in simulation, thus the real world performance remains to be validated.

V. Conclusion

The original path planner assumed infinite acceleration but since that is physically infeasible it induces a transient response. This paper introduced a feasible trajectory for the robot by implementing linear change in curvature using clothoids. This proved a reduction in wheel acceleration by 59%, which minimises the transient response. This improvement enhances precision, by limiting the curvature rate, resulting in better lines, but at the cost of a longer execution time.

VI. Acknowledgements

We would like to express our gratitude to our supervisor, Kirsten Mølgaard, for her guidance on the project.

We would also like to extend our thanks to the entire team at Turf Tank for their collaboration. Their expertise and knowledge proved a great help throughout the project. Their experience greatly helped to shape the trajectory of this project.

References

[1] I. A. Hameed, "Path planning for line marking robots using 2d dubins' path," in *Advances in Intelligent Systems and Computing*, vol. 533. Cham: Springer International Publishing, 2017, pp. 900–910.

[2] A. Scheuer and T. Fraichard, "Continuous-curvature path planning for car-like vehicles," in *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97*, vol. 2, 1997, pp. 997–1003 vol.2.

[3] T. Fraichard and A. Scheuer, "From reeds and shepp's to continuous-curvature paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025–1035, 2004.

[4] M. Candeloro, A. M. Lekkas, A. J. Sørensen, and T. I. Fossen, "Continuous curvature path planning using voronoi diagrams and fermat's spirals," *ScienceDirect*, 2013.

[5] K. Yang and S. Sukkarieh, "3d smooth path planning for a uav in cluttered natural environments," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 794–800.

[6] D. J. Balkcom and M. T. Mason, "Time optimal trajectories for bounded velocity differential drive vehicles," *The International journal of robotics research*, vol. 21, no. 3, pp. 199–217, 2002.

[7] A. Giese, "A comprehensive, step-by-step tutorial on computing dubin's curves," New York NY, Tech. Rep., 1985.