# On the Initialization Problem for Timed-Elastic Bands ⋆

**Niklas Persson** * **Martin C. Ekström** * **Mikael Ekström** *
**Alessandro V. Papadopoulos** *

* *Mälardalen University, Västerås, Sweden*
*(e-mail:Firstname.Surname@mdu.se)*

**Abstract:** Path planning is an important part of navigation for mobile robots. Several approaches have been proposed in the literature based on a discretisation of the map, including A*, Theta*, and RRT*. While these approaches have been widely adopted also in real applications, they tend to generate non-smooth paths, which can be difficult to follow, based on the kinematic and dynamic constraints of the robot. Time-Elastic-Bands (TEB) have also been used in the literature, to deform an original path in real-time to produce a smoother path, and to handle potential local changes in the environment, such as the detection of an unknown obstacle. This work analyses the effects on the overall path for different choices of initial paths fed to TEB. In particular, the produced paths are compared in terms of total distance, curvature, and variation in the desired heading. The optimised version of the solution produced by Theta* shows the highest performance among the considered methods and metrics, and we show that it can be successfully followed by an autonomous bicycle.

*Keywords:* Planning, Optimisation, Time-Elastic-Bands, Intelligent Autonomous Vehicles, Navigation

## 1. INTRODUCTION

Planning a path between two points in a known, partially known or completely unknown environment is called path planning. A map is commonly divided into cells and graph-based search methods, such as Dijkstra's algorithm (Dijkstra, 1959), can be used to find the shortest path between two given cells. Other popular search methods include A*, Theta*, D* Lite and Rapidly-exploring Random Tree (RRT). A* is an extension of Dijkstra's, by using a heuristic to focus its search towards the goal (Hart et al., 1968). Moreover, D* Lite and Theta* are extensions of A* where D* Lite is intended to be used in an unknown environment (Koenig and Likhachev, 2002). Theta* belongs to any angle path planning algorithms and is not constrained by the edges of the cells which is the case of A*, D* Lite, and Dijkstra's (Daniel et al., 2010).

The initial path planned by A*, Theta*, D* Lite and its variants, can be tracked by many different types of robots, such as differential drive robots or omni-wheeled robots, which can rotate around their centre axis without forward or backward motion, thus making sharp turns. The algorithms are also popular in computer games where a low execution time is desirable (Yap et al., 2011). However, many other vehicles such as cars and bicycles are subject to non-holonomic constraints and are also unable to turn around their own axis, and thus, can not track the paths planned directly by A*, Theta*, or D* Lite. Instead, a kinematically feasible path is required. To address this issue, there are several ways of post-smoothing the path, such as utilising B-splines, Dubin's Curve, or polynomial interpolation (Ravankar et al., 2018). As an alternative, it is also possible to define the problem as an optimisation problem which has the advantage that it is possible to constrain state variables. Time-Elastic-Bands (TEB) (Rösmann et al., 2012) is formulated as a nonlinear optimisation problem with constraints on parameters such as the maximum velocity and acceleration, minimum turning radius, and minimum distance to obstacles in the optimisation problem. The TEB has been used for trajectory planning for numerous different robots, such as differential drive robots (Rösmann et al., 2013), carlike robots (Yongzhe et al., 2018), and mobile base platforms (Deray et al., 2019). However, the initial guess to the nonlinear optimisation problem is often assumed known beforehand.

In this paper, we investigate four different path-finding algorithms and compare the results by providing them as initial paths for the TEB. The basic A* algorithm, an any angle path finder represented by Theta*, a smooth path finder represented by Hybrid A*, and finally, RRT* as a sample-based path-finder are considered. To evaluate the performance of the optimised and non-optimised paths, the length of the path, the integrated absolute value of the heading derivative, and the curvature of the path are taken into account. Furthermore, to demonstrate the feasibility of the approach an autonomous bicycle is tracking the paths in a realistic multi-body simulation using a previously designed Model Predictive Controller (MPC) (Persson et al., 2021).

The paper is structured in the following way, first, background on the different path planners and related work is presented in Section 2. The optimisation of the paths using

⋆ The work is partly funded by Eskilstuna kommun och Eskilstuna Fabriksförening.

the TEB is described in Section 3. Next, the metrics used to evaluate the different path planners are presented in Section 4 followed by the results in Section 5. Concluding remarks and future work are outlined in Section 6.

## 2. BACKGROUND

In this section, the background and details of the path planners considered in this paper are presented. Next, work conducted using TEB and related work in terms of path planning for autonomous bicycles is given.

### *2.1 Path planning*

In this paper, we consider a map, $M^{m\times n}$, which is represented by a 2-dimensional binary occupancy grid, and each cell or node, $m_i$, is either free $m_i = 0$ or occupied $m_i = 1$. Four different path planners are used to find a path between the start and the goal cell. A* is a graph search algorithm where a heuristic is used to focus the search towards the goal. It was the first path planning algorithm that combined the cost, $f(m_i)$, from the start cell to the current cell, $g(m_i)$, with the heuristic, $h(m_i)$, between the current cell and the goal:

$$f(m_i) = g(m_i) + h(m_i). \tag{1}$$

Where the euclidean distance is chosen as the heuristic. Moreover, A* is a complete path planner, meaning that if there is a path of free cells between the start and goal node, it will be found. In the A* algorithm, each node has eight neighbouring nodes, i.e., its horizontal, vertical and diagonal neighbours. In Figure 1(a) it is clear that A* is constrained to movements in the directions of these neighbours.

Theta* works similarly to A* and in fact, they are sharing the same main loop. As in the case of A*, Theta* considers eight nodes as neighbours and searches the grid, i.e. the edges of the cells. In the case of A*, the parent of a node will be within the neighbours of its current node. However, the parent of the current node in Theta* is not constrained to the neighbours of the current node. Instead, Theta* checks if the current node and the parent node lie within Line Of Sight (LOS) of each other (Daniel et al., 2010), i.e. the two nodes do not need to be connected. Thus, the resulting path of Theta* is made up of several line segments which have an arbitrary angle and in general result in a path with fewer turns and shorter paths compared to a path planned by for example A* where the heading is constrained (Daniel et al., 2010). However, it is important to note that Theta* requires longer execution time compared to A*, due to the LOS check, which may be important in some applications (Uras and Koenig, 2015). From Figure 1(b), it is clear that the path planned by Theta* can have an arbitrary angle between two nodes.

In this paper, we also consider the Hybrid A* algorithm which is a path planner designed for creating kinematic feasible paths. The paths planned by Hybrid A* are constrained by the minimum turning radius of the vehicle, the length of the motions, and the number of motion primitives generated (Petereit et al., 2012). Instead of planning on a grid as in the case of A* and Theta*, Hybrid A* generates $N$ number of smooth motion primitives from the current node and the planner is constrained to these motion primitives. As a consequence, Hybrid A* is not constrained to only search on the grid or the centre of the cells, which is the case of A* and Theta*. Instead, the nodes of Hybrid A* can be placed anywhere within a free cell. This is illustrated in Figure 1(c), where the five motion primitives, in red, generated by Hybrid A* are not constrained by the edges or the centre of the cells. The resulting smooth path is visualised in green.

There are also sampling-based path planners, such as RRT and RRT*. A tree structure is obtained by repeatedly sampling a new randomly selected node in space and connecting this node with the closest node already in the tree. The advantage of the sampling-based algorithms compared to graph-based search methods is that they can effectively find feasible paths in larger state spaces and are not constrained to discrete cells in the map (LaValle, 2006). Another advantage of the RRT* is that even if the original planned path is found unfeasible due to some unknown obstacle, a new path can quickly be planned by using the already generated tree structure (Karaman and Frazzoli, 2011). Similarly to Hybrid A*, RRT* is not constrained to discrete cells either, and the nodes can be anywhere within the free space of the map. Furthermore, as in the case of A* and Theta*, RRT* is a complete path planner if a sufficient number of iterations are performed. In fact, RRT* will converge towards the optimal path as the number of nodes approaches infinity as it continues to optimise the path after the goal is reached. This is the main difference between RRT and RRT* (Karaman and Frazzoli, 2011). However, an infinite number of nodes is impracticable. Instead, it is up to the designer to determine the maximum number of iterations and the maximum number of nodes. Another important parameter for RRT* is the maximum distance between a new sample and the nodes in the tree as this choice will have a high impact on the convergence time. In Figure 1(d), the tree of RRT* after 500 iterations and a maximum connection distance of 1m is illustrated together with the shortest path in purple.

However, both Hybrid-A* and RRT* share the drawback of often producing jagged paths where an increasing number of heading changes is required, as compared to straight paths. This will lead to increased energy consumption and longer reference paths. Moreover, it might increase the complexity of the path tracker. In the case of an autonomous bicycle, the change of heading can be slow, especially if the bicycle is only equipped with a propulsion motor and a steering motor, as the steering regulation would also be in charge of balancing the bicycle (Zhao et al., 2017). Furthermore, the resulting path from Theta*, A*, and RRT* have sharp turns which would require the vehicle tracking the path to turn around its own axis. This is a manoeuvre that is not possible for vehicles which adhere to non-holonomic constraints, such as an autonomous bicycle, instead a smooth path is desirable.

### *2.2 Related work*

One approach for smoothing the path is to use TEB (Rösmann et al., 2012) which are based on the Elastic Bands proposed by Quinlan and Khatib (1993). An advantage of the TEB compared to Elastic Bands is that constraints on the kinodynamic properties of the vehicle can easily be included in the nonlinear optimisation

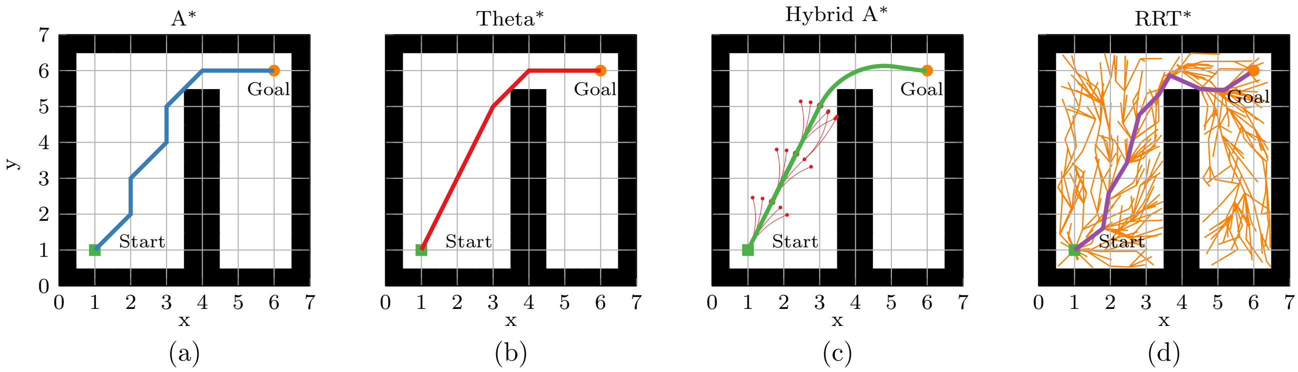


Fig. 1. The path planned by A*, Theta*, Hybrid A*, and RRT* respectively.

problem while keeping a safe distance from obstacles and minimising travel time. Thus, the TEB optimises a trajectory, instead of a path. In the work of Deray et al. (2019) the Timed-Elastic Smooth Curve (TESC), an extension of TEB that relies on Lie groups, is proposed and compared to the TEB. Both TESC and TEB are given the task to plan between the start position and randomly selected goal position, i.e. no initial path is planned. Both planners fail repeatedly in environments with static obstacles, something that could have been avoided if an initial path was planned with a complete path planner such as A* or Theta*. In the work of Yongzhe et al. (2018) a car-like robot used TEB to park the robot in a parking lot. The initial path was planned by A* and the strategy was evaluated in both simulations and experiments with promising results. A* was also used in the work of Ma et al. (2021) where the number of heading changes in the planned path was reduced by minimising the snap of the trajectory. Next, TEB was used to find a local optimal path.

Planning a feasible path for an autonomous bicycle requires the path to be smooth due to the non-holonomic constraints of the bicycle. This has been solved using different methods such as in the work of von Wissel and Nikoukhah (1995), where a path is planned based on the manoeuvres which can be performed by an autonomous bicycle, similar to Hybrid-A*. The manoeuvres are optimised to minimise the time of travel of the bicycle. The trajectory for a bicycle is also considered in the work of Yuan et al. (2014), where the trajectory is optimised by means of Particle Swarm Optimisation (PSO). A curve in the XY plane is parameterized by two third-order polynomials while satisfying initial and final constraints on the yaw angle and the x,y position. This leaves two free parameters, one for each polynomial which can be used by PSO to minimise the maximum lean angle of the bicycle. However, the initial path is assumed known, moreover, it is not clear how the proposed method would handle obstacles between the two points. In the work of Turnwald and Liu (2019) motion planning of a bicycle is investigated. They conclude that models that possess a positive trail (which most bicycles do), can produce paths which are best suitable for an autonomous bicycle.

## 3. PATH OPTIMISATION

Since we are interested in smooth paths which can be tracked by an autonomous bicycle, the planned paths are smoothed by TEB (Rösmann et al., 2012). The TEB can be visualised as putting an elastic band on top of the previously planned path, then tightening the band between the start and goal position to remove any slack and create a smooth path, while keeping a safe distance to obstacles, $\mathbf{o}_{min}$, and adhere to a minimum turning radius, $r_{min}$. Moreover, the algorithm minimises the time to travel from the start pose, $\mathbf{x}_s = [x_s, y_s, \theta_s]^\top$, to the goal pose, $\mathbf{x}_g = [x_g, y_g, \theta_g]^\top$ while considering constraints on the velocity, acceleration, angular velocity, and angular acceleration of a bicycle model

$$\dot{x} = v\cos(\theta) \tag{2}$$
$$\dot{y} = v\sin(\theta) \tag{3}$$
$$\dot{\theta} = \frac{v\tan(\delta)}{b}, \tag{4}$$

here $\delta$ is the steering angle, $b$ the wheelbase, $x$, $y$ are the position on the plane, and $\theta$ is the heading. TEB is a multi-objective optimisation problem where the states of the vehicle, $\mathcal{X} = \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ and time of travel between states, $T = \sum_{k=1}^{n-1} \Delta T_1, \Delta T_2, \dots, \Delta T_{n-1}$, are the variables and collected as $\mathcal{M} := \{\mathbf{x}_1, \Delta T_1, \mathbf{x}_2, \Delta T_2, \dots, \mathbf{x}_{n-1}, \Delta T_{n-1}, \mathbf{x}_n\}$. Following the approach in the work of Rösmann et al. (2017), the optimisation problem can be formulated as:

$$\begin{aligned}
\min_{\mathcal{M}} \quad & \sum_{k=1}^{n-1} \Delta T_k^2 \\
\text{s.t.} \quad & \mathbf{x}_1 = \mathbf{x}_s, \\
& \mathbf{x}_n = \mathbf{x}_g, \\
& \mathbf{h}_k(\mathbf{x}_{k+1}, \mathbf{x}_k) = \mathbf{0}, \\
& r_k - r_{min} \geq 0, \\
& \mathbf{o}_k(\mathbf{x}_k) - \mathbf{o}_{min} \geq \mathbf{0}, \\
& |v_k| \leq v_{max}, |a_k| \leq a_{max}, \\
& |\omega_k| \leq \omega_{max}, |\alpha_k| \leq \alpha_{max},
\end{aligned} \tag{5}$$

with

$$\mathbf{h}_k(\mathbf{x}_{k+1}, \mathbf{x}_k) = \begin{bmatrix} \cos(\theta_k) + \cos(\theta_{k+1}) \\ \sin(\theta_k) + \sin(\theta_{k+1}) \\ 0 \end{bmatrix} \times \mathbf{d}_k = \mathbf{0}, \tag{6}$$

where the direction vector, $\mathbf{d}_k$, is computed as

$$\mathbf{d}_k = \begin{bmatrix} x_{k+1} - x_k \\ y_{k+1} - y_k \\ 0 \end{bmatrix}. \tag{7}$$

Furthermore, $\mathbf{h}_k(\mathbf{x}_{k+1}, \mathbf{x}_k) = \mathbf{0}$ iff two consecutive poses $\mathbf{x}_k, \mathbf{x}_{k+1}$ are located on a common arc of constant curvature. Thus, this constraint affects the smoothness of the resulting path. $\mathbf{o}_k(\mathbf{x}_k)$ is the distance to a set of obstacles in the proximity of $\mathbf{x}_k$ and $r_k$ is the turning radius. Moreover, $v_{max}$, $a_{max}$, $\omega_{max}$, and $\alpha_{max}$ define the maximum velocity, acceleration, angular velocity and angular acceleration respectively. The nonlinear program in equation 5 is solved by means of Levenberg-Marquard solver by approximating the problem as a nonlinear least square problem where the constraints are used as penalty terms in the objective (Rösmann et al., 2017). Moreover, each penalty term is weighted to express the importance of each constraint. The constraints and the corresponding weights are presented in Table 1.

As the problem is a nonlinear program, there is no guarantee for converging to the optimal solution. The solution is heavily dependent on the initial guess, which in this case are the initial path and the initial velocities and accelerations. The initial path is the path planned by the A*, Theta*, Hybrid A*, and RRT* respectively and with, $v_1 = 0$, $a_1 = 0$, $\omega_1 = 0$, and $\alpha = 0$.

## 4. EVALUATION

The paths planned by Theta*, A*, Hybrid-A*, RRT* and their optimised versions are compared in 300 randomised maps. The size of each map is $100 \times 100$m with a resolution of 1 cell per meter. For each map, a maze is created with a wall thickness of 3m and a passage width of 8m, where the structure of the maze is randomised. Three different scenarios are used, in the first scenario, the passages in the maze are made up of free space. In the second scenario, the free space is cluttered with 50 randomly positioned obstacles and in the third scenario 100 randomly positioned obstacles are used. The start and goal positions are placed randomly on the map, with a minimum distance of 70m apart. Moreover, to realise a safety distance to the obstacles and the walls of the mazes, the obstacles and walls are inflated by a radius of 1m. To evaluate the performance of the different path planners the path length, the curvature of the paths and the integrated absolute value of the heading derivative ($IAT$) are considered. A shorter path length is desirable as it can save both time and energy for the vehicle tracking the path. To compute the path length, the euclidian distance is computed for each path and normalised with respect to Optimised Theta*, the average and standard deviation is computed for each scenario. The number of heading changes metric is also related to energy efficiency as a vehicle consumes more energy when it has to change its heading a lot. It is also related to the comfort of the ride, as constantly changing the steering direction will make an uncomfortable ride. However, the metric is better suited to be used in non-continuous paths where sharp heading changes are applied such as those produced by A* or Theta*. In the case of smooth paths, there may be small variations in the heading even on paths that appear straight. Moreover, small and large heading changes would count the same which makes the metric favouring large changes which are rarely found on smooth paths. Instead, the $IAT$ is considered and is defined as:

$$IAT = \sum_{k}^{n-1} \left| \frac{\theta_{k+1} - \theta_k}{T_s} \right|, \tag{8}$$

where $T_s$ is the sampling time $\theta_k$ is the heading in sample $k$. This metric combines the magnitude of the heading changes with the frequency of heading changes. Before $IAT$ is computed each path is interpolated linearly over 1000 samples. The resulting value is normalised with respect to Optimised Theta* for each map and the mean and standard deviation of 100 iterations for mazes with 0, 50, and 100 obstacles are computed. The curvature of the paths is only computed for the interpolated paths planned by the optimised versions of the path planners and the Hybrid A*, as the paths planned by Theta*, A*, and RRT* are made up of line segments and thus are not smooth and have curvature equal to zero everywhere. As in the case of the path distance and the $IAT$, the curvature is normalised with respect to the Optimised Theta* for each map and the mean and the standard deviation are computed for each scenario.

Table 1. Constraints and weights for TEB

| Constraint | Value | w | Constraint | Value | w |
|---|---|---|---|---|---|
| $v_{max}$ | 5m/s | 1 | $r_{min}$ | 3m | 10 |
| $a_{max}$ | 2m/s$^2$ | 1 | $\Delta T$ | 0.1s | - |
| $\omega_{max}$ | 0.3rad/s | 1 | $\sum_{k=1}^{n-1} \Delta T_k^2$ | - | 20 |
| $\alpha_{max}$ | 0.5rad/s$^2$ | 1 | $\mathbf{h}$ | - | 1000 |
| $\mathbf{o}_{min}$ | 1m | 3 | | | |

## 5. RESULTS

In this section, the results from the different path planning algorithms are presented. In the comparison, the maximum distance between a new node and the tree in RRT* is set to 10m, and $10^5$ iterations are performed with a maximum of $3 \times 10^4$ nodes in the tree. The Hybrid A* uses a minimum turning radius of 3m, a motion primitive length of 1.5m and 15 motion primitives are sampled at each node. Moreover, only forward motion is considered for all path planners. The resulting paths planned by A*, Theta*, Hybrid A*, and RRT* are optimised using the TEB as described in Section 3. Moreover, the path tracking results are presented where an autonomous bicycle is tracking a path planned by Optimised Theta*. The code for the comparisons and simulation, together with a video of the simulation, are available online [1]. The section is wrapped up with a discussion of the results.

### *5.1 Path planning & simulation results*

In Figure 2, the mean and standard deviation of the normalised path lengths are presented. The mean and standard deviation of the curvature is presented in Figure 3. Moreover, the mean and standard deviation for the $IAT$ value for all paths are given in Figure 4.

As the Optimised Theta* produces the most promising results when compared to the other path planners, it is used to plan a path for an autonomous bicycle in a realistic multi-body dynamics simulation using Simscape. A randomised maze of size $50 \times 50$m with a resolution of

[1] `https://github.com/NiklasPerssonMDU/On-the-Initial-of-Timed-Elastic-Bands.git`

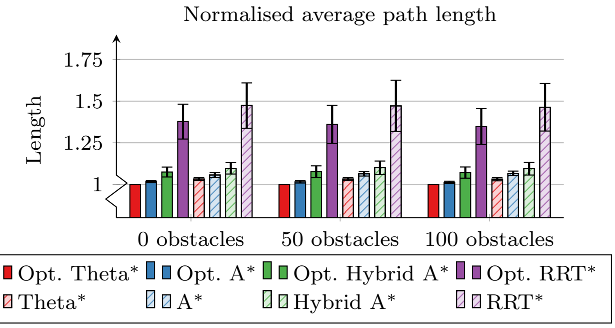


Fig. 2. Average and standard deviation of the normalised path, with respect to Optimised Theta*, lengths for 100 iterations with zero obstacles, 50 randomised obstacles, and 100 randomised obstacles respectivly in a randomised maze.

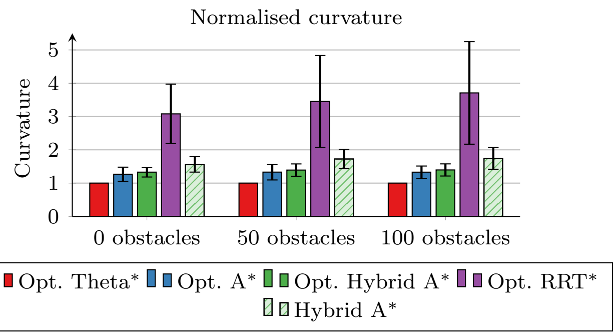


Fig. 3. The mean and the standard deviation for the curvature of the planned paths, normalised to Optimised Theta* in every iteration.

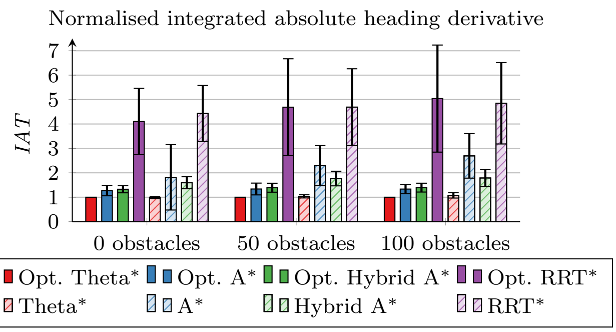


Fig. 4. Integrated absolute value of the heading derivative normalised concerning Optimised Theta*.

1 cell per meter with no further obstacles is considered. A minimum turning radius of 3m is used and a safety distance of 1m is considered. The 2-dimensional map is generated as a 3-dimensional environment in Simscape and a SolidWorks model of an ordinary-sized bicycle is imported and controlled through Simulink. Based on previous work (Persson et al., 2021), an MPC is used to track the difference, $\mathbf{r}_\Delta$, between the reference trajectory, $\boldsymbol{\Gamma} = [\theta_r, x_r, y_r, \varphi_r, \delta_r]^\top$, and the output denoted $\mathbf{y} = [\theta, x, y, \varphi, \delta]^\top$. Where $\theta_r, x_r, y_r$ are computed by Optimised Theta* and $\varphi_r = \delta_r = 0$. In the inner loop, the error between the lean angle, $\varphi$, and the reference lean angle, $\varphi*$, is fed to a PID controller that is used for balancing the bicycle by actuation of the steering velocity, $\dot{\delta}$. The inner loop is executing at 100Hz and the outer trajectory

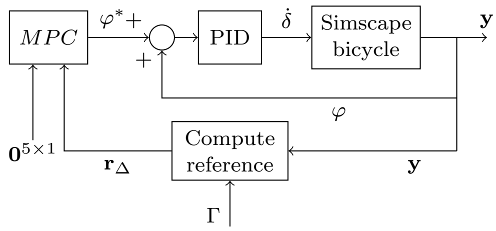


Fig. 5. Simulation setup of the control system.

tracking loop is running at 10Hz, while the bicycle model is simulated in continuous time. The control strategy is illustrated in Figure 5. To ensure a uniform sampling time of the optimised trajectory it is re-sampled with a sampling time of $T_s = 0.1$s before the simulation starts. The planned path and path-tracking performance of the autonomous bicycle are presented in Figure 6.

### 5.2 Discussion

From Figure 2, 3, 4 it is clear that the paths optimised using the TEB are shorter, have less curvature, and do not require as much heading regulation as compared to their non-optimised counterparts in general. However, RRT* is actually performing worse when optimised using TEB in terms of IAT for mazes which are cluttered with obstacles. Due to the cluttered environments RRT* tends to plan paths which have lots of nodes on a short distance which makes it difficult for the TEB to respect some constraints such as the minimum turning radius. An increased number of iterations and nodes allowed in the solution could improve the results of RRT* but at the cost of the execution time which is already high compared to the other path planners. Moreover, tuning of the maximum distance could have a positive effect on the results.

The paths length are decreased with $4.4\%, 3.1\%, 2.1\%$ and $7.3\%$ for A*, Theta*, Hybrid A*, and RRT* respectively when computing the average of the three different obstacle scenarios. Furthermore, the Optimised Theta* produces the shortest paths which were expected as Theta*, in

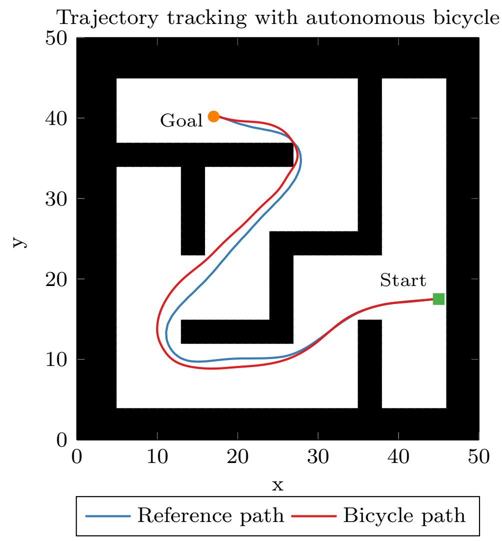


Fig. 6. Autonomous bicycle tracking the reference path planned by Optimised Theta*.

general, produces short paths, by smoothing the path using an elastic band the slack at the curves can be minimised. However, the TEB perform worse on paths which are already smooth, as in the case of Hybrid A*. This can be explained by the ratio of the weights and that the TEB favours smoothing the path and gets stuck in a locally optimal solution. The results highlights the importance of the initial conditions given to the NLP in equation 5. For the initial condition of the TEB, a discrete path with sharp turns, but with a low number of heading changes and short path length is performing better compared to an already smooth path with a longer path length and an increasing number of heading changes such as in the case of Hybrid A*. The results also suggests that TEB performs better on paths planned by grid search algorithms compared to sampled based algorithms and hybrid search algorithms. Moreover, Figure 6 shows that the path planned by Optimised Theta* successfully can be tracked by an autonomous bicycle in an environment with static obstacles.

## 6. CONCLUSION

In this paper, four different path planners are compared in 300 different maps. The resulting paths are optimised using Timed-Elastic-Bands which creates smooth paths that adhere to a number of different constraints, including maximum velocity, acceleration, and minimum turning radius. The results highlight the importance of the initial path fed to the TEB. Moreover, the results show that the resulting paths from the optimised Theta* have the shortest path length, the lowest curvature, and the lowest *IAT*. The optimisation does not only smooth the paths but in general improves the path planned by all algorithms in terms of all path lengths, heading changes, and curvature. This emphasizes the importance of optimisation when it comes to path planning for non-holonomic constrained vehicles. Furthermore, to demonstrate that it is possible to track the resulting path of the Optimised Theta*, an autonomous bicycle is used in a multi-body dynamic simulation. An MPC is utilised as a path tracker and a PID is used to keep the bicycle balanced by steering the bicycle into the fall. In the future, a TEB formulation with an more elaborated bicycle model could be investigated to constrain the maximum lean angle of the bicycle and include dynamic and unknown obstacles in the path planning.

## REFERENCES


Daniel, K., Nash, A., Koenig, S., and Felner, A. (2010). Theta*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research*, 533–579.

Deray, J., Magyar, B., Solà, J., and Andrade-Cetto, J. (2019). Timed-elastic smooth curve optimization for mobile-base motion planning. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, 3143–3149.

Dijkstra, E.W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 269–271.

Hart, P.E., Nilsson, N.J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *Trans. on Systems Science and Cybernetics*, 100–107.

Karaman, S. and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The Int. journal of robotics research*, 846–894.

Koenig, S. and Likhachev, M. (2002). D* lite. In *Conf. on Artificial Intelligence (AAAI)*, 476–483.

LaValle, S.M. (2006). *Planning Algorithms*. Cambridge university press.

Ma, Z., Qiu, H., Wang, H., Yang, L., Huang, L., and Qiu, R. (2021). A* algorithm path planning and minimum snap trajectory generation for mobile robot. In *Int. Conf. on Robotics, Control and Automation Engineering (RCAE)*, 284–288.

Persson, N., Ekström, M.C., Ekström, M., and Papadopoulos, A.V. (2021). Trajectory tracking and stabilisation of a riderless bicycle. In *Int. Intelligent Transportation Systems Conf. (ITSC)*, 1859–1866.

Petereit, J., Emter, T., Frey, C.W., Kopfstedt, T., and Beutel, A. (2012). Application of Hybrid A* to an autonomous mobile robot for path planning in unstructured outdoor environments. In *German Conf. on Robotics (ROBOTIK)*, 1–6.

Quinlan, S. and Khatib, O. (1993). Elastic bands: connecting path planning and control. In *Int. Conf. on Robotics and Automation (ICRA)*, 802–807.

Ravankar, A., Ravankar, A., Kobayashi, Y., Hoshino, Y., and Peng, C.C. (2018). Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges. *Sensors*, 3170.

Rösmann, C., Feiten, W., Wösch, T., Hoffmann, F., and Bertram, T. (2012). Trajectory modification considering dynamic constraints of autonomous robots. In *German Conf. on Robotics (ROBOTIK)*, 1–6.

Rösmann, C., Feiten, W., Wösch, T., Hoffmann, F., and Bertram, T. (2013). Efficient trajectory optimization using a sparse model. In *European Conf. on Mobile Robots (ECMR)*, 138–143.

Rösmann, C., Hoffmann, F., and Bertram, T. (2017). Kinodynamic trajectory optimization and control for car-like robots. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, 5681–5686.

Turnwald, A. and Liu, S. (2019). Motion planning and experimental validation for an autonomous bicycle. In *Conf. of the Industrial Electronics Society(IECON)*, 3287–3292.

Uras, T. and Koenig, S. (2015). An empirical comparison of any-angle path-planning algorithms. In *Int. Symposium on Combinatorial Search*, 206–210.

von Wissel, D. and Nikoukhah, R. (1995). Maneuver-based obstacle-avoiding trajectory optimization: Example of a riderless bicycle. *Mechanics of Structures and Machines*, 223–255.

Yap, P., Burch, N., Holte, R., and Schaeffer, J. (2011). Any-angle path planning for computer games. In *Conf. on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 201–207.

Yongzhe, Z., Ma, B., and Wai, C.K. (2018). A practical study of time-elastic-band planning method for driverless vehicle for auto-parking. In *Int. Conf. on Intelligent Autonomous Systems (ICoIAS)*, 196–200.

Yuan, J., Chen, H., Sun, F., and Huang, Y. (2014). Trajectory planning and tracking control for autonomous bicycle robot. *Nonlinear Dynamics*, 421–431.

Zhao, M., Stasinopoulos, S., and Yu, Y. (2017). Obstacle detection and avoidance for autonomous bicycles. In *Conf. on Automation Science and Engineering (CASE)*, 1310–1315.