2018 26th Mediterranean Conference on Control and Automation (MED)
Zadar, Croatia, June 19-22, 2018

Mobile Robot Trajectory Generation from Ordered Point-Set Using Time-Optimal Bezier Segments

Konstantinos Giannousakis[1] and Anthony Tzes[2]

***Abstract*— This paper addresses the problem of constructing a trajectory from a given set of points, considering a mobile robot. Following the initial pose of the robot, the points are connected in sequence by cubic Bezier segments. After selecting the tangential velocity at the connection points, these segments are time-optimized under the velocity and acceleration constraints of the robot, both tangential and angular, using non-linear constrained optimization. The optimization problem is simplified by exploitation of polynomials. Experimental studies demonstrate the effectiveness of the algorithm.**

I. INTRODUCTION

To effectively navigate a mobile robot from one point to another, it is required to provide the reference position of the robot through time. Therefore, trajectory creation is one of the most fundamental parts in all mobile robot applications. Generating a trajectory from a set of points is usually done by connecting the points with trajectory segments, forming a piecewise total trajectory. One of the most common approaches is the B-spline interpolation, used in [1] and [2]. The polynomial order of the splines depends on the trajectory characteristics required by the application. An alternative approach involves the Bezier curve, which is used in this paper and has also been studied by the authors in [3]–[8].

Trajectory generation is generally an optimization problem. The optimization parameter may be the length of the trajectory or, more commonly, its duration. Naturally, the optimization process should take into account the velocity and acceleration constraints of the robot; a problem which has been previously studied by the authors in [9]–[11].

In this paper, we propose a method of creating a trajectory from a given set of points, using the cubic Bezier curves. Every two consecutive points are interpolated with Bezier segments, ensuring the continuity of the tangential velocity at the connections. The tangential velocity connecting the segments is selected in a manner that prevents tangential or angular acceleration strains at either of the connecting trajectory segments. Using non-linear constraint optimization, each segment is time-optimized under the tangential and angular velocity and acceleration constraints of the robot. We achieve to simplify this optimization problem by computing the bounding values of the constraints using root-finding and evaluation of polynomials. Moreover, the initial velocity of the trajectory is a parameter, hence the trajectory can be calculated or recalculated online, given the current tangential velocity and orientation of the robot.

The paper is structured in the following manner. In Section II the model of the robot is presented and the required trajectory information are defined. In Section III the Bezier segments are formulated and the constrained optimization problem is introduced. Section IV introduces a trajectory controller, while Section V presents some experimental results. Finally, we provide some concluding remarks in Section VI.

II. PROBLEM STATEMENT

Let us consider a nonholonomic mobile robot with differential drive, as shown in Fig. 1, with constrained tangential and angular velocities and accelerations. The kinematic equations are given as

$$\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{\theta}(t)\end{bmatrix} = \begin{bmatrix}\cos\theta(t) & 0\\ \sin\theta(t) & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v(t)\\ \omega(t)\end{bmatrix},$$

where $v(t)$ and $\omega(t)$ are the tangential and angular velocity of the robot, respectively. The robot kinematic constraints are defined as follows

$$\begin{aligned} v_{\min} &\leq v(t) \leq v_{\max},\\ \omega_{\min} &\leq \omega(t) \leq \omega_{\max},\\ a_{\min} &\leq a(t) \leq a_{\max},\\ \alpha_{\min} &\leq \alpha(t) \leq \alpha_{\max}, \end{aligned} \tag{1}$$

where $v_{\min},\ v_{\max} \in \mathbb{R}$ and $\omega_{\min},\ \omega_{\max} \in \mathbb{R}$, are the tangential and angular velocity bounds, respectively and $a_{\min},\ a_{\max} \in \mathbb{R}$ and $\alpha_{\min},\ \alpha_{\max} \in \mathbb{R}$, are the tangential and angular acceleration bounds, respectively. Moreover, we assume that the robot has a sampling time interval $T_s \in \mathbb{R}^+$. At each iteration $k \in \mathbb{Z}^+$ the trajectory controller is applied and we have access to the robot's measured position $\mathbf{x}_m(kT_s) = \begin{bmatrix} x_m(kT_s) & y_m(kT_s)\end{bmatrix}^T$ and orientation $\theta_m(kT_s) \in \mathbb{R}$. For simplicity, a value at time instance kT_s will be denoted with just the subscript k, for example $\mathbf{x}_m(kT_s) = \mathbf{x}_{m,k}$.

Let $\mathbf{X} = \begin{bmatrix}\mathbf{X}_0 & \mathbf{X}_1 & \cdots & \mathbf{X}_n\end{bmatrix}$ be a given ordered set of points that must be traversed by the mobile robot, with $n \in \mathbb{Z}^+$ being the number of points and $\mathbf{X}_0$ the initial position of the robot. The task is to connect these points through a trajectory function $\mathbf{T}(t) = \mathbf{x}(t) = \begin{bmatrix}x(t) & y(t)\end{bmatrix}^T$, where $t \in [t_0, t_n]$ is the time instance variable and $t_i \in$

*This work was not supported by any organization

[1]The author is with the Electrical & Computer Engineering Department, University of Patras, Rio 26500, Greece. Corresponding author's email: giannousakis@ece.upatras.gr

[2]The author is with the Electrical & Computer Engineering Program, New York University Abu Dhabi, Abu Dhabi, P.O. Box 129188, United Arab Emirates.

978-1-5386-7891-6/18/$31.00 ©2018 IEEE

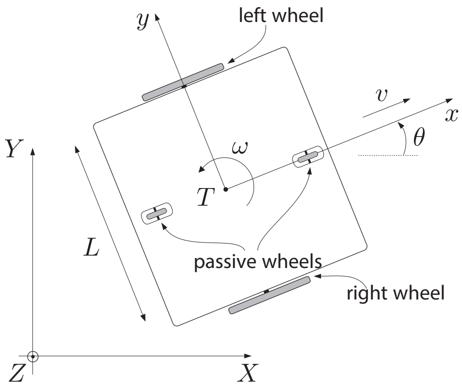


Fig. 1: Robot architecture [12].

$\mathbb{R}^+$, $i \in [0, n]$ are the unknown time instances such that $\mathbf{T}(t_i) = \mathbf{X}_i$.

To achieve this, we split the trajectory into segments. These segments will be denoted as $\mathbf{T}_j(t)$, with $j \in [0, m]$, $m = n-1$. As a matter of fact, the total trajectory takes the form

$$\mathbf{T}(t) = \begin{cases} \mathbf{T}_0(t), & t_0 \leq t < t_1 \\ \quad\cdots \\ \mathbf{T}_j(t), & t_j \leq t < t_{j+1} \\ \quad\cdots \\ \mathbf{T}_m(t), & t_m \leq t \leq t_{m+1} \end{cases}$$

where its continuity, along with its first derivative, is ensured by the following properties

$$\mathbf{T}_j(t_j) = \mathbf{X}_j, \quad \mathbf{T}_j(t_{j+1}) = \mathbf{X}_{j+1},$$
$$\mathbf{T}_j^{(1)}(t_{j+1}) = \mathbf{T}_{j+1}^{(1)}(t_{j+1}).$$

The reference tangential velocity vector for the robot, is given by the first derivative of the trajectory, being $\mathbf{T}^{(1)}(t) = \mathbf{v}(t) = \begin{bmatrix}\dot{x}(t) & \dot{y}(t)\end{bmatrix}^T$. Breaking it apart, we get the reference tangential velocity and orientation of the robot through the trajectory, which are respectively

$$v(t) = \sqrt{\dot{x}^2(t) + \dot{y}^2(t)}, \tag{2}$$

$$\theta(t) = \arctan_2(\dot{y}(t), \dot{x}(t)). \tag{3}$$

We will denote the tangential velocity vector and orientation at each point $\mathbf{X}_i$ as $\mathbf{V}_i$ and Θ_i, respectively. If the trajectory is computed while the robot is moving, the robot may have an initial velocity $\mathbf{V}_0$, and depending on the application, the robot may require a specific orientation Θ_n at the end of the trajectory.

Taking the derivatives of equations (2) and (3) we define the rest of the required information for the trajectory. Specifically, the reference tangential acceleration is $a(t) = \dot{v}(t)$, while the angular velocity and acceleration are $\omega(t) = \dot{\theta}(t)$ and $\alpha(t) = \dot{\omega}(t)$. Furthermore, we need the tangential and angular jerk which are, $\dot{a}(t)$ and $\dot{\alpha}(t)$, respectively.

III. Trajectory Generation

In our study, we use of the cubic Bezier curve [13], which is expressed as

$$\mathrm{B}(\lambda) = (1-\lambda)^3 P_0 + 3(1-\lambda)^2 \lambda P_1 + 3(1-\lambda)\lambda^2 P_2 + \lambda^3 P_3$$

where $\lambda \in [0, 1]$ is the curve parameter and $P_l \in \mathbb{R}$, $l \in \{0, 1, 2, 3\}$ are the four control points. Expanding it into its polynomial form we obtain

$$\mathrm{B}(\lambda) = \begin{bmatrix} P_3 - 3P_2 + 3P_1 - P_0 \\ 3(P_2 - 2P_1 + P_0) \\ 3(P_1 - P_0) \\ P_0 \end{bmatrix}^T \begin{bmatrix} \lambda^3 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}$$

and its first four derivatives are

$$\mathrm{B}^{(1)}(\lambda) = \frac{\partial \mathrm{B}(\lambda)}{\partial \lambda} = \begin{bmatrix} 3(P_3 - 3P_2 + 3P_1 - P_0) \\ 6(P_2 - 2P_1 + P_0) \\ 3(P_1 - P_0) \end{bmatrix}^T \begin{bmatrix} \lambda^2 \\ \lambda \\ 1 \end{bmatrix},$$

$$\mathrm{B}^{(2)}(\lambda) = \frac{\partial^2 \mathrm{B}(\lambda)}{\partial \lambda^2} = \begin{bmatrix} 6(P_3 - 3P_2 + 3P_1 - P_0) \\ 6(P_2 - 2P_1 + P_0) \end{bmatrix}^T \begin{bmatrix} \lambda \\ 1 \end{bmatrix},$$

$$\mathrm{B}^{(3)}(\lambda) = \frac{\partial^3 \mathrm{B}(\lambda)}{\partial \lambda^3} = 6(P_3 - 3P_2 + 3P_1 - P_0),$$

$$\mathrm{B}^{(4)}(\lambda) = \frac{\partial^4 \mathrm{B}(\lambda)}{\partial \lambda^4} = 0.$$

Thus, for the four control points we have

$$P_0 = \mathrm{B}(0), \qquad P_1 = P_0 + \frac{\mathrm{B}^{(1)}(0)}{3},$$
$$P_3 = \mathrm{B}(1), \qquad P_2 = P_3 - \frac{\mathrm{B}^{(1)}(1)}{3}.$$

In our two-dimentional case, the trajectory function for each segment has the form

$$\mathbf{T}_j(t) = \mathbf{x}_j(t) = \begin{bmatrix} x_j(t) \\ y_j(t) \end{bmatrix} = \begin{bmatrix} \mathrm{B}_{j,x}\left(\frac{t-t_j}{dt_j}\right) \\ \mathrm{B}_{j,y}\left(\frac{t-t_j}{dt_j}\right) \end{bmatrix}$$

where $t \in [t_j, t_{j+1}]$ is the segment's time instance variable, dt_j is the segment's duration to be minimized and t_j, $t_{j+1} = t_j + dt_j$ are the starting and ending time instances of the segment respectively. In addition, the derivatives take the form

$$\mathbf{T}_j^{(n)}(t) = \begin{bmatrix} x_j^{(n)}(t) \\ y_j^{(n)}(t) \end{bmatrix} = \frac{1}{dt_j^n} \begin{bmatrix} \mathrm{B}_{j,x}^{(n)}\left(\frac{t-t_j}{dt_j}\right) \\ \mathrm{B}_{j,y}^{(n)}\left(\frac{t-t_j}{dt_j}\right) \end{bmatrix}, \tag{4}$$

with the first one being the vector of the reference tangential velocity for the robot

$$\mathbf{T}_j^{(1)}(t) = \mathbf{v}_j(t) = \begin{bmatrix} \dot{x}_j(t) \\ \dot{y}_j(t) \end{bmatrix} = \frac{1}{dt_j} \begin{bmatrix} \mathrm{B}_{j,x}^{(1)}\left(\frac{t-t_j}{dt_j}\right) \\ \mathrm{B}_{j,y}^{(1)}\left(\frac{t-t_j}{dt_j}\right) \end{bmatrix}.$$

Now, we can express the four control points of the segment, which are two-dimentional, as follows

$$\mathbf{P}_{j,0} = \mathbf{x}_j(t_j), \quad \mathbf{P}_{j,1} = \mathbf{x}_j(t_j) + \frac{dt_j}{3}\mathbf{v}_j(t_j),$$
$$\mathbf{P}_{j,3} = \mathbf{x}_j(t_{j+1}), \quad \mathbf{P}_{j,2} = \mathbf{x}_j(t_{j+1}) - \frac{dt_j}{3}\mathbf{v}_j(t_{j+1}). \tag{5}$$

Hence, all we need in order to define the segment, are its starting and ending points along with their tangential velocity vector and the duration of the segment.

We emphasise here that if $\mathbf{v}_j(t_j) = 0$ or $\mathbf{v}_j(t_{j+1}) = 0$, then we would have $\mathbf{P}_{j,1} = \mathbf{P}_{j,0}$ or $\mathbf{P}_{j,2} = \mathbf{P}_{j,3}$. This is a singular case, where we lose control over the initial or final direction of the robot in the trajectory segment. Although having zero velocity in the trajectory is not allowed, we may require the initial or final velocity of the total trajectory to be zero. We will resolve this singularity using a practical approach.

A. Selecting the segment's velocity

If the initial velocity is zero we need to determine a suitable value for $\mathbf{V}_0$, in order to avoid the aforementioned singularity. Making use of the maximum tangential acceleration of the robot, $a_{\max}$, its initial orientation, Θ_0 and the sampling duration T_s we choose

$$\mathbf{V}_0 = a_{\max} T_s \angle \Theta_0,$$

which is the maximum velocity that the robot can achieve in a sampling interval.

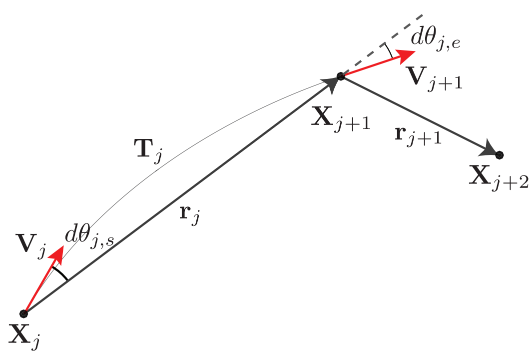


Fig. 2: Velocity Selection

For each trajectory segment, $\mathbf{T}_j$, we just need to select its final velocity, $\mathbf{V}_{j+1}$, since the initial, $\mathbf{V}_j$, is already determined from the previous segment or is $\mathbf{V}_0$. Starting with its angle Θ_{j+1}, the direction should be towards the next point, $\mathbf{X}_{j+2}$, but in order to avoid steep orientation changes for the robot, it should also not deviate a lot from the normal path orientation. With that in mind, we define the normal path vector $\mathbf{r}_j = \mathbf{X}_{j+1} - \mathbf{X}_j$ and chose

$$\Theta_{j+1} = \angle(\mathbf{r}_j + \mathbf{r}_{j+1}).$$

The amplitude of the velocity needs to be large enough to account for an optimization process, but small enough respect the constraints and to avoid unnecessary angular velocity or acceleration strains. We know that, for the linear part, the robot needs at most $t = v_{\max}/a_{\max}$ time to achieve maximum velocity and that the mean velocity to traverse a distance d is $v = d/t$. Hence we chose the maximum allowed velocity as

$$v_{a,j} = \min\left(v_{\max}, \min\left(\|\mathbf{r}_j\|, \|\mathbf{r}_{j+1}\|\right)\frac{a_{\max}}{v_{\max}}\right).$$

Moreover, we define the velocity limiting factor due to orientation deviations

$$f_j = (1 - \xi \sin^2 d\theta_{j,s}) \cos^2 d\theta_{j,e},$$

with $d\theta_{j,s} = \Theta_j - \angle\mathbf{r}_j$ and $d\theta_{j,e} = \angle\mathbf{r}_j - \Theta_{j+1}$ being the deviations of the velocity angles from the normal path vector, while $\xi \in (0, 1)$ is a heuristic constant, closely connected to the maximum allowed angular velocity and acceleration of the robot. Finally, the velocity is chosen as

$$\mathbf{V}_{j+1} = f_j v_{a,j} \angle\Theta_{j+1}.$$

The larger the value of the constant ξ, the smaller the value of $\mathbf{V}_{j+1}$ with respect to $d\theta_{j,s}$ becomes, resulting in smaller angular velocity and acceleration in the neighbourhood of $\mathbf{X}_{j+1}$.

Considering the final velocity of the robot, $\mathbf{V}_n$, we continue in the same manner. If Θ_n is not specified, following the previous logic, we chose

$$\Theta_{m+1} = \angle\mathbf{r}_m + d\theta_{m,s},$$

which becomes

$$\Theta_n = 2\angle\mathbf{r}_m - \Theta_m.$$

Furthermore, as the robot needs to stop at the end of the trajectory, as with $\mathbf{V}_0$, we chose

$$\mathbf{V}_n = a_{\max} T_s \angle\Theta_n.$$

B. Constrained non-linear optimization

Before establishing the optimization problem for each segment, we need, first, to extract some additional information. Assuming forward movement, the robot's reference tangential and angular velocities in the trajectory are respectively

$$v(t) = \sqrt{\dot{x}^2(t) + \dot{y}^2(t)} = \sqrt{v_r(t)},$$
$$\omega(t) = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{v^2(t)} = \frac{\omega_r(t)}{v^2(t)},$$

with $v_r(t)$ and $\omega_r(t)$ given as

$$v_r(t) = \dot{x}^2(t) + \dot{y}^2(t),$$
$$\omega_r(t) = \dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t).$$

We obtain the tangential and angular accelerations

$$a(t) = \dot{v}(t) = \frac{a_r(t)}{v(t)},$$
$$\alpha(t) = \dot{\omega}(t) = \frac{\alpha_r(t)}{v^4(t)},$$

where $a_r(t)$ and $\alpha_r(t)$ are respectively

$$a_r(t) = \dot{x}(t)\ddot{x}(t) + \dot{y}(t)\ddot{y}(t),$$
$$\alpha_r(t) = \big(\dot{x}(t)\dddot{y}(t) - \dot{y}(t)\dddot{x}(t)\big)v_r(t) - 2a_r(t)\omega_r(t).$$

Finally, knowing that $\ddddot{x}(t) = \ddddot{y}(t) = 0$, the tangential and angular jerks are

$$\dot{a}(t) = \frac{\dot{a}_r(t)}{v^3(t)},$$

$$\dot{\alpha}(t) = \frac{\dot{\alpha}_r(t)}{v^6(t)},$$

with $\dot{a}_r(t)$ and $\dot{\alpha}_r(t)$ given as

$$\begin{aligned}\dot{a}_r(t) = &\big(\ddot{x}^2(t) + \ddot{y}^2(t) \\ &+\dot{x}(t)\dddot{x}(t) + \dot{y}(t)\dddot{y}(t)\big)v_r(t) - a_r(t)\end{aligned}$$

$$\begin{aligned}\dot{\alpha}_r(t) = \Big[&\big(\dot{x}(t)\dddot{y}(t) - \dot{y}(t)\dddot{x}(t)\big)v_r(t) \\ &-2\big(\ddot{x}^2(t) + \ddot{y}^2(t), \\ &+\dot{x}(t)\dddot{x}(t) + \dot{y}(t)\dddot{y}(t)\big)\omega_r\Big]v_r(t) \\ &- 4a_r(t)\alpha_r(t).\end{aligned}$$

The above equations hold for the whole trajectory, as they also do for each segment. Since the optimization is per segment, we know that $x_j(t)$, $y_j(t)$ and their derivatives are polynomials. With the exception of $v(t)$, all of the extracted trajectory information are formed as rational polynomials of t, with dt_j as an external parameter. To highlight the segment's dependency on dt_j for the required equations and polynomial nominators, we will use the more complex notations $v_j(t; dt_j)$, $\omega_j(t; dt_j)$, $a_j(t; dt_j)$, $\alpha_j(t; dt_j)$ and $a_{r,j}(t; dt_j)$, $\alpha_{r,j}(t; dt_j)$, $\dot{a}_{r,j}(t; dt_j)$, $\dot{\alpha}_{r,j}(t; dt_j)$. Now, it is possible to state the velocity and acceleration bounds of the segment with respect to the minimazation parameter dt_j as follows

$$\begin{aligned}v_{j,\min}(dt_j) &= \min_{t \in \text{roots}\, a_{r,j}(t; dt_j)} v(t), \\ v_{j,\max}(dt_j) &= \max_{t \in \text{roots}\, a_{r,j}(t; dt_j)} v(t),\end{aligned}$$

$$\begin{aligned}\omega_{j,\min}(dt_j) &= \min_{t \in \{t_j, t_{j+1}, \text{roots}\, \alpha_{r,j}(t; dt_j)\}} \omega_(t), \\ \omega_{j,\max}(dt_j) &= \max_{t \in \{t_j, t_{j+1}, \text{roots}\, \alpha_{r,j}(t; dt_j)\}} \omega_(t),\end{aligned}$$

$$\begin{aligned}a_{j,\min}(dt_j) &= \min_{t \in \{t_j, t_{j+1}, \text{roots}\, \dot{a}_{r,j}(t; dt_j)\}} a(t), \\ a_{j,\max}(dt_j) &= \max_{t \in \{t_j, t_{j+1}, \text{roots}\, \dot{a}_{r,j}(t; dt_j)\}} a(t),\end{aligned}$$

$$\begin{aligned}\alpha_{j,\min}(dt_j) &= \min_{t \in \{t_j, t_{j+1}, \text{roots}\, \dot{\alpha}_{r,j}(t; dt_j)\}} \alpha(t), \\ \alpha_{j,\max}(dt_j) &= \max_{t \in \{t_j, t_{j+1}, \text{roots}\, \dot{\alpha}_{r,j}(t; dt_j)\}} \alpha(t).\end{aligned}$$

Finally, we can formally state the optimization problem as

$$\min_{dt_j \in \mathbb{R}^+} dt_j \quad \text{subject to} \quad \begin{cases} v_{j,\min}(dt_j) \geq v_{\min} \\ v_{j,\max}(dt_j) \leq v_{\max} \\ \\ \omega_{j,\min}(dt_j) \geq \omega_{\min} \\ \omega_{j,\max}(dt_j) \leq \omega_{\max} \\ \\ a_{j,\min}(dt_j) \geq a_{\min} \\ a_{j,\max}(dt_j) \leq a_{\max} \\ \\ \alpha_{j,\min}(dt_j) \geq \alpha_{\min} \\ \alpha_{j,\max}(dt_j) \leq \alpha_{\max} \end{cases} \tag{6}$$

and all we need is an initial condition. We observe from (4) that

$$\begin{aligned}v_j(t; dt_j) &\simeq \frac{v_j(t; 1)}{dt_j}, \\ \omega_j(t; dt_j) &\simeq \frac{\omega_j(t; 1)}{dt_j^2}, \\ a_j(t; dt_j) &\simeq \frac{a_j(t; 1)}{dt_j^2}, \\ \alpha_j(t; dt_j) &\simeq \frac{\alpha_j(t; 1)}{dt_j^4},\end{aligned}$$

which is only an approximation as the Bezier control points dependent on dt_j as well. Nevertheless, based on that we chose

$$dt_{j,0} = \max \begin{bmatrix} \max\left(0, \frac{v_{j,\min}(1)}{v_{\min}}, \frac{v_{j,\max}(1)}{v_{\max}}\right) \\ \sqrt{\max\left(0, \frac{\omega_{j,\min}(1)}{\omega_{\min}}, \frac{\omega_{j,\max}(1)}{\omega_{\max}}\right)} \\ \sqrt{\max\left(0, \frac{a_{j,\min}(1)}{a_{\min}}, \frac{a_{j,\max}(1)}{a_{\max}}\right)} \\ \sqrt[4]{\max\left(0, \frac{\alpha_{j,\min}(1)}{\alpha_{\min}}, \frac{\alpha_{j,\max}(1)}{\alpha_{\max}}\right)} \end{bmatrix} \tag{7}$$

as our initial condition.

IV. Trajectory Controller

The trajectory controller that we use is the one introduced in [12]. Without getting into the details, we will just state the equations for the control inputs of the robot. Firstly, we have the feedforward tangential and angular velocities $v_{f,k} = v_k$ and $\omega_{f,k} = \omega_k$, respectively. Additionally, the error from the reference path is

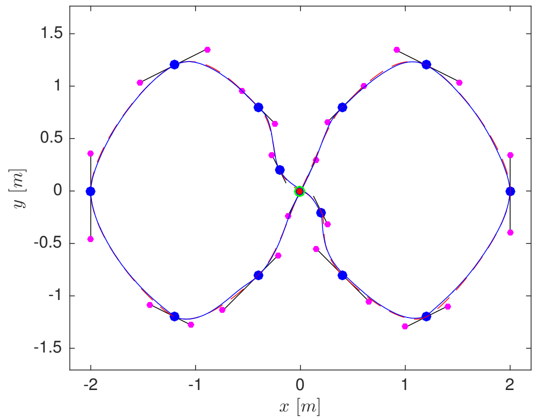
$$\begin{bmatrix} e_{x,k} \\ e_{y,k} \\ e_{\theta,k} \end{bmatrix} = \begin{bmatrix} \cos\theta_{m,k} & \sin\theta_{m,k} & 0 \\ -\sin\theta_{m,k} & \cos\theta_{m,k} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k - x_{m,k} \\ y_k - y_{m,k} \\ \theta_k - \theta_{m,k} \end{bmatrix}.$$

The system characteristic frequency can be found by $\omega_{n,k} = \sqrt{\omega_{f,k}^2 + g\, v_{f,k}^2}$ and the controller gains are given as $K_{1,k} = K_{3,k} = 2\zeta\omega_{n,k}$, $K_{2,k} = g\,|v_{f,k}|$, where g is a controller design parameter and $\zeta \in (0, 1)$ is the desired dumping coefficient. The control law is

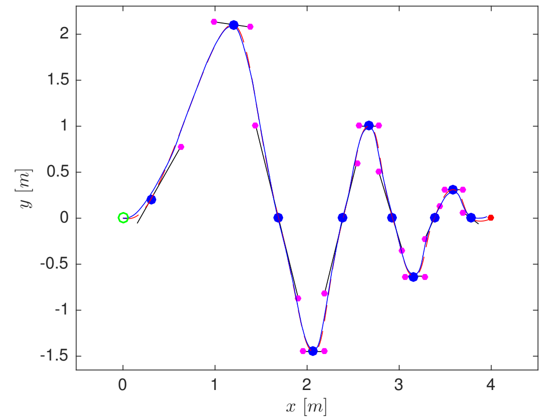
$$\begin{bmatrix} v_{c,k} \\ \omega_{c,k} \end{bmatrix} = \begin{bmatrix} -K_{1,k} & 0 & 0 \\ 0 & -K_{2,k}\,\text{sign}(v_{f,k}) & -K_{3,k} \end{bmatrix} \begin{bmatrix} e_{x,k} \\ e_{y,k} \\ e_{\theta,k} \end{bmatrix}$$

and finally, the robot inputs are

$$\begin{aligned}v_{cf,k} &= v_{f,k}\cos e_{3,k} - v_{c,k}, \\ \omega_{cf,k} &= \omega_{f,k} - \omega_{c,k}.\end{aligned}$$

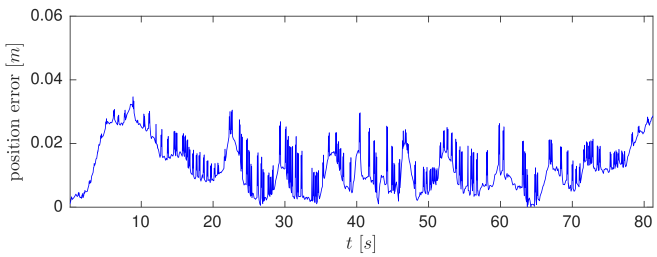


(a) Point-set Ia

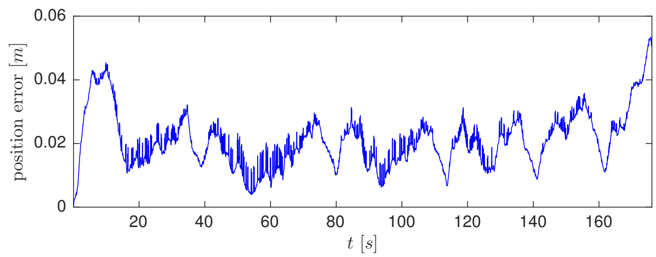


(b) Point-set Ib

Fig. 3: Reference (– –) and robot (–) trajectory for point-set (●), along with the rest Bezier segments' control points (●).



(a) Point-set Ia



(b) Point-set Ib

Fig. 4: Robot position deviation.

TABLE I: Point-Sets used in our experiments.

(a)

X	x $[m]$	y $[m]$
0	0.00	0.00
1	0.20	-0.20
2	0.40	-0.80
3	1.20	-1.20
4	2.00	-0.00
5	1.20	1.20
6	0.40	0.80
7	0.00	0.00
8	-0.40	-0.80
9	-1.20	-1.20
10	-2.00	0.00
11	-1.20	1.20
12	-0.40	0.80
13	-0.20	0.20
14	0.00	-0.00

(b)

X	x $[m]$	y $[m]$
0	0.000	0.000
2	0.300	0.200
3	1.195	2.105
4	1.690	0.000
5	2.070	-1.450
6	2.390	-0.000
7	2.670	1.000
8	2.930	0.000
9	3.160	-0.630
10	3.380	0.000
11	3.585	0.310
12	3.779	0.000
13	4.000	0.000

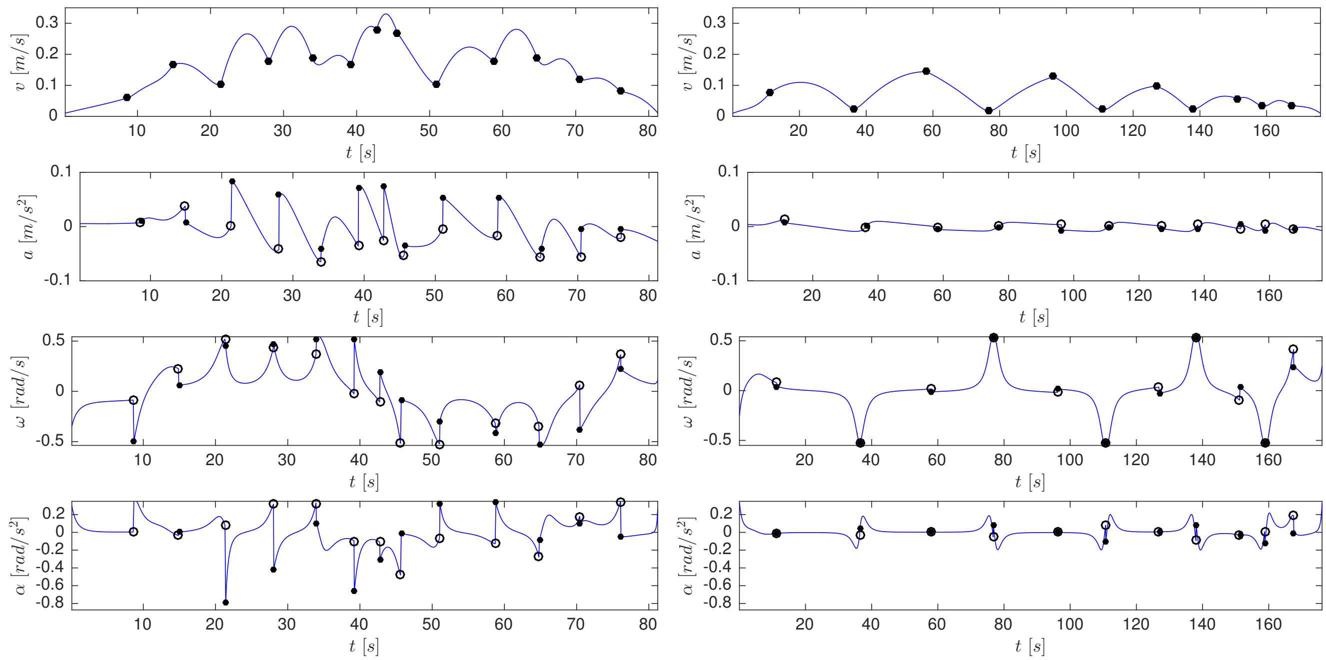
V. Experimental Results

In our experiments we used the AmigoBot from ActiveMedia, shown in Fig. 5. The position of the robot is based on encoder measurements from its wheels and is automatically given by the robot. Additionally, the sampling interval is $Ts \approx 100\ ms$ and the constraints used are $v(t) \in [0, 0.35]\ m/s$, $a(t) \in [-0.1, 0.1]\ m/s^2$, $\omega(t) \in [-30, 30]\ deg/s$ and $\alpha(t) \in [-50, 20]\ deg/s^2$. Moreover, after trial and error, the trajectory heuristic parameter was selected as $\xi = 0.6$, since it produced better results. Additionally, the trajectory controller parameters were selected as $g = 30$ and $\zeta = 0.6$.

Fig. 5: ActiveMedia's AmigoBot used in the experiments.

We present two experiments using the point-sets shown in Table I. Without loss of generality we assume that the initial orientation of the robot is $\Theta_0 = 0\ rad$ and its initial tangential velocity is $0\ m/s$. The resulting reference trajectory is depicted along with the measured one in Fig. 3. Additionally, Fig. 4 shows the position error, which is given by $e = \sqrt{(x_k - x_{m,k})^2 + (y_k - y_{m,k})^2}$. And finally, the reference tangential and angular velocities and accelerations are depicted in Fig. 6.

The tangential velocity is continuous by design and as it seems, for these trajectories, it never reaches its bounding values. On the other hand, the tangential acceleration does



(a) Point-set Ia

(b) Point-set Ib

Fig. 6: Reference tangential and angular velocities and accelerations.

not retain continuity on the segment changes, but again the constraints are satisfied. As expected from the optimization process, in each segment, at least one constraint is at its limit value. In both our cases, this happens with the angular velocity or acceleration. As with the tangential acceleration, neither is the angular velocity nor the acceleration continuous on the segment changes. Moreover, at these time instances the reference angular acceleration has theoretically infinite value. However, the robot cannot exceed the constraints (1), so in reality, one of its bounding values will be used instead. On the other hand, observing the position error, the robot smoothly adheres to the reference trajectory. The maximum deviation is $3.46\ cm$ and $5.34\ cm$, for point-set Ia and Ib respectively, while the mean deviation is barely at $1.32\ cm$ for the first and $2.12\ cm$ for the other.

VI. Conclusions

This paper exploits the Bezier curve to construct a trajectory from a given point-set. Constrained non-linear optimization is used for the trajectory segments in-between each two points, where the constraints are evaluated through polynomials. Finally, experimental studies demonstrate that a trajectory controller successfully tracks the constructed trajectory.

References

[1] R. Walambe, N. Agarwal, S. Kale, and V. Joshi, "Optimal trajectory generation for car-type mobile robot using spline interpolation," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 601–606, 2016.

[2] A. Piazzi, C. G. L. Bianco, and M. Romano, "3-splines for the smooth path generation of wheeled mobile robots," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1089–1095, 2007.

[3] J. W. Choi, R. Curry, and G. Elkaim, "Path planning based on bezier curve for autonomous ground vehicles," in *Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*, pp. 158–166, 2008.

[4] I. Arvanitakis, A. Tzes, and M. Thanou, "Geodesic motion planning on 3d-terrains satisfying the robot's kinodynamic constraints," in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, pp. 4144–4149, 2013.

[5] K. R. Simba, N. Uchiyama, and S. Sano, "Real-time trajectory generation for mobile robots in a corridor-like space using bezier curves," in *Proceedings of the 2013 IEEE/SICE International Symposium on System Integration*, pp. 37–41, 2013.

[6] J.-H. Hwang, R. C. Arkin, and D.-S. Kwon, "Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 2, pp. 1444–1449 vol.2, 2003.

[7] K. Simba, N. Uchiyama, and S. Sano, *Real-time smooth trajectory generation for nonholonomic mobile robots using Bézier curves*, vol. 41. 10 2016.

[8] C. Chen, Y. He, C. Bu, J. Han, and X. Zhang, "Quartic bezier curve based trajectory generation for autonomous vehicles with curvature and velocity constraints," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6108–6113, 2014.

[9] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *J. ACM*, vol. 40, pp. 1048–1066, Nov. 1993.

[10] B. Lau, C. Sprunk, and W. Burgard, "Kinodynamic motion planning for mobile robots using splines," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2427–2433, 2009.

[11] J. W. Choi and K. Huhtala, "Constrained path optimization with bézier curve primitives," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 246–251, 2014.

[12] G. Klancar, D. Matko, and S. Blazic, "Mobile robot control on a reference path," in *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, pp. 1343–1348, 2005.

[13] H. Prautzsch, W. Boehm, and M. Paluszny, *Bezier and B-Spline Techniques*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2002.