Cooperative Collision Avoidance between Multiple Robots based on Bernstein-Bézier Curves

Igor Škrjanc and Gregor Klančar

Faculty of Electrical Engineering, University of Ljubljana
Tržaška 25, SI-1000 Ljubljana, Slovenia

Abstract. In this paper a new cooperative collision-avoidance method for multiple nonholonomic robots based on Bernstein- Bézier curves is presented. The reference path of each robot from the start pose to the goal pose, is obtained by minimizing the penalty function, which takes into account the sum of all the paths subjected to the distances between the robots, which should be bigger than the minimal distance defined as the safety distance. When the reference paths are defined the model predictive trajectory tracking is used to define the control. A prediction model derived from linearized tracking-error dynamics is used to predict future system behavior. A control law is derived from a quadratic cost function consisting of the system tracking error and the control effort. The results of the simulation and some future work ideas are discussed.

1 Introduction

Collision avoidance is one of the main issues in applications for a wide variety of tasks in industry, human-supported activities, and elsewhere. Often, the required tasks cannot be carried out by a single robot, and in such a case multiple robots are used cooperatively. The use of multiple robots may lead to a collision if they are not properly navigated. Collision-avoidance techniques tend to be based on speed adaptation, route deviation by one vehicle only, route deviation by both vehicles, or a combined speed and route adjustment. When searching for the best solution that will prevent a collision many different criteria are considered: time delay, total travel time, planned arrival time, etc. Our optimality criterion will be the minimal travel time, which directly implies a minimal total length of the robot paths, subject to a minimal safety distance between all the robots.

In the literature many different techniques for collision avoidance have been proposed. The first approaches proposed avoidance, when a collision between robots is predicted, by stopping the robots for a fixed period or by changing their directions. The combination of these techniques is proposed in [1]. The behavior-based motion planning of multiple mobile robots in a narrow passage is presented in [2]. Intelligent learning techniques were incorporated into neural and fuzzy control for mobile-robot navigation to avoid a collision as proposed in [3].

In our paper the control of multiple mobile robots to avoid collisions in a two-dimensional free-space environment is separated into the path planning for each individual robot to reach its goal pose as fast as possible. The second part of the task is to design the control that will ensure the perfect trajectory tracking of the mobile robots.

Škrjanc I. and Klančar G. (2007).
Cooperative Collision Avoidance between Multiple Robots based on Bernstein-Bézier Curves.
In *Proceedings of the 3rd International Workshop on Multi-Agent Robotic Systems*, pages 34-43

Several controllers were proposed for mobile robots with nonholonomic constraints. An extensive review of nonholonomic control problems can be found in [4]. In trajectory-tracking control a reference trajectory is usually obtained by using a reference robot; therefore, all kinematics constraints are implicitly considered by a reference trajectory. From the reference trajectory a feed-forward system of inputs combined with a feedback control law are mostly used [5]. Lyapunov stable time-varying state-tracking control laws were pioneered by [9]. The stabilization to the reference trajectory requires a nonzero motion condition. Many variations and improvements to this state-tracking controller followed in subsequent research [10]. A tracking controller obtained with input-output linearization is used in [5], a saturation feedback controller is proposed in [11] and a dynamic feedback linearization technique is used in [6].

The paper is organized as follows. In Section 2 the problem is stated. The concept of path planning is shown in Section 3. The idea of optimal collision avoidance for multiple mobile robots based on Bézier curves is discussed in Section 4. The trajectory-tracking controller design where the control strategy consists of feed-forward and feedback actions is introduced in Section 5. In Section 5.1 the proposed model predictive controller is derived. The simulation results of the obtained collision-avoidance control are presented in Section 6 and the conclusion is given in Section 7.

2 Statement of the Problem

The collision-avoidance control problem of multiple nonholonomic mobile robots is proposed in a two-dimensional free-space environment. The simulations are performed for a small two-wheel differentially driven mobile robot of dimension $7.5 \times 7.5 \times 7.5$ cm. The architecture of our robots has a nonintegrable constraint in the form $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$ resulting from the assumption that the robot cannot slip in a lateral direction where $q(t) = [x(t)\ y(t)\ \theta(t)]^T$ are the generalized coordinates The kinematics model of the mobile robot is

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

where $v(t)$ and $\omega(t)$ are the tangential and angular velocities of the platform. During low-level control the robot's velocities and accelerations are bounded within the maximal allowed velocities and accelerations, which prevents the robot from slipping.

The danger of a collision between multiple robots is avoided by determining the strategy of the robots' navigation, where we define the reference path to fulfil certain criteria. The reference path of each robot from the start pose to the goal pose is obtained by minimizing the penalty function, which takes into account the sum of all the paths subjected to the distances between the robots, which should be larger than the defined safety distance. When the reference paths are defined the model predictive trajectory tracking is used to define the control.

3 Path Planning based on Bernstein-Bézier Curves

Given a set of control points $P_0, P_1, \ldots, P_b$, the corresponding Bernstein-Bézier curve (or Bézier curve) is given by

$$\mathbf{r}(\lambda) = \sum_{i=0}^{b} B_{i,b}(\lambda)\mathbf{p}_i$$

where $B_{i,b}(\lambda)$ is a Bernstein polynomial, λ is a normalized time variable ($\lambda = t/T_{max}$, $0 \le \lambda \le 1$) and $\mathbf{p}_i,\ 0 = 1, \ldots, b$ stands for the local vectors of the control point P_i ($\mathbf{p}_i = P_{i_x}\mathbf{e}_x + P_{i_y}\mathbf{e}_y$, where $P_i = \left(P_{i_x}, P_{i_y}\right)$ is the control point with coordinates P_{i_x} and P_{i_y}, and $\mathbf{e}_x$ and $\mathbf{e}_y$ are the corresponding base unity vectors). The Bernstein-Bézier polynomials, which are the base functions in the Bézier-curve expansion, are given as follows:

$$B_{i,b}(\lambda) = \binom{b}{i} \lambda^i \left(1-\lambda\right)^{b-i},\ i = 0, 1, \ldots, b$$

which have the following properties: $0 \le B_{i,b}(\lambda) \le 1,\ 0 \le (\lambda) \le 1$ and $\sum_{i=0}^{b} B_{i,b} = 1$.

The Bézier curve always passes through the first and last control point and lies within the convex hull of the control points. The curve is tangent to the vector of the difference $\mathbf{p}_1 - \mathbf{p}_0$ at the start point and to the vector of the difference $\mathbf{p}_b - \mathbf{p}_{b-1}$ at the goal point. A desirable property of these curves is that the curve can be translated and rotated by performing these operations on the control points. The undesirable properties of Bézier curves are their numerical instability for large numbers of control points, and the fact that moving a single control point changes the global shape of the curve. The former is sometimes avoided by smoothly patching together low-order Bézier curves.

The properties of Bézier curves are used in path planning for nonholonomic mobile robots. In particular, the fact of the tangentiality at the start and at the goal points and the fact that moving a single control point changes the global shape of the curve. Let us assume the starting pose of the mobile robot is defined in the generalized coordinates as $\mathbf{q}_s = [x_s, y_s, \theta_s]^T$ and the goal pose is defined as $\mathbf{q}_g = [x_g, y_g, \theta_g]^T$, which means that the robot starts in position $P_s(x_s, y_s)$ with orientation θ_s and has a goal defined with position $P_g(x_g, y_g)$ with orientation θ_g. The property of tangentiality requires the definition of the neighboring points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$, which become

$$P_1(x_s + d\cos\theta_s, y_s + d\sin\theta_s),\ P_2(x_g + d\cos(\theta_g + \pi), y_g + d\sin(\theta_g + \pi)) \quad (2)$$

where d stands for the distance between P_s and P_1 and between P_g and P_2. The distance d is usually defined relatively to the distance between the start and the goal point D ($D = \mid \mathbf{p}_g - \mathbf{p}_s \mid$) defined as $d = \gamma D,\ 0 < \gamma < 0.5$. These four control points P_s, P_1, P_2 and P_g uniformly define the third order Bézier curve. The need for flexibility of the global shape and the fact that moving a single control point changes the global shape of the curve imply the introduction of another point, which will be denoted as $P_o(x_o, y_o)$. By changing the position of point P_o the global shape of the curve changes. This means that having in mind the flexibility of the global shape of the curve and the start and the goal pose of the mobile robot, the path can be planned by four fixed points and one variable point. The Bézier curve is now defined as a sequence of points P_s, P_1, P_o, P_2 and P_g in Fig 1. This means that we are dealing with Bernstein polynomials of the fourth order ($B_{i,b},\ i=0,\ldots,b,\ b=4$). The curve is defined as follows:

$$\mathbf{r}(\lambda)=B_{0,4}\mathbf{p}_s+B_{1,4}\mathbf{p}_1+B_{2,4}\mathbf{p}_o+B_{3,4}\mathbf{p}_2+B_{4,4}\mathbf{p}_g \tag{3}$$

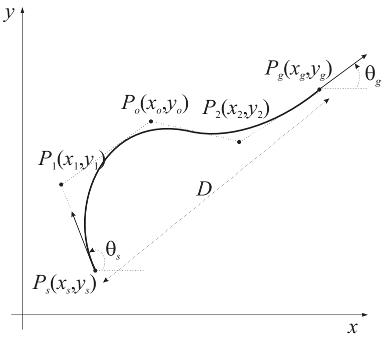

Fig. 1. The Bézier curve.

4 Optimal Collision Avoidance based on Bernstein-Bézier Curves

In this subsection a detailed presentation of cooperative multiple robots collision avoidance based on Bézier curves will be given. Let as assume the number of robots equals n. The i-th robot is denoted as R_i and has the start position defined as $P_{si}\,(x_{si},y_{si})$ and the goal position defined as $P_{gi}\,(x_{gi},y_{gi})$. The reference path of i-th robot will be denoted with the Bézier curve $\mathbf{r}_i(\lambda)=[x_i(\lambda),y_i(\lambda)]^T$. By choosing maximal time of the experiment T_{max} ($t=T_{max}\lambda$, $0\leq\lambda\leq1$) the robots tangential velocity profiles are determined. T_{max} is determined by the fastest robot R_i as $T_{max}=\frac{\max_i(v_i(\lambda))}{v_{max}}$, $i=1,\ldots,n,\ 0\leq\lambda\leq1$, where v_{max} is maximal allowed tangential robot velocity. Maximal time T_{max} is then common to all robots R_i. In Fig. 2 a collision avoidance for $n=2$ is presented for reasons of simplicity.

The safety margin to avoid a collision between two robots is, in this case, defined as the minimal necessary distance between these two robots. The distance between the robot R_i and R_j is $r_{ij}(\lambda)=\mid\mathbf{r}_i(\lambda)-\mathbf{r}_j(\lambda)\mid,\ i=1,\ldots,n,\ j=1,\ldots,n,\ i\neq j$.

Defining the minimal necessary safety distance as d_s, the following condition for collision avoidance is obtained $r_{ij}\geq d_s,\ 0\leq\lambda\leq1,\ i,j$. Fulfilling this criteria means that the robots will never meet in the same region defined by a circle with radius d_s, which is called a non-overlapping criterion. At the same time we would like to minimize the length of the path for each robot, which is defined as s_i. The length $s_i(\lambda)$ is defined as $s_i(\lambda)=\int_0^\lambda v_i(\lambda)d\lambda$, where $v_i(\lambda)$ stands for the tangential velocity in the normalized

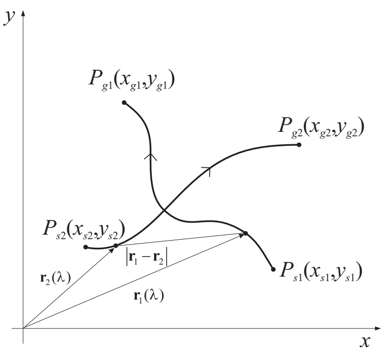

Fig. 2. Collision avoidance based on Bernstein-Bézier.

variable λ and the length of the path of the robot R_i from the start control point to the goal point is now calculated as:

$$v_i(\lambda) = \mid \dot{\mathbf{r}}(\lambda) \mid = \left(\dot{x}_i^2(\lambda) + \dot{y}_i^2(\lambda)\right)^{\frac{1}{2}}, \; s_i = \int_0^1 \left((\dot{x}_i^2(\lambda)) + \dot{y}_i^2(\lambda))\right)^{\frac{1}{2}} d\lambda$$

where $\dot{x}_i(\lambda)$ stands for $\frac{dx_i(\lambda)}{d\lambda}$ and $\dot{y}_i(\lambda)$ for $\frac{dy_i(\lambda)}{d\lambda}$. Assuming that the start and goal control points are known, the global shape and length of each path can be optimized by changing the flexible control point P_{oi}. The collision-avoidance problem is now defined as an optimization problem as follows:

$$minimize \sum_{i=1}^{n} s_i \; subject\ to\ d_s - r_{ij}(\lambda) \leq 0, \; \forall i, j, \; i \neq j, \;\; 0 \leq \lambda \leq 1 \qquad (4)$$

The minimization problem is called an *inequality optimization* problem and can be introduced as the minimization of the following penalty function

$$F(\mathbf{P}_o) = \sum_i s_i + c \sum_{ij} \Gamma_{ij}, \;\; \Gamma_{ij} = \begin{cases} 1, & \min r_{ij}(\lambda) < d_s \\ 0, & \min r_{ij}(\lambda) > d_s \end{cases}, \;\; i, j, \; i \neq j, \;\; 0 \leq \lambda \leq 1 \quad (5)$$

where c stands for a large scalar to penalize the unfulfillment of constraints. The solution of the minimization problem $\min_{\mathbf{P}_o} F$ is a set of n control points $\mathbf{P}_o = \{P_{o1}, \ldots, P_{on}\}$. Each optimal control point $P_{oi}, \; i = 1, \ldots, n$ uniformly defines one optimal path, which ensures collision avoidance in the sense of a safety distance and will be used as a reference trajectory of the ith robot and will be denoted as $\mathbf{r}_{ri}(\lambda)$.

To define the feasible reference path that will be collision safe, the real time should be introduced. In the real system the tangential and the angular velocities are limited to $(v_{max}, \; \omega_{max})$. Using the relation $v(t) = \frac{v(\lambda)}{T_{max}}$ the maximal time T_{max} can be defined to fulfil the velocity limitation ($T_{max} \geq \frac{\max v(\lambda)}{v_{max}}$).

5 Path Tracking

The previously obtained optimal collision-avoidance path for the ith robot is defined as $\mathbf{r}_{ri}(t) = [x_{ri}(t), y_{ri}(t)]^T, \; i = 1, \ldots, n$. In this section the development of a predictive path-tracking controller will be presented. The path-tracking control is realized as a sum of the feed-forward and feed-back controls. The feed-forward control for the ith robot is calculated from a feasible reference path $\mathbf{r}_{ri}(t) = [x_{ri}(t), y_{ri}(t)]^T$, which enables us to reach a desired pose. The feed-forward control inputs $v_{ri}(t)$ and $\omega_{ri}(t)$ are derived using a kinematic model (1). The tangential velocity $v_{ri}(t)$ and the tangent angle of each point on the path are calculated as follows

$$v_{ri}(t) = \left(\dot{x}_{ri}^2(t) + \dot{y}_{ri}^2(t)\right)^{\frac{1}{2}}, \quad \omega_{ri}(t) = \frac{\dot{x}_{ri}(t)\ddot{y}_{ri}(t) - \dot{y}_{ri}(t)\ddot{x}_{ri}(t)}{\dot{x}_{ri}^2(t) + \dot{y}_{ri}^2(t)} = v_{ri}(t)\kappa(t) \tag{6}$$

where $\kappa(t)$ is the path curvature. The necessary condition in the path-design procedure is a twice-differentiable path and a nonzero tangential velocity $v_{ri}(t) \neq 0$.

If for some time t the tangential velocity is $v_{ri}(t)$=0, the robot rotates at a fixed point with the angular velocity $\omega_{ri}(t)$ calculated from an explicitly given $\theta_{ri}(t)$.

The feedback control law is derived from a linear time-varying system obtained by an approximate linearization around the trajectory. The obtained linearization is shown to be controllable as long as the trajectory does not come to a stop, which implies that the system can be asymptotically stabilized by smooth time-varying linear or nonlinear feedback. The tracking error $\mathbf{e}(t) = [e_1(t)\; e_2(t)\; e_3(t)]^T$ of a mobile robot expressed in the frame of the real robot reads

$$\mathbf{e} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} (\mathbf{q}_{ri} - \mathbf{q})\,. \tag{7}$$

Considering the robot kinematics (1) and derivating relations (7) the following kinematics model is obtained

$$\dot{\mathbf{e}} = \begin{bmatrix} \cos e_3 & 0 \\ \sin e_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{ri} \\ \omega_{ri} \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \mathbf{u} \tag{8}$$

where $\mathbf{u} = [v\,\omega]^T$ is the velocity input vector and v_{ri} and ω_{ri} are already defined in (6). The robot input vector $\mathbf{u}$ is further defined as the sum of the feed-forward and feedback control actions ($\mathbf{u} = \mathbf{u}_F + \mathbf{u}_B$) where the feed-forward input vector, $\mathbf{u}_F$, is obtained by a nonlinear transformation of the reference inputs $\mathbf{u}_F = [v_{ri}\cos e_3\;\; \omega_{ri}]^T$ and the feedback input vector, is $\mathbf{u}_B = [u_{B_1}\, u_{B_2}]^T$, which is the output of the controller defined in section 5.1.

Using the relation $\mathbf{u} = \mathbf{u}_F + \mathbf{u}_B$, rewriting (8) and furthermore, by linearizing the error dynamics around the reference trajectory ($e_1 = e_2 = e_3 = 0$, $u_{B_1} = u_{B_2} = 0$) the following linear model is obtained

$$\dot{\mathbf{e}} = \begin{bmatrix} 0 & \omega_{ri} & 0 \\ -\omega_{ri} & 0 & v_{ri} \\ 0 & 0 & 0 \end{bmatrix} \mathbf{e} + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \mathbf{u}_B \tag{9}$$

which in the state-space form is $\dot{\mathbf{e}} = \mathbf{A}_c\mathbf{e} + \mathbf{B}_c u_B$. According to Brockett's condition [12] a smooth stabilization of the system (1) or its linearization is only possible with time-varying feedback. In the following the obtained linear model is used in the derived predictive control law.

5.1 Model Predictive Control based on a Robot Tracking-error Model

To design the controller for trajectory tracking the system (9) will be written in discrete-time form as

$$\mathbf{e}(k+1) = \mathbf{A}\mathbf{e}(k) + \mathbf{B}\mathbf{u}_B(k)$$

where $\mathbf{A} \in \mathbb{R}^n \times \mathbb{R}^n$, n is the number of state variables and $\mathbf{B} \in \mathbb{R}^n \times \mathbb{R}^m$, m is the number of input variables. The discrete matrix $\mathbf{A}$ and $\mathbf{B}$ can obtained as follows:$\mathbf{A} = \mathbf{I} + \mathbf{A}_c T_s, \quad \mathbf{B} = \mathbf{B}_c T_s$ which is a good approximation during a short sampling time T_s.

The idea of the moving-horizon control concept is to find the control-variable values that minimize the receding-horizon quadratic cost function (in a certain interval denoted with h) based on the predicted robot-following error:

$$J(u_B, k) = \sum_{i=1}^{h} \epsilon^T(k,i)\mathbf{Q}\epsilon(k,i) + \mathbf{u}_B^T(k,i)\mathbf{R}\mathbf{u}_B(k,i) \tag{10}$$

where $\epsilon(k,i) = \mathbf{e}_{ri}(k+i) - \mathbf{e}(k+i|k)$ and $\mathbf{e}_{ri}(k+i)$ and $\mathbf{e}(k+i|k)$ stands for the reference robot following-trajectory and the robot-following error, respectively, and $\mathbf{Q}$ and $\mathbf{R}$ stand for the weighting matrices where $\mathbf{Q} \in \mathbb{R}^n \times \mathbb{R}^n$ and $\mathbf{R} \in \mathbb{R}^m \times \mathbb{R}^m$, with $\mathbf{Q} \geq 0$ and $\mathbf{R} \geq 0$.

Output prediction in the discrete-time framework In the moving time frame the model output prediction at the time instant h can be written as:

$$\mathbf{e}(k+h|k) = \Pi_{j=1}^{h-1}\mathbf{A}(k+j|k)\mathbf{e}(k) + \sum_{i=1}^{h}\left(\Pi_{j=i}^{h-1}\mathbf{A}(k+j|k)\right)\mathbf{B}(k+i-1|k)\cdot \\ \cdot\mathbf{u}_B(k+i-1) + +\mathbf{B}(k+h-1|k)\mathbf{u}_B(k+h-1)\,. \tag{11}$$

Defining the robot-tracking prediction-error vector

$$\mathbf{E}^*(k) = \left[e(k+1|k)^T \; e(k+2|k)^T \ldots e(k+h|k)^T\right]^T$$

where $\mathbf{E}^* \in \mathbb{R}^{n \cdot h}$ for the whole interval of observation (h) and the control vector

$$\mathbf{U}_B(k) = \left[\mathbf{u}_B^T(k) \; \mathbf{u}_B^T(k+1) \ldots \mathbf{u}_B^T(k+h-1)\right]^T$$

and

$$\boldsymbol{\Lambda}(k,i) = \Pi_{j=i}^{h-1}\mathbf{A}(k+j|k)$$

the robot-tracking prediction-error vector is written in the form

$$\mathbf{E}^*(k) = \mathbf{F}(k)\mathbf{e}(k) + \mathbf{G}(k)\mathbf{U}_B(k) \tag{12}$$

where

$$\mathbf{F}(k) = \left[\mathbf{A}(k|k) \; \mathbf{A}(k+1|k)\mathbf{A}(k|k) \; \ldots \; \boldsymbol{\Lambda}(k,0)\right]^T, \tag{13}$$

and $\mathbf{G}(k) = [g_{ij}]\,,\; i = 1,...,n,\; j = 1,...,b,\; b = max(h,n)$, $g_{11} = \mathbf{B}(k|k)$,$g_{21} = \mathbf{A}(k+1|k)\mathbf{B}(k|k)$, $g_{22} = \mathbf{B}(k+1|k)$, $g_{n1} = \boldsymbol{\Lambda}(k,1)\mathbf{B}(k|k)$, $g_{n2} = \boldsymbol{\Lambda}(k,2)\mathbf{B}(k+1|k)$, $g_{nh} = \mathbf{B}(k+h-1|k)$. and $\mathbf{F}(k) \in \mathbb{R}^{n \cdot h} \times \mathbb{R}^n$, $\mathbf{G}(k) \in \mathbb{R}^{n \cdot h} \times \mathbb{R}^{m \cdot h}$.

The objective of the control law is to drive the predicted robot trajectory as close as possible to the future reference trajectory, i.e., to track the reference trajectory. This implies that the future reference signal needs to be known. Let us define the reference error-tracking trajectory in state-space as $\mathbf{e}_{ri}(k+i) = \mathbf{A}_{ri}^{i}\mathbf{e}(k)$. for $i = 1, \ldots, h$. This means that the future control error should decrease according to dynamics defined with the reference model matrix $\mathbf{A}_{ri}$. Defining the robot reference-tracking error vector

$$\mathbf{E}_{ri}^{*}(k) = \left[\mathbf{e}_{ri}(k+1)^{T} \; \mathbf{e}_{ri}(k+2)^{T} \ldots \mathbf{e}_{ri}(k+h)^{T}\right]^{T}, \; \mathbf{E}_{ri}^{*} \in \mathbb{R}^{n \cdot h}$$

for the whole interval of observation (h) the following is obtained

$$\mathbf{E}_{ri}^{*}(k) = \mathbf{F}_{ri}e(k), \; \mathbf{F}_{ri} = \left[\mathbf{A}_{ri} \; \mathbf{A}_{ri}^{2} \; \ldots \; \mathbf{A}_{ri}^{h}\right]^{T}, \; \; \mathbf{F}_{ri} \in \mathbb{R}^{n \cdot h} \times \mathbb{R}^{n}. \tag{14}$$

Control law The idea of MPC is to minimize the difference between the predicted robot-trajectory error and the reference robot-trajectory error in a certain predicted interval.

The cost function is, according to the above notation, now written as

$$J(U_B) = \left(\mathbf{E}_{ri}^{*} - \mathbf{E}^{*}\right)^{T} \overline{\mathbf{Q}} \left(\mathbf{E}_{ri}^{*} - \mathbf{E}^{*}\right) + \mathbf{U}_{B}^{T}\overline{\mathbf{R}}\mathbf{U}_{B} \, . \tag{15}$$

The control law is obtained by the minimization ($\frac{\partial J}{\partial \mathbf{U}_B} = 0$) of the cost function and becomes

$$\mathbf{U}_B(k) = \left(\mathbf{G}^{T}\overline{\mathbf{Q}}\mathbf{G} + \overline{\mathbf{R}}\right)^{-1} \mathbf{G}^{T}\overline{\mathbf{Q}} \left(\mathbf{F}_{ri} - \mathbf{F}\right) \mathbf{e}(k) \tag{16}$$

where $\overline{\mathbf{Q}} = diag(\mathbf{Q})$ and $\overline{\mathbf{R}} = diag(\mathbf{R})$. This means that $\overline{\mathbf{Q}} \in \mathbb{R}^{n \cdot h} \times \mathbb{R}^{n \cdot h}$ and $\overline{\mathbf{R}} \in \mathbb{R}^{m \cdot h} \times \mathbb{R}^{m \cdot h}$.

Let us define the first m rows of the matrix $\left(\mathbf{G}^{T}\overline{\mathbf{Q}}\mathbf{G} + \overline{\mathbf{R}}\right)^{-1} \mathbf{G}^{T}\overline{\mathbf{Q}} \left(\mathbf{F}_{ri} - \mathbf{F}\right) \in \mathbb{R}^{m \cdot h} \times \mathbb{R}^{n}$ as $\mathbf{K}_{mpc}$. Now the feedback control law of the model predictive control is given by

$$\mathbf{u}_B(k) = \mathbf{K}_{mpc} \cdot \mathbf{e}(k), \; \mathbf{K}_{mpc} \in \mathbb{R}^{m} \times \mathbb{R}^{n} \tag{17}$$

6 Simulation Results

In this section the simulation results of the optimal cooperative collision avoidance between three mobile robots are shown. The study was made to elaborate the possible use in the case of a real mobile-robot platform. In the real platform we are faced with the limitation of control velocities and accelerations. The maximal allowed tangential velocity and angular velocity were $v_{max} = 0.5 \; m/s$ and $\omega_{max} = 13 \; rad/s$, while the maximal allowed tangential wheel acceleration is $a_{max} = 3m/s^2$. Because of relatively hight maximal angular velocity and tangential wheel acceleration only the tangential velocity was taken into account to define the maximal time between the start position of the robots and the goal position, which is defined as $T_{max} \geq \frac{\max v(\lambda)}{v_{max}} = \frac{2.1m}{0.5ms^{-1}} = 4.02s$ where the maximal normalized tangential velocity $\max_i v_i(\lambda) = 2.1m$ is defined from Fig. 3. The starting pose of the first mobile robot R_1 in generalized coordinates is defined as $\mathbf{q}_{s1} = \left[0, 1, \frac{\pi}{2}\right]^{T}$ and the goal pose as $\mathbf{q}_{g1} = \left[1, 0, -\frac{\pi}{4}\right]^{T}$. The second robot

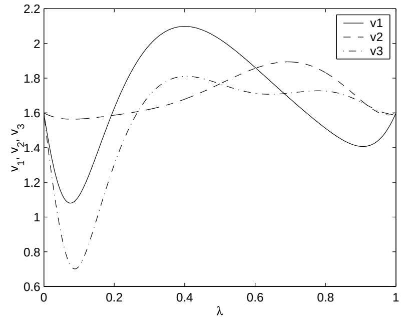

Fig. 3. The velocities of avoiding robots R_1, R_2 and R_3 in normalized time variable.

R_2 starts in $\mathbf{q}_{s2} = \left[1, 0, -\frac{3\pi}{4}\right]^T$ and has the goal pose $\mathbf{q}_{g2} = \left[0, 1, \frac{3\pi}{4}\right]^T$. The third robot R_3 has the start pose $\mathbf{q}_{s3} = \left[0, 0, -\frac{\pi}{4}\right]^T$ and the goal pose $\mathbf{q}_{g3} = \left[1, 1, \frac{\pi}{4}\right]^T$. The x and y coordinates are defined in meters. The safety distance is defined as $d_s = 0.35m$. The parameter d, which is used to define the control points P_{1i} and P_{2i}, equals $0.4m$ ($\min_{ij} D_{ij} = 1m, \gamma = 0.4$). In Fig. 4 the distances between the mobile robots are

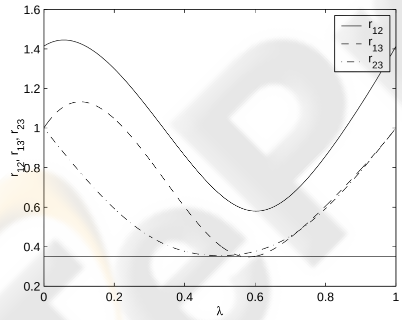

Fig. 4. The distances between avoiding robots R_1, R_2 and R_3.

shown. It is also shown that all the distances r_{12}, r_{13} and r_{23} satisfy the safety-distance condition. They are always bigger than prescribed safety distance d_s.

7 Conclusion

The optimal cooperative collision-avoidance approach based on Bézier curves allows us to include different criteria in the penalty functions. In our case the reference path of each robot from the start pose to the goal pose is obtained by minimizing the penalty function, which takes into account the sum of all the paths subjected to the distances between the robots, which should be bigger than the minimal distance defined as the safety distance. Current approach as presented does not include explicit velocity and acceleration constraints to be imposed to each robot, this remaining the future research work.

References

1. R.C. Arkin, Cooperation without communication: multiagent schema-based robot navigation, J Robot Syst., 1992, 9, 351–364,
2. L. Shan and T. Hasegawa, Space reasoning from action observation for motion planning of multiple robots: mutal collision avoidance in a narrow passage, J Robot Soc Japan, 1996, 14, 1003–1009.
3. C.G. Kim and M.M.Triverdi, A neuro-fuzzy controller for mobile robot navigation and multirobot convoying, IEEE Trans Sys Man Cybernet SMC Part-B, 1998, 28, 829–840.
4. I. Kolmanovsky and N. H. McClamroch, Developments in Nonholonomic Control Problems, IEEE Control Systems, 1995, 15, 6, 20–36.
5. N. Sarkar, X. Yun and V. Kumar, Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robot, The International Journal of Robotic Research, 1994, 13, 1, 55–69.
6. G. Oriolo, A. Luca and M. Vandittelli, WMR Control Via Dynamic Feedback Linearization: Design, Implementation, and Experimental Validation, IEEE Transactions on Control Systems Technology, 2002, 10, 6, 835–852.
7. C. Canudas de Wit and O. J. Sordalen, Exponential Stabilization of Mobile Robots with Nonholonomic Constraints, IEEE Transactions on Automatic Control, 1992, 37, 11, 1791–1797.
8. A. Luca, G. Oriolo and M. Vendittelli, Control of wheeled mobile robots: An experimental overview, RAMSETE - Articulated and Mobile Robotics for Services and Technologies, 2001, S. Nicosia, B.Siciliano, A. Bicchi, P. Valigi, Springer-Verlag.
9. Y. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi, A stable tracking control method for an autonomous mobile robot, Proceedings of the 1990 IEEE International Conference on Robotics and Automation, 1990, 1, 384–389.
10. F. Pourboghrat and M. P. Karlsson, Adaptive control of dynamic mobile robots with nonholonomic constraints, Computers and electrical Engineering, 2002, 28, 241–253.
11. T. C. Lee, K. T. Song, C. H. Lee and C. C. Teng, Tracking control of unicycle-modeled mobile robots using a saturation feedback controller, IEEE Transactions on Control Systems Technology, 2001, 9, 2, 305–318.
12. R. W. Brockett, Asymptotic stability and feedback stabilization, Differential Geometric Control Theory, 1983, R. W. Brockett, R.S. Millman and H. J. Sussmann, 181–191.