

# A Case Study of the Collision-Avoidance Problem Based on Bernstein–Bézier Path Tracking for Multiple Robots with Known Constraints

Gregor Klančar · Igor Škrjanc

Received: 23 April 2009 / Accepted: 25 March 2010 / Published online: 9 April 2010  
© Springer Science+Business Media B.V. 2010

**Abstract** In this paper a case study of a new, cooperative, collision-avoidance method for multiple, nonholonomic robots based on Bernstein–Bézier curves is given. In the presented examples the velocities and accelerations of the mobile robots are constrained and the start and the goal velocity are defined for each robot. This means that the proposed method can be used as a subroutine in a huge path-planning problem in real time, in a way to split the whole path into smaller partial paths. 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 path lengths subjected to the distances between the robots, which should be bigger than the minimum distance defined as the safety distance, and subjected to the velocities and accelerations which should be lower than the maximum allowed for each robot. When the reference paths are defined the model-predictive trajectory tracking is used to define the control. The prediction model derived from the linearized tracking-error dynamics is used to predict future system behavior. The control law is derived from a quadratic cost function consisting of the system tracking error and the control effort. The proposed method was tested with a simulation and with a real-time experiment in which four robots were used.

**Keywords** Mobile robots · Collision avoidance · Path planning · Bernstein–Bézier curves · Predictive control

---

G. Klančar · I. Škrjanc (✉)  
Faculty of Electrical Engineering, University of Ljubljana,  
Tržaška 25, 1000 Ljubljana, Slovenia  
e-mail: igor.skrjanc@fe.uni-lj.si

G. Klančar  
e-mail: gregor.klancar@fe.uni-lj.si

## 1 Introduction

When dealing with mobile, autonomous robots one is always faced with the serious problem of collision avoidance, which is one of the main issues in applications for a wide variety of tasks in industry. When the required tasks cannot be carried out by a single robot, then multiple robots are used cooperatively. However, this may lead to a collision if they are not properly navigated. Different paradigms of collision avoidance are known. Some of them are based on speed adaptation, route deviation by one vehicle only, route deviation by both vehicles, or a combined speed and route adjustment. The optimal solution to prevent a collision can be defined in many different ways to fulfill different criteria. The most frequently used are: the planned arrival time the total traveling distance or time and the time delay. In our case the optimal criterion will be the minimum total traveling distance of all the mobile robots involved in the task, subject to the constraints of a minimal safety distance between all the robots and subject to the velocity and acceleration constraints of each mobile robot.

Many different approaches to collision avoidance have been proposed in the past. The simplest approaches solve the problem of collision by stopping the robots for a fixed period or by changing their directions, when the collision is predicted based on the robots' kinematic models. A combination of both techniques is proposed in [1] and [21]. The behavior-based motion planning of multiple mobile robots in a narrow passage is presented in [20]. Some of the approaches involve the intelligent learning techniques in the avoidance algorithm using neural and fuzzy theory to control and navigate the autonomous mobile robots. This is suggested in [8] and [12]. Some adaptive navigation techniques for mobile robots' navigation also appeared, as proposed in [4].

Our approach deals with cooperative collision avoidance. This means that we are defining the paths of all the robots off-line in a cooperative manner by changing their paths to achieve the goal. The whole task of the collision avoidance of multiple mobile robots in a two-dimensional, free-space environment is mainly separated into two parts: the path planning for each individual robot to reach its goal pose as fast as possible, and the trajectory-tracking control to follow the optimal path. The first part is the design of an optimal trajectory for each robot, which is made based on Bernstein–Bézier curves and optimization. In the second part the design of the control that will ensure the perfect trajectory tracking of the real mobile robots is proposed. Several different trajectory-tracking control techniques were proposed for mobile robots with nonholonomic constraints. An extensive review of nonholonomic control problems can be found in [11]. The trajectory-tracking control approach usually defines a reference trajectory by using a reference robot; therefore, all the kinematics constraints can be implicitly involved in a reference trajectory. Based on the reference trajectory a feed-forward control is made, which is then combined with a feedback control law. These kinds of approaches are the most widely used [2, 16, 19]. The stable, time-varying, state-tracking control laws based on Lyapunov theory were pioneered by [7] and [18]. Some variations and improvements to the basic state-tracking controller followed in subsequent research [3, 17]: a tracking controller obtained with input–output linearization is used in [19], a saturation feedback controller is proposed in [6], and a dynamic feedback linearization technique is used in [16].

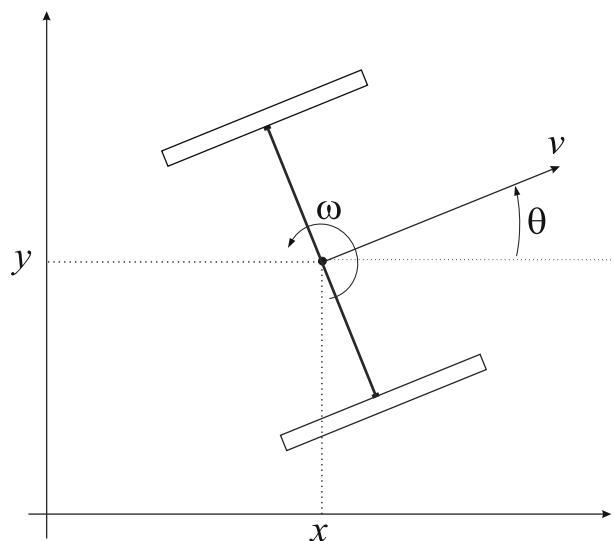
The predictive approaches to path-tracking seem to be very promising in the field of mobile robotics because of the existence of a precise kinematic model and the fact that the reference trajectory is known beforehand. For example, in [15] a generalized predictive control is chosen to control the mobile robot, minimizing the quadratic cost function. The modification of a generalized predictive controller using the Smith predictor to cope with an estimated system time-delay is presented in [14]. The time-varying description of the system together with model predictive control is proposed in [13]. And finally, in [5] the multi-layer neural network predictive-controller is described. The main idea of the proposed control approach is to minimize the difference between the future robot trajectory and the reference-trajectory path. One of the most important advantages of the proposed control law is analytical derivation [10]; therefore, it is computationally effective and can be easily used in fast, real-time implementations.

The paper is organized in the following way. 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. In Section 5 the proposed model predictive controller is presented. The simulation and experimental results of the obtained collision-avoidance control are presented in Section 6 and the conclusion is given in Section 7.

## 2 Description of the Collision-Avoidance Control Problem

The study of collision-avoidance was performed for multiple, nonholonomic mobile robots in a two-dimensional, free-space environment. The mobile robots are small, two-wheel, differentially driven vehicles. The scheme is given in Fig. 1. The architecture of our robots has a non-integrable 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

**Fig. 1** The generalized coordinates of the mobile robot—differential drive



$q(t) = [x(t) \ y(t) \ \theta(t)]^T$  are the generalized coordinates, as defined in Fig. 1. The kinematics model of the mobile robot is described in Eq. 1

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

where  $v(t)$  and  $\omega(t)$  are the tangential and angular velocities of the robot. The low-level control of the robot's ensures the bounded velocity, which prevents the robot from slipping.

To prevent the robots from colliding the strategy of the robots' navigation is determined, where we define the reference path of each robot 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 all the important variables of the problem. These variables are the sum of all the absolute maximum times subjected to the distances between the robots, which should be larger than the defined safety distance and the maximum allowed velocities and the maximum allowed acceleration of each mobile robot.

### 3 Path Planning Based on Bernstein–Bézier Curves

A Bernstein–Bézier curve of  $b$ -th order is defined by a set of control points  $P_0, P_1, \dots, P_b$ . The corresponding Bernstein–Bézier curve in a parametrical form (or Bézier curve) is given as follows

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

where  $B_{i,b}(\lambda)$  is a Bernstein polynomial,  $\lambda$  is a normalized time variable ( $\lambda = t/T_{\max}$ ,  $0 \leq \lambda \leq 1$ ) and  $\mathbf{p}_i$ ,  $0 = 1, \dots, b$  stands for the local vectors of the control point  $P_i$  ( $\mathbf{p}_i = P_{ix} \mathbf{e}_x + P_{iy} \mathbf{e}_y$ , where  $P_i = (P_{ix}, P_{iy})$  is the control point with coordinates  $P_{ix}$  and  $P_{iy}$ , and  $\mathbf{e}_x$  and  $\mathbf{e}_y$  are the corresponding base unity vectors). The absolute maximum time  $T_{\max}$  is the time needed to pass the path between the start control point  $P_0$  and the goal control point  $P_b$ . The Bernstein–Bézier polynomials, which are the base functions in the Bézier-curve expansion, are given as follows:

$$B_{i,b}(\lambda) = \binom{b}{i} \lambda^i (1 - \lambda)^{b-i}, \quad i = 0, 1, \dots, b$$

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

The property of a Bézier curve is that it always passes through the first and last control point and lies within the convex hull of the control points. The curve is tangential 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 last control 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. This is not always undesirable and is usefully used in our approach.

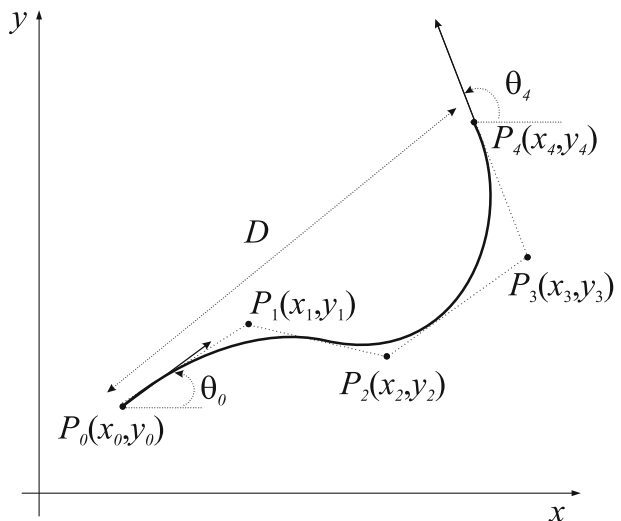
The most important properties of Bézier curves are used in path planning for nonholonomic mobile robots. In particular, the fact of the tangentiality at the start and at the end control points and the fact that moving a single control point changes the global shape of the curve. Let us assume the starting pose of the mobile robot is defined in the generalized coordinates as  $\mathbf{q}_0 = [x_0, y_0, \theta_0]^T$  and the velocity in the start pose as  $v_0$ . The goal pose is defined as  $\mathbf{q}_b = [x_b, y_b, \theta_b]^T$  with the velocity in the end pose as  $v_b$ . This means that the robot starts in position  $P_0(x_0, y_0)$  with the orientation  $\theta_0$  and the velocity  $v_0$  and has a goal defined with the position  $P_b(x_b, y_b)$ , the orientation  $\theta_b$  and the velocity  $v_b$ .

This means that having in mind the flexibility of the global shape of the curve and the start and the end pose of the mobile robot, the path can be planned using four fixed points and one variable control point, as shown in Fig. 2. The control points  $P_1(x_1, y_1)$  and  $P_3(x_3, y_3)$  are defined to fulfill the velocity and orientation requirements in the path. 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 a flexible control point denoted as  $P_2(x_2, y_2)$ . By changing the position of point  $P_2$  the global shape of the curve changes. The Bernstein polynomials of the fourth order ( $B_{i,b}, i = 0, \dots, b, b = 4$ ), and the control points define the curve as follows: or

$$\mathbf{r}(\lambda) = (1 - \lambda)^4 \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + 4\lambda(1 - \lambda)^3 \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + 6\lambda^2(1 - \lambda)^2 \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + 4\lambda^3(1 - \lambda) \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} + \lambda^4 \begin{bmatrix} x_4 \\ y_4 \end{bmatrix} \tag{2}$$

The control point  $P_2$  will be defined using optimization, and the control points  $P_1$  and  $P_3$  are defined from the boundary velocity conditions. The velocity as a function

**Fig. 2** The Bézier curve with five points



of the normalized time  $\lambda$  is obtained from a derivation of the path vector  $\mathbf{r}(\lambda)$  as follows:

$$\mathbf{v}(\lambda) = \frac{d\mathbf{r}(\lambda)}{d\lambda} = b \sum_{i=0}^{b-1} (\mathbf{p}_{i+1} - \mathbf{p}_i) B_{b-1,i}(\lambda) \tag{3}$$

If the path vector is of  $b$ -th order, the velocity vector  $\mathbf{v}(\lambda)$  becomes  $(b - 1) - st$  order. At the start position ( $\lambda = 0$ ) all the Bernstein polynomials become zero  $B_{b-1,i}(0) = 0, i = 1, \dots, b - 1$  except  $B_{b-1,0}(0)$ , which equals 1. And in the goal position ( $\lambda = 1$ ) all the Bernstein polynomials become zero  $B_{b-1,i}(1) = 0, i = 1, \dots, b - 2$  except  $B_{b-1,b-1}(0)$  which equals 1. This defines the velocity vectors at the start position  $\mathbf{v}(0) = 4(\mathbf{p}_1 - \mathbf{p}_0)$  and in the goal position  $\mathbf{v}(1) = 4(\mathbf{p}_4 - \mathbf{p}_3)$ , which means that the vectors of the control points  $\mathbf{p}_1$  and  $\mathbf{p}_3$  are defined as follows:

$$\mathbf{p}_1 = \mathbf{p}_0 + \frac{1}{4}\mathbf{v}(0), \quad \mathbf{p}_3 = \mathbf{p}_4 - \frac{1}{4}\mathbf{v}(1) \tag{4}$$

Using the known tangential velocities and the orientations of the robot at the start and at the end position, the velocity vector in normalized time can be decomposed into the  $v_x$  and  $v_y$  components as follows:

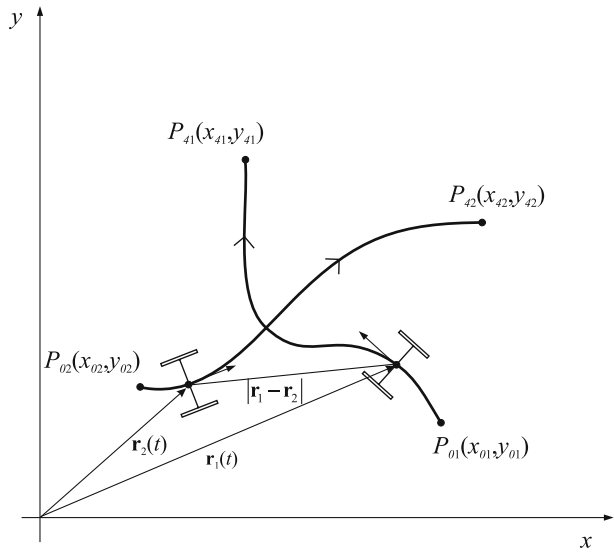
$$\begin{aligned} \mathbf{v}(0) &= [v_x(0) \ v_y(0)]^T = [v(0) \cos \theta_0 \ v(0) \sin \theta_0]^T \\ \mathbf{v}(1) &= [v_x(1) \ v_y(1)]^T = [v(1) \cos \theta_4 \ v(1) \sin \theta_4]^T \end{aligned} \tag{5}$$

where  $v(0)$  and  $v(1)$  stand for the start and the end tangential velocities of the robot. Using Eqs. 4 and 5, the control points  $P_1$  and  $P_3$  are uniformly defined. The only unknown control point remains  $P_2$ , which is defined using optimization to obtain the optimal path that is collision-safe, and which fulfills all the constraints. This approach with one free point is used in the sense of computation time and can be used when the number of robots is relatively small (around 10). When we are dealing with a larger number of robots, the number of free points should be increased, which also increases the computing time.

### 4 Path Planning for Optimal Cooperative Collision Avoidance

A detailed presentation of the path planning for cooperative, multiple-robots, collision avoidance based on Bézier curves will be given next. The path is obtained on the basis of an optimization and by taking into account the velocity and acceleration constraints of the mobile robots. The  $i$ -th robot is denoted as  $R_i$  and has a start position defined as  $P_{0i}(x_{0i}, y_{0i})$  and a goal position defined as  $P_{4i}(x_{4i}, y_{4i})$ . The normalized time variable of the  $i$ -th robot is denoted as  $\lambda_i = t / T_{\max_i}$ , where  $T_{\max_i}$  stands for the absolute maximum time of the  $i$ -th robot. The reference path will be denoted by the curve  $\mathbf{r}_i(\lambda_i) = [x_i(\lambda_i), y_i(\lambda_i)]^T$ . The number of robots treated in the problem is denoted as  $n$ . In Fig. 3 the path planning for two robots ( $n = 2$ ) is presented. To avoid a collision between the robots, a safety margin is defined. This safety margin is the minimum allowed distance between the two robots. The distance between the two robots,  $R_i$  and  $R_j$ , is  $r_{ij}(t) = |\mathbf{r}_i(t) - \mathbf{r}_j(t)|, i = 1, \dots, n, j = 1, \dots, n, i \neq j$ . By

**Fig. 3** Collision avoidance for two robots based on Bernstein–Bézier



defining the safety distance as  $d_s$ , the following condition for collision avoidance is obtained

$$r_{ij} \geq d_s, \quad 0 \leq \lambda \leq 1, \quad i, j$$

By fulfilling these criteria, the robots will never meet in the same region, defined by a circle with radius  $d_s$ , which is called a non-overlapping criterion. In parallel the sum of the travelled paths  $s_i$  of all the robots has to be minimized. The length of the path of the  $i$ -th robot at the normalized time is defined as  $s_i(\lambda_i) = \int_0^{\lambda_i} v_i(\lambda_i) d\lambda_i$  where

$$v_i(\lambda_i) = |\dot{\mathbf{r}}(\lambda_i)| = \left( v_{xi}^2(\lambda_i) + v_{yi}^2(\lambda_i) \right)^{\frac{1}{2}}$$

and where  $v_{xi}(\lambda_i)$  stands for  $\frac{dx_i(\lambda_i)}{d\lambda_i}$  and  $v_{yi}(\lambda_i)$  for  $\frac{dy_i(\lambda_i)}{d\lambda_i}$ .

The feasible reference-path trajectory should also satisfy the constraints of the maximum velocity  $v_{\max_i}$  and the maximum acceleration  $a_{\max_i}$  of the mobile robot, which are allowed in the sense of physical realization. The relationship between the tangential velocity and the acceleration in the normalized time framework and the real tangential velocity and acceleration is the following

$$v_i(\lambda_i) = T_{\max_i} v_i(t) \quad , \quad a_i(\lambda_i) = T_{\max_i}^2 a_i(t)$$

The length of the path of the robot  $R_i$  from the start control point to the goal point is now calculated as:

$$s_i = \int_0^1 \left( v_{xi}^2(\lambda_i) + v_{yi}^2(\lambda_i) \right)^{\frac{1}{2}} d\lambda_i$$

The start  $P_{0i}$ , the goal  $P_{4i}$  and the  $P_{1i}$  and  $P_{3i}$  control points are known, the global shape and length of each path can be optimized by changing the flexible control

point  $P_{2i}$ . The collision-avoidance problem is solved with an optimization problem as follows:

$$\begin{aligned}
 & \min \sum_{i=1}^n s_i \\
 & \text{subject to } d_s - r_{ij}(t) \leq 0, \quad v_i(t) - v_{\max_i}, \quad a_i(t) - a_{\max_i} \leq 0 \\
 & \forall i, j, i \neq j, \quad 0 \leq t \leq \max_i (T_{\max_i})
 \end{aligned} \tag{6}$$

The minimization problem is called an *inequality optimization* problem. The methods using penalty functions transform a constrained problem into an unconstrained problem. The constraints are placed into the objective function via the penalty parameter in such a way as to penalize any violation of the constraints. In our case the following penalty function should be used to have an unconstrained optimization problem

$$\begin{aligned}
 F &= \sum_i s_i + c_1 \sum_{ij} \max_{ij} (0, 1/r_{ij}(t) - 1/d_s) + c_2 \sum_i \max_i (0, v_i(t) - v_{\max_i}) \\
 & \quad + c_3 \sum_i \max_i (0, a_i(t) - a_{\max_i}) \\
 \min \quad & F \\
 \text{subject to } & \mathbf{P}_2, \mathbf{T}_{\max} \\
 & i, j, i \neq j, \quad 0 \leq t \leq \max_i (T_{\max_i})
 \end{aligned} \tag{7}$$

where  $c_1, c_2$  and  $c_3$  stand for a large scalar to penalize the violation of the constraints and the solution of the minimization problem is a set of  $n$  control points  $\mathbf{P}_2 = \{P_{21}, \dots, P_{2n}\}$  and  $\mathbf{T}_{\max}$  is a set of  $n$  maximum times  $\mathbf{T}_{\max} = \{T_{\max_1}, \dots, T_{\max_n}\}$ . Each optimal control point  $P_{2i}, i = 1, \dots, n$  uniformly defines one optimal path, which ensures collision avoidance in the sense of a safety distance and will be used as a reference trajectory for the  $i$ th robot and will be denoted as  $\mathbf{r}_i(\lambda)$ . The optimal solution is also subjected to the time, because the velocities and accelerations of the robots are also taken into account in the penalty function in Eq. 7.

### 5 Trajectory-Tracking Control Law

The proposed trajectory-tracking control law  $\mathbf{u}$  is composed of two parts: a feedforward  $\mathbf{u}_f$  and a feedback control law  $\mathbf{u}_b$ . The trajectory-tracking control signal is the sum of both the control signals  $\mathbf{u} = \mathbf{u}_f + \mathbf{u}_b$ .

#### 5.1 Feedforward Control

The feedforward control for the robot is calculated from a feasible reference path for that robot denoted as  $\mathbf{r}_r(t) = [x_r(t), y_r(t)]^T$ , which enables us to reach a desired pose. The feedforward part compensates for the nonlinearities of the plant and forces the output of the system close to the reference trajectory. The reference trajectory can also be represented by the reference robot that ideally follows the reference path with the tangential velocity  $v_r(t)$  and the angular velocity  $\omega_r(t)$  in real time, which are both calculated from  $\mathbf{r}_r(t)$  as follows

$$v_r = (v_{rx}^2 + v_{ry}^2)^{\frac{1}{2}} \tag{8}$$



and

$$\omega_r = \frac{v_{rx}a_{ry} - v_{ry}a_{rx}}{v_{rx}^2 + v_{ry}^2} \tag{9}$$

where  $v_{rx}, v_{ry}$  and  $a_{rx}, a_{ry}$  stand for the  $x$  and  $y$  components of the tangential velocity ( $v_{rx} = \dot{x}_r, v_{ry} = \dot{y}_r$ ) and acceleration ( $a_{rx} = \ddot{x}_r, a_{ry} = \ddot{y}_r$ ) in real time. The necessary condition in the path-design procedure is a twice-differentiable path and a non-zero tangential velocity  $v_r \neq 0$ .

The tracking error  $\mathbf{e} = [e_1 \ e_2 \ e_3]^T$  of a mobile robot is defined as the error between the reference robot and the real robot and is expressed in the frame of the real-robot coordinates as follows

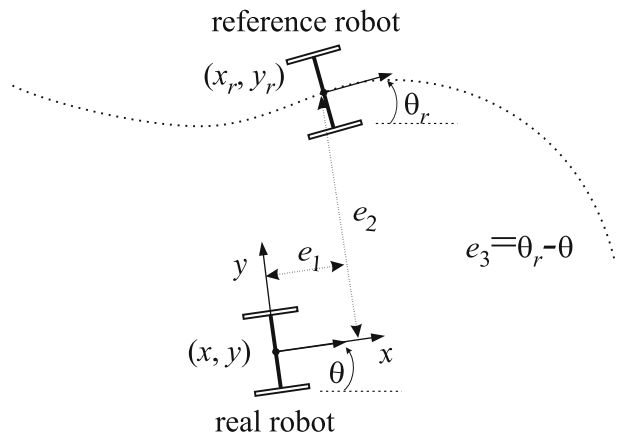
$$\mathbf{e} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} (\mathbf{q}_r - \mathbf{q}) \tag{10}$$

where  $\mathbf{q}_r$  is the generalized coordinates vector of the reference robot and  $\mathbf{q}$  is the generalized coordinates vector of the corresponding real robot. In Fig. 4 the tracking-error transformation is given. The tracking-error model defines the errors between the real robot and the reference robot in real-robot coordinates. Considering the robot kinematics defined in Eq. 1 and using the tracking-error model from Eq. 10 in its derivative form the following nonlinear dynamic kinematics error model is obtained

$$\dot{\mathbf{e}} = \begin{bmatrix} \cos e_3 & 0 \\ \sin e_3 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}_r + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \mathbf{u} \tag{11}$$

where  $\mathbf{u}_r = [v_r \ \omega_r]^T$  stands for the input vector of the reference robot. The input consists of  $v_r$  and  $\omega_r$ , defined in Eqs. 8 and 9, and  $\mathbf{u}$  is the real-robot input vector, defined as  $\mathbf{u} = [v \ \omega]^T = \mathbf{u}_f + \mathbf{u}_b$ .

**Fig. 4** The tracking-error transformation



The feedforward control signal is defined to force the real robot to the equilibrium  $\dot{e}_1 = \dot{e}_2 = \dot{e}_3 = 0$ , which is defined with the reference trajectory. An approximate solution of the equilibrium defined in Eq. 11.

$$v_r \cos e_3 - v + \omega e_2 = 0 \tag{12}$$

$$v_r \sin e_3 - \omega e_1 = 0 \tag{13}$$

$$\omega_r - \omega = 0 \tag{14}$$

is given as  $\mathbf{u}_f = [v_r \cos e_3 \ \omega_r]^T$ .

### 5.2 Feedback Control

The feedback control law is defined to compensated for the fine error between the reference trajectory and the real robots. The design of the feedback control is based on the deviation kinematic model. The deviation model is obtained by linearization around the reference trajectory, which is the equilibrium solution ( $e_1 = e_2 = e_3 = 0$  and  $v_b = \omega_b = 0$ ) of the system, where  $\mathbf{u}_b = [v_b \ \omega_b]^T$  stands for the feedback control signal of the real robot.

Inserting the input vector  $\mathbf{u} = \mathbf{u}_f + \mathbf{u}_b = [v_r \cos e_3 \ \omega_r]^T + [v_b \ \omega_b]^T$  into Eq. 11, the resulting model is given by

$$\begin{aligned} \dot{e}_1 &= \omega_r e_2 - v_b + e_2 \omega_b \\ \dot{e}_2 &= -\omega_r e_1 + v_r \sin e_3 - e_1 \omega_b \\ \dot{e}_3 &= -\omega_b \end{aligned} \tag{15}$$

The linearization of the nonlinear dynamic kinematics error model in Eq. 15 around the reference trajectory is as follows

$$\dot{\mathbf{e}} = \begin{bmatrix} 0 & \omega_r + \omega_b & 0 \\ -\omega_r - \omega_b & 0 & v_r \cos e_3 \\ 0 & 0 & 0 \end{bmatrix}_{\mathbf{u}_b=0} \mathbf{e} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix}_{\mathbf{e}=0} \mathbf{u}_b \tag{16}$$

This results in the linear, time-varying, tracking-error kinematic model given in Eq. 17.

$$\dot{\mathbf{e}} = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} \mathbf{e} + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \mathbf{u}_b \tag{17}$$

which can be, in more compact form, described as  $\dot{\mathbf{e}} = \mathbf{A}_c \mathbf{e} + \mathbf{B}_c \mathbf{u}_b$ .

#### 5.2.1 Predictive Control Based on a Robot Tracking-Error Model

The design of the feedback control law using the predictive control paradigm is realized in discrete time. The tracking-error model is therefore transformed in discrete time 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 and the discrete matrix  $\mathbf{A}$  and  $\mathbf{B}$  can be obtained as  $\mathbf{A} = \mathbf{I} + \mathbf{A}_c T_s$  and  $\mathbf{B} = \mathbf{B}_c T_s$ , which is a good approximation when a short sampling time  $T_s$  is used.

The main goal 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 by  $h$ ) based on the predicted robot-following error:

$$J(u_B, k) = \sum_{i=1}^h \boldsymbol{\varepsilon}^T(k, i) \mathbf{Q} \boldsymbol{\varepsilon}(k, i) + \mathbf{u}_B^T(k, i) \mathbf{R} \mathbf{u}_b(k, i) \tag{18}$$

where  $\boldsymbol{\varepsilon}(k, i) = \mathbf{e}_i(k+i) - \mathbf{e}(k+i)$  and  $\mathbf{e}_i(k+i)$  and  $\mathbf{e}(k+i)$  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  $\mathbf{R} \geq 0$ .

In the moving time frame the model output prediction at the time instant  $h$  can be written as:

$$\begin{aligned} \mathbf{e}(k+h) &= \mathbf{S}(k)\mathbf{e}(k) + \sum_{i=1}^h \mathbf{S}(k)\mathbf{B}(k+i-1)\mathbf{u}_b(k+i-1) + \mathbf{B}(k+h-1)\mathbf{u}_B(k+h-1) \\ \mathbf{S}(k) &= \Pi_{j=1}^{h-1} \mathbf{A}(k+j) \end{aligned} \tag{19}$$

Defining the robot-tracking prediction-error vector

$$\mathbf{E}^*(k) = [e(k+1)^T \ e(k+2)^T \ \dots \ e(k+h)^T]^T$$

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

$$\mathbf{U}_b(k) = [\mathbf{u}_b^T(k) \ \mathbf{u}_b^T(k+1) \ \dots \ \mathbf{u}_b^T(k+h-1)]^T$$

and

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

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

$$\mathbf{E}^*(k) = \mathbf{F}(k)\mathbf{e}(k) + \mathbf{G}(k)\mathbf{U}_b(k) \tag{20}$$

where

$$\mathbf{F}(k) = [\mathbf{A}(k) \ \mathbf{A}(k+1)\mathbf{A}(k) \ \dots \ \boldsymbol{\Lambda}(k, 0)]^T, \tag{21}$$

and

$$\mathbf{G}(k) = \begin{bmatrix} \mathbf{B}(k) & 0 & \dots & 0 \\ \mathbf{A}(k+1)\mathbf{B}(k) & \mathbf{B}(k+1) & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Lambda}(k, 1)\mathbf{B}(k) & \boldsymbol{\Lambda}(k, 2)\mathbf{B}(k+1) & \dots & \mathbf{B}(k+h-1) \end{bmatrix} \tag{22}$$

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 actual robot trajectory as close as possible to the reference trajectory. This can be achieved by predicting the future behavior of the robot using a kinematic model and minimizing the error between the predicted robot trajectory and the reference trajectory. This implies that the future reference signal needs to be known. Let us define the reference error-tracking trajectory for the  $i$ -th robot in state-space as

$$\mathbf{e}_i(k+i) = \mathbf{A}_i^i \mathbf{e}(k) \tag{23}$$

for  $i = 1, \dots, h$ . This means that the future control error should decrease according to the dynamics defined by the reference model matrix  $\mathbf{A}_i$ . Defining the robot reference tracking-error vector

$$\mathbf{E}_i^*(k) = [\mathbf{e}_i(k+1)^T \ \mathbf{e}_i(k+2)^T \ \dots \ \mathbf{e}_i(k+h)^T]^T$$

where  $\mathbf{E}_i^* \in \mathbb{R}^{n \cdot h}$  for the whole interval of observation ( $h$ ) the following is obtained

$$\mathbf{E}_i^*(k) = \mathbf{F}_i \mathbf{e}(k), \quad \mathbf{F}_i = [\mathbf{A}_i \ \mathbf{A}_i^2 \ \dots \ \mathbf{A}_i^h]^T \tag{24}$$

and  $\mathbf{F}_i \in \mathbb{R}^{n \cdot h} \times \mathbb{R}^n$

### 5.2.2 Predictive Control Law

The main goal of the predictive control law 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

$$J(\mathbf{U}_b) = (\mathbf{E}_i^* - \mathbf{E}^*)^T \overline{\mathbf{Q}} (\mathbf{E}_i^* - \mathbf{E}^*) + \mathbf{U}_b^T \overline{\mathbf{R}} \mathbf{U}_b \tag{25}$$

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

$$\mathbf{U}_b(k) = (\mathbf{G}^T \overline{\mathbf{Q}} \mathbf{G} + \overline{\mathbf{R}})^{-1} \mathbf{G}^T \overline{\mathbf{Q}} (\mathbf{F}_i - \mathbf{F}) \mathbf{e}(k) \tag{26}$$

where

$$\overline{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & 0 & \dots & 0 \\ 0 & \mathbf{Q} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{Q} \end{bmatrix}, \quad \overline{\mathbf{R}} = \begin{bmatrix} \mathbf{R} & 0 & \dots & 0 \\ 0 & \mathbf{R} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{R} \end{bmatrix} \tag{27}$$

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

$$\mathbf{u}_b(k) = \mathbf{K}_{mpc} \cdot \mathbf{e}(k) \tag{28}$$

with  $\mathbf{K}_{mpc} \in \mathbb{R}^m \times \mathbb{R}^n$ .

## 6 Simulation and Experimental Results

In this section the path-planning results of the optimal, cooperative, collision-avoidance strategy between two and three mobile robots are shown and the experimental results obtained on a real platform using model-predictive, trajectory-tracking control are given. The study was made to elaborate a possible use in the case of a real mobile-robot platform. In a real platform we are faced with the limitation of control velocities and accelerations. The sampling time for all the presented simulations and experiments is  $T_s = 33$  ms. Additional details about the real set-up and video clips of the experiments are available at our website [9]. The study was made for two and four mobile robots.

### 6.1 Case Study for Two Mobile Robots

The maximum allowed tangential velocity and the maximum allowed acceleration of the first mobile robot are  $v_{\max_1} = 0.3$  m/s and  $a_{\max_1} = 0.4$  m/s<sup>2</sup>. The maximum allowed tangential velocity and maximum allowed acceleration of the second mobile robot are defined as  $v_{\max_2} = 0.25$  m/s and  $a_{\max_2} = 0.4$  m/s<sup>2</sup>.

The starting pose of the first mobile robot  $R_1$  in generalized coordinates is defined as  $\mathbf{q}_{01} = [0.2, 1, -\frac{\pi}{4}]^T$  and the goal pose as  $\mathbf{q}_{41} = [1, 0.5, -\frac{3\pi}{4}]^T$ . The boundary velocities of the first mobile robot are the start tangential velocity  $v_1(0) = 0.10$  m/s and the goal tangential velocity  $v_1(T_{\max_1}) = 0.10$  m/s. The second robot  $R_2$  starts in  $\mathbf{q}_{02} = [1, 0.2, -\frac{3\pi}{4}]^T$  and has the goal pose  $\mathbf{q}_{42} = [0.6, 1, -\frac{3\pi}{4}]^T$ . The boundary velocities of the second mobile robot are the start tangential velocity  $v_2(0) = 0.10$  m/s and the goal tangential velocity  $v_2(T_{\max_2}) = 0.10$  m/s. The  $x$  and  $y$  coordinates are defined in meters. The safety distance is defined as  $d_s = 0.40$  m.

The optimal set  $\mathbf{P}_2$  can be found by using one of the unconstrained optimization methods, but the initial conditions are very important. The optimization should be started with initial parameters that ensure a feasible solution. We are optimizing the total sum of all the paths that are subjected to certain conditions relating to the safety distances and velocities of the robots. The velocity condition implies the implementation of the maximum time for each robot into the optimization routine. This implies that the initial set  $\mathbf{P}_2$  will be defined as

$$\mathbf{P}_2 = \{(x_{21}, y_{21}), (x_{22}, y_{22})\}$$

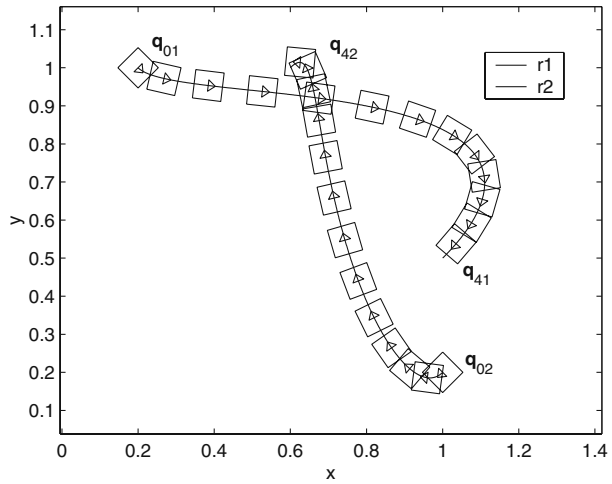
where  $x_{2i}$  and  $y_{2i}$  are defined as follows:

$$x_{2i} = \frac{x_{0i} + x_{4i}}{2}, \quad y_{2i} = \frac{y_{0i} + y_{4i}}{2}, \quad i = 1, 2$$

The initial maximum times are defined as  $T_{\max_1} = 10$  s and  $T_{\max_2} = 20$  s to fulfill the maximum velocity constraints. The penalty function (7) parameters are  $c_1 = 100$ ,  $c_2 = 100$  and  $c_3 = 100$ . The obtained results of the optimization routine are the following  $P_{21}(1.4552, 1.0113)$ ,  $P_{22}(0.6138, 0.5883)$  and  $T_{\max_1} = 6.4374$  s and  $T_{\max_2} = 6.4375$  s. The minimum value of the penalty function  $F$  is 2.2511.

The calculated trajectories of both robots that are cooperatively avoiding the collision are shown in Fig. 5. The robot shapes in Fig. 5 are drawn over the planned

**Fig. 5** The paths of collision-avoiding robots  $R_1$  and  $R_2$

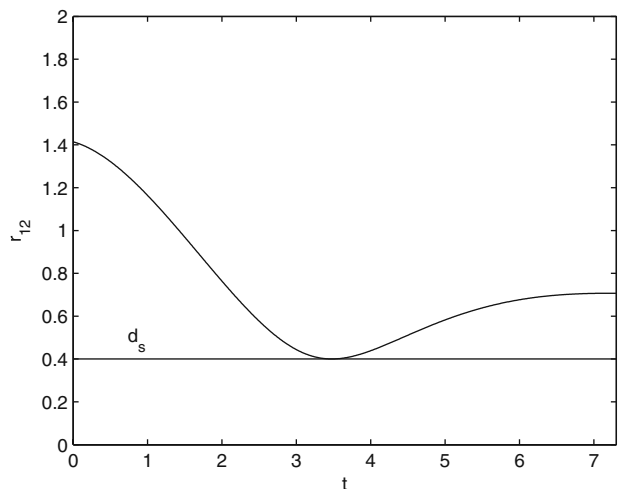


trajectory each 0.5 s. In Fig. 6 the distances between the mobile robots are shown. It is also shown that all the distances  $r_{12}$  satisfy the safety-distance condition. They are always bigger than prescribed safety distance  $d_s$ .

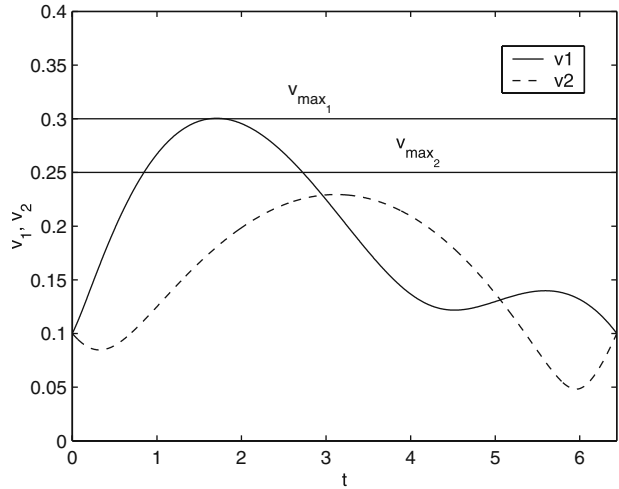
The real tangential velocity profiles of the avoiding robots  $R_1$  and  $R_2$  in the time variable are given in Fig. 7. It shows that the velocity profiles of both robots fulfill the boundary-velocities requirements and also fulfill the allowed maximum velocities conditions. The acceleration profiles of the robots  $R_1$  and  $R_2$  in time variable are given in Fig. 8. All the accelerations fulfill the allowed maximum acceleration conditions.

In Fig. 9 the results of the experiment performed on a small-sized, real-robots platform (the size of each robot is  $7.5 \times 7.5 \times 7.5\text{cm}$ ) are shown. It is clear that

**Fig. 6** The distance  $r_{12}$  between robots  $R_1$  and  $R_2$



**Fig. 7** The real velocities of the avoiding robots  $R_1$  and  $R_2$

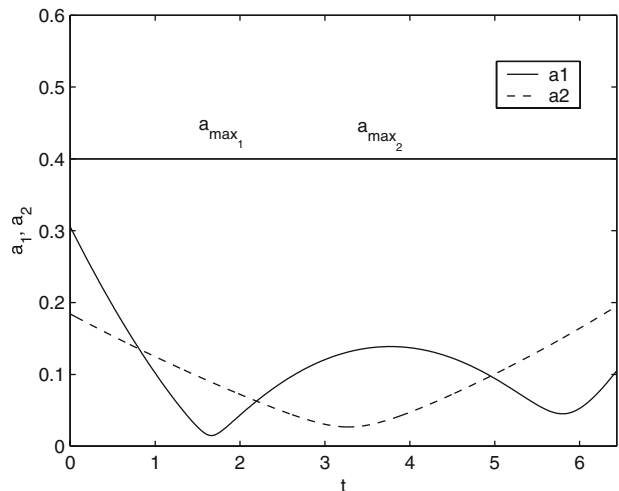


robots' initial postures  $q_{01}$  and  $q_{02}$  have some initial error relating to the planned trajectories. These initial errors were introduced intentionally to demonstrate the operation of the designed predictive controller. The predictive controller successfully drives the robot to follow the reference trajectories despite the noise in the position (standard deviation of 2 mm) and orientation measurements (standard deviation of 0.1 rad).

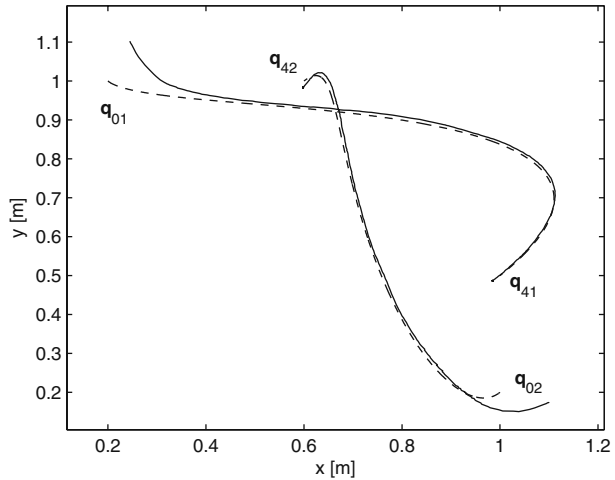
### 6.2 Case Study for Four Mobile Robots

The maximum allowed tangential velocities of the mobile robots are  $v_{max_i} = 0.8$  m/s and the maximum allowed accelerations are  $a_{max_i} = 0.5$  m/s<sup>2</sup> where  $i = 1, 2, 3, 4$ .

**Fig. 8** The real accelerations of the avoiding robots  $R_1$  and  $R_2$



**Fig. 9** The control of the collision-avoiding robots  $R_1$  and  $R_2$  (solid line) on the reference trajectories (dashed line); real experiment



The starting poses and the goal poses for the robots  $R_1, R_2, R_3,$  and  $R_4$  are  $\mathbf{q}_{01} = [0.8, 0.8, \frac{\pi}{4}]^T, \mathbf{q}_{41} = [0.2, 0.2, -\frac{3\pi}{4}]^T, \mathbf{q}_{02} = [1.4, 0.2, \frac{3\pi}{4}]^T, \mathbf{q}_{42} = [0.2, 1.4, \frac{\pi}{2}]^T, \mathbf{q}_{03} = [0.2, 1.4, -\frac{\pi}{4}]^T, \mathbf{q}_{43} = [1.4, 0.2, 0]^T,$  and  $\mathbf{q}_{04} = [0.2, 0.2, \frac{\pi}{4}]^T, \mathbf{q}_{44} = [1.4, 1.4, \frac{\pi}{4}]^T,$  respectively. The boundary velocities of the robots  $R_i$  consisting of the start tangential velocities  $v_i(0) = 0.25$  m/s and the goal tangential velocities  $v_i(T_{\max_i}) = 0.25$  m/s, where  $i = 1, 2, 3, 4$ . The safety distance is selected as  $d_s = 0.25$  m.

The optimal set  $\mathbf{P}_2$  is found by using the proposed unconstrained optimization method where the initial set  $\mathbf{P}_2$  is defined as

$$\mathbf{P}_2 = \{(x_{21}, y_{21}), (x_{22}, y_{22}), (x_{23}, y_{23}), (x_{24}, y_{24})\}$$

where  $x_{2i}$  and  $y_{2i}$  are defined as follows:

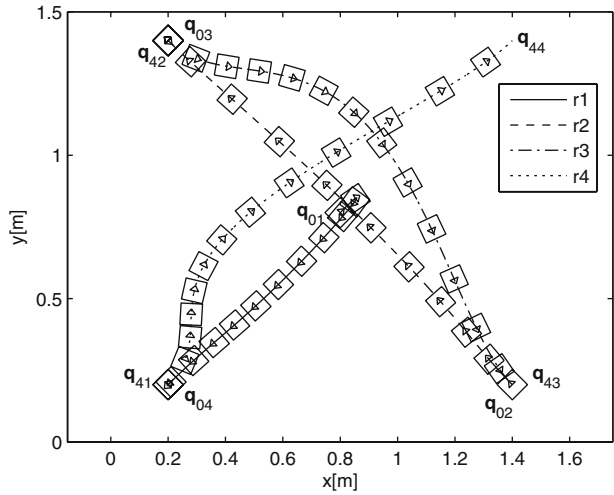
$$x_{2i} = \frac{x_{0i} + x_{4i}}{2}, \quad y_{2i} = \frac{y_{0i} + y_{4i}}{2}, \quad i = 1, 2, 3, 4$$

The initial maximum times are defined as  $T_{\max_i} = 5$  s ( $i = 1, 2, 3, 4$ ) to fulfill the maximum velocity constraints. The penalty function (7) parameters are  $c_i = 100$  ( $i = 1, 2, 3$ ). The obtained results of the optimization routine are the following  $P_{21}(0.49, 0.40), P_{22}(1.14, 0.61), P_{23}(1.00, 1.67), P_{24}(-0.21, 0.65)$  and  $T_{\max_1} = 5.5562$  s,  $T_{\max_2} = 4.8588$  s,  $T_{\max_3} = 6.2665$  s and  $T_{\max_4} = 6.4044$  s. The minimum value of the penalty function  $F$  is 6.9524.

The simulated positions of all robots ( $R_1, R_3, R_3$  and  $R_4$ ) that are cooperatively avoiding the collision are shown in Fig. 10. Robot  $R_1$  starts from the centre and finishes in the lower left corner; robot  $R_2$  starts from the lower right and finishes in the upper left corner; robot  $R_3$  starts from the upper left and finishes in the lower right corner; and robot robot  $R_4$  starts from the lower left and finishes in the upper right corner. Obviously, the paths of these robots cross. The robots (the square shapes on the trajectories in Fig. 10) are drawn for each tenth sample time to illustrate their progress during the experiment. It is clear that the robots adjust their



**Fig. 10** The paths of the collision-avoiding robots  $R_1$ ,  $R_2$ ,  $R_3$  and  $R_4$

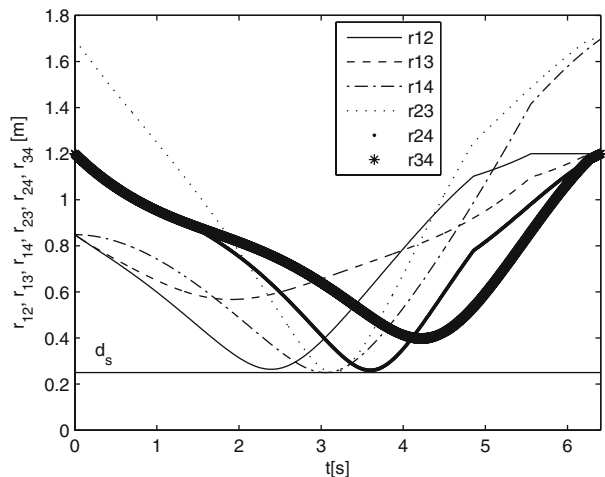


velocity profiles as well as their trajectories in order to fulfill the design constraints ( $d_s$ ,  $v_{max_i}$  and  $a_{max_i}$ ).

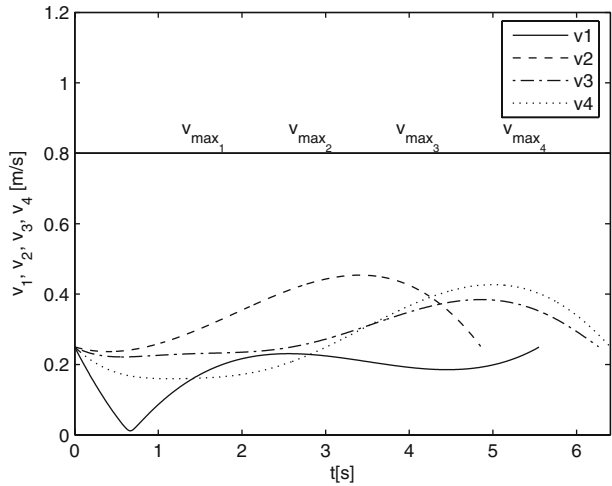
In Fig. 11 the distances between the mobile robots are shown. It is also clear that all the distances ( $r_{ij}$ , ( $ij = 12, 13, 14, 23, 24, 34$ )) satisfy the safety-distance condition. They are always larger than the prescribed safety distance  $d_s$ .

The real tangential velocity profiles of the avoiding robots  $R_1$ ,  $R_2$ ,  $R_3$  and  $R_4$  are given in Fig. 12. It is clear that the velocity profiles of all three robots fulfill the boundary-velocity requirements and the allowed maximum velocity conditions. The acceleration profiles of the robots  $R_1$ ,  $R_2$ ,  $R_3$  and  $R_4$  are given in Fig. 13. All the accelerations fulfill the allowed maximum acceleration conditions.

**Fig. 11** The distances  $r_{ij}$ , ( $ij = 12, 13, 14, 23, 24, 34$ ) between the robots  $R_1$ ,  $R_2$ ,  $R_3$  and  $R_4$



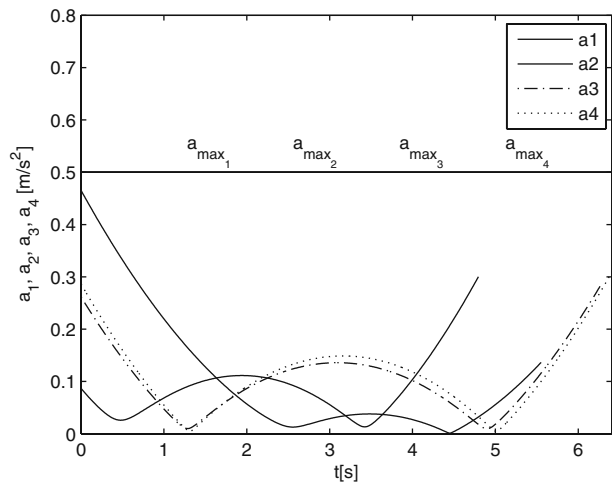
**Fig. 12** The real velocities of the avoiding robots  $R_1, R_2, R_3$  and  $R_4$



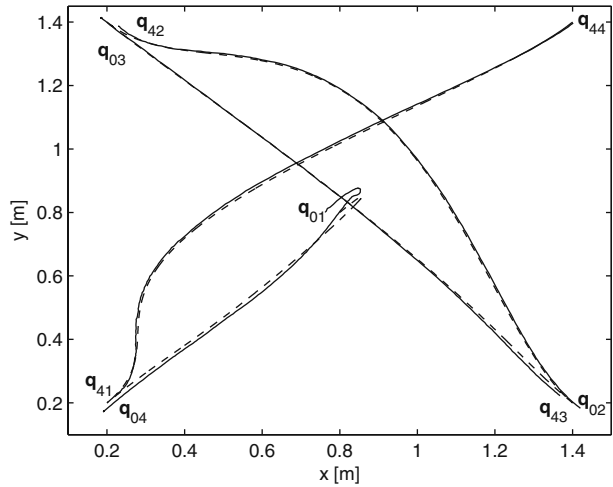
The proposed algorithm includes an optimization to find a solution for the collision-avoidance task, which makes it computationally demanding. For a smaller number of robots the real-time operation can still be achieved, especially if a faster computer is used. When increasing the number of robots the optimization problem becomes more computationally intense. In the presented examples it took some 0.25 s to compute the two-robot experiment and 0.8 s for the four-robot experiment on a Pentium IV 1.8 GHz computer in the Matlab environment.

Figure 14 shows the results of the experiment performed on a small-sized, real-robots platform (the size of each robot is  $7.5 \times 7.5 \times 7.5\text{cm} \times$ ). Like in the two-robot experiment (Section 6.1) it is clear that the robots's initial postures  $\mathbf{q}_{01}, \mathbf{q}_{02}, \mathbf{q}_{03}$  and  $\mathbf{q}_{04}$  have some initial error (they are not on the planned trajectory). Nevertheless,

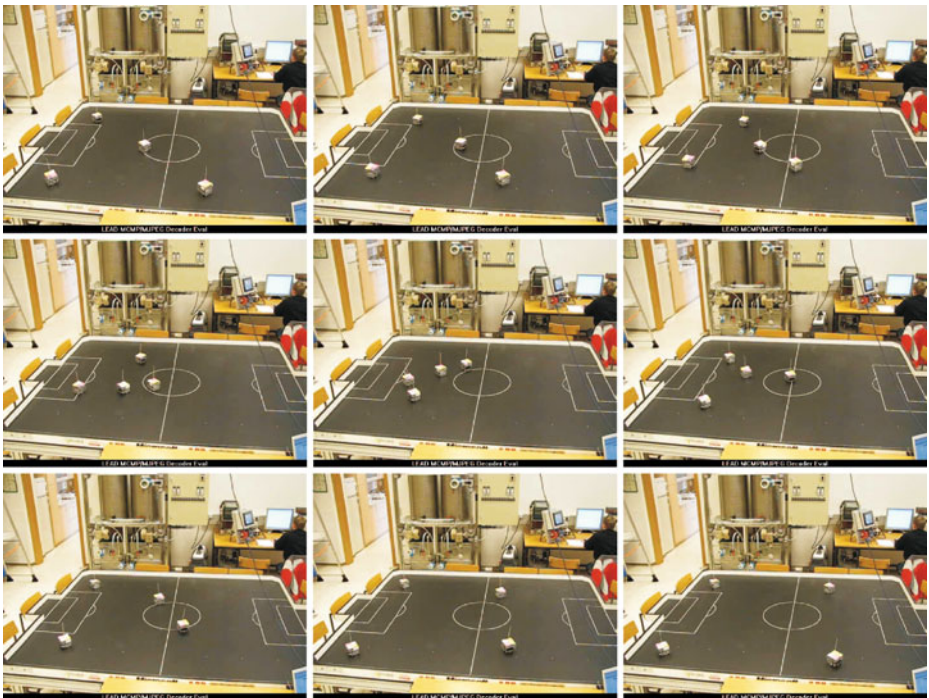
**Fig. 13** The real accelerations of the avoiding robots  $R_1, R_2, R_3$  and  $R_4$



**Fig. 14** The control of the collision-avoiding robots  $R_1$ ,  $R_2$ ,  $R_3$  and  $R_4$  (solid line) on the reference trajectories (dashed line); real experiment



the predictive controller successfully drives the robot on the reference trajectories despite the noise in the position and orientation measurements (for details about the noise rates see Section 6.1). The sequence of images during the experiment is shown in Fig. 15.



**Fig. 15** Sequence of images from the experiment in Fig. 14. Images are taken at a 0.76 s time interval

## 7 Conclusion

In this paper a collision-avoidance problem based on Bernstein–Bézier path tracking for multiple robots with known constraints has been shown. The optimal approach 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 travelled paths of the robots subjected to the distances between the robots, which should be larger than the minimum distance defined as the safety distance, the maximum velocities of the robots and the maximum allowed accelerations of the robots. A combination of feedforward and feedback control laws is used to control the robots on the obtained reference paths. The feedforward control law is obtained from the kinematic model and the reference trajectory, and the feedback control law is obtained from the kinematic error model. The solution to the predictive control is analytically derived, which enables fast real-time implementations. However, the path-optimization algorithm includes an optimization and is more computationally demanding, especially for a larger number of robots (more than ten). Future improvements will focus on decreasing the computational time of the optimization and on an increased robustness of the presented algorithm to larger tracking errors, mainly resulting from the wrong initial robot posture. The concept of landing curves, which guarantees an exponential convergence to the reference trajectory, will be included. The proposed cooperative collision-avoidance method for multiple nonholonomic robots based on Bézier curves and predictive reference tracking shows great potential and in the future will be implemented on a real, large-scale, mobile-robot, Pioneer 3-AT platform.

## References

1. Arkin, R.C.: Cooperation without communication: multiagent schema-based robot navigation. *J. Robot. Syst.* **9**(3), 351–364 (1992)
2. Balluchi, A., Bicchi, A., Balestrino, A., Casalino, G.: Path tracking control for Dubin's cars. In: *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, pp. 3123–3128, Minnesota (1996)
3. Pourboghrat, F., Karlsson, M.P.: Adaptive control of dynamic mobile robots with nonholonomic constraints. *Comput. Electr. Eng.* **28**(4), 241–253 (2002)
4. Fujimori, A., Nikiforuk, P.N., Gupta, M.M.: Adaptive navigation of mobile robots with obstacle avoidance. *IEEE Trans. Robot. Autom.* **13**(4), 596–602 (1999)
5. Gu, D., Hu, H.: Neural predictive control for a car-like mobile robot. *Robot. Auton. Syst.* **39**(2), 73–86 (2002)
6. Lee, T.C., Song, K.T., Lee, C.H., Teng, C.C.: Tracking control of unicycle-modeled mobile robots using a saturation feedback controller. *IEEE Trans. Control Syst. Technol.* **9**(2), 305–318 (2001)
7. Kanayama, Y., Kimura, Y., Miyazaki, F., Noguchi, T.: A stable tracking control method for an autonomous mobile robot. In: *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 384–389, Cincinnati (1990)
8. Kim, C.G., Triverdi, M.M.: A neuro-fuzzy controller for mobile robot navigation and multirobot convoying. *IEEE Trans. Syst. Man Cybern. Part B* **28**(6), 829–840 (1998)
9. Klančar, G.: Optimal collision avoidance experiments. <http://msc.fe.uni-lj.si/PublicWWW/Klančar/ColisionAvoidance.html> (2009). Accessed 28 December 2009
10. Klančar, G., Škrjanc, I.: Tracking-error model-based predictive control for mobile robots in real time. *Robot. Auton. Syst.* **55**(6), 460–469 (2007)
11. Kolmanovskiy, I., McClamroch, N.H.: Developments in nonholonomic control problems. *IEEE Control Syst.* **15**(6), 20–36 (1995)

12. Kubota, N., Morioka, T., Kojima, F., Fukuda, T.: Adaptive behavior of mobile robot based on sensory network. *JSME Trans.* **65**, 1006–1012 (1999)
13. Kühne, F., Gomes da Silva Jr., J.M., Lages, W.F.: Model predictive control of a mobile robot using linearization. In: *Mechatronics and Robotics 2004*, pp. 524–530, Aachen (2004)
14. Normey-Rico, J.E., Gomez-Ortega, J., Camacho, E.F.: A Smith-predictor-based generalised predictive controller for mobile robot path-tracking. *Control Eng. Pract.* **7**(6), 729–740 (1999)
15. Ollero, A., Amidi, O.: Predictive path tracking of mobile robots. Application to the CMU Navlab. In: *Proceedings of 5th International Conference on Advanced Robotics, Robots in Unstructured Environments (ICAR '91)*, vol. 2, pp. 1081–1086, Pisa (1991)
16. Oriolo, G., Luca, A., Vandittelli, M.: WMR control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Trans. Control Syst. Technol.* **10**(6), 835–852 (2002)
17. Raimondi, F.M., Melluso, M.: A new robust fuzzy dynamics controller for autonomous vehicles with nonholonomic constraints. *Robot. Auton. Syst.* **52**(2–3), 115–131 (2005)
18. Samson, C.: Time-varying feedback stabilization of car like wheeled mobile robot. *Int. J. Rob. Res.* **12**(1), 55–64 (1993)
19. Sarkar, N., Yun, X., Kumar, V.: Control of mechanical systems with rolling constraints: application to dynamic control of mobile robot. *Int. J. Rob. Res.* **13**(1), 55–69 (1994)
20. Shan, L., Hasegawa, T.: Space reasoning from action observation for motion planning of multiple robots: mutual collision avoidance in a narrow passage. *J. Robot. Soc. Japan* **14**, 1003–1009 (1996)
21. Sugihara, K., Suzuki, I.: Distributed algorithms for formation of geometric patterns with many mobile robots. *J. Robot. Syst.* **13**(13), 127–139 (1996)