Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Robotics and Autonomous Systems 45 (2003) 199–210

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Time optimal path planning considering acceleration limits

Marko Lepetič*, Gregor Klančar, Igor Škrjanc, Drago Matko, Boštjan Potočnik

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

Received 22 January 2003; received in revised form 15 August 2003; accepted 22 September 2003

Abstract

A robot path planning technique is proposed in the paper. It was developed for robots with differential drive, but with minor modifications it could be used for all types of nonholonomic robots. The path was planned in the way to minimise the time of reaching the end point in desired direction and with desired velocity, starting from the initial state described by the start point, initial direction and initial velocity. The limitation was the grip of the tires that results in the acceleration limits. The path is presented as a spline curve and was optimised by placing the control points through which the curve should pass.

Keywords: Mobile robots; Path planning; Acceleration limits; Spline curve; Velocity profile

1. Introduction

Mobile, autonomous robots are about to become an important element of the 'factory of the future' [15]. Their flexibility and their ability to react in different situations [12] open up totally new applications, leaving no limit to the imagination. To drive the mobile robot from its initial point to the target point, the robot must follow previously planned path. A well-planned path together with the robot capabilities assures the desired efficiency of the robot. The path could be optimised considering different aspects such as minimum time [16], minimum fuel, minimum length and others [5,9,10,13]. When the path is planned in details, the robot's capabilities are exactly known and that gives an advantage when co-ordinating several mobile robots [4,6].

This paper deals with time optimal path planning considering acceleration limits caused by limited friction force between the ground and the tires. The problem for which the solution is presented in this paper is the following: We want to find the path for the robot that would give the robot minimum time to move from the start point (SP) to the end point (EP) where the robot kicks the ball. Besides SP and EP, also the orientation and velocity in both points should be considered. The robot should stay inside its acceleration limits all the time. It could be said the paper presents an anti-skid path design. Similar problem was dealt by Wu et al. [16] where special consideration was given to different constraints in robot motion.

They proposed a technique and presented on the robot soccer system, which became very popular recently. It is an excellent test bed for various research interests such as path planning [5,9,10,13], obstacle avoidance [5], multi-agent co-operation [4,6,14], autonomous vehicles, game strategy [2,11], robotic vision [3,8], artificial intelligence and control. The robot soccer has also proven to be an excellent approach in engineering education, because it is attractive and through the game the students get immediate

* Corresponding author. Tel.: +386-1-476-87-02;
fax: +386-1-426-46-31.
E-mail address: marko.lepetic@fe.uni-lj.si (M. Lepetič).

0921-8890/$ – see front matter
doi:10.1016/j.robot.2003.09.007

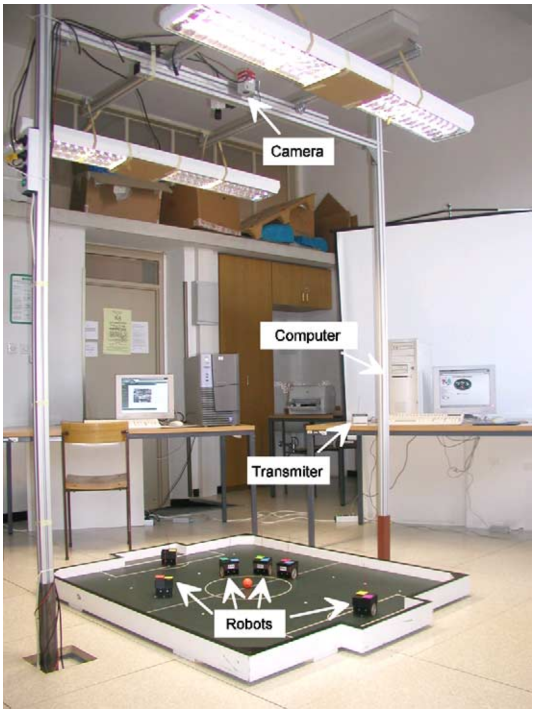

Fig. 1. The robot soccer system.

feedback about the quality of their algorithms. The system is shown in Fig. 1.

Mirosot is one of the games, for which the rules are provided by FIRA (Federation of International Robot-soccer Association). The robot size is limited by a cube of 7.5 cm side length. The navigation of the robots is provided by a vision system. The obtained positions of the robots and the ball are used for calculating the commands that are then sent to each robot radio transmitter. There are two leagues of Mirosot. The small league is a game of 3 against 3 robots on the playground of 1.5 m × 1.3 m, while 5 robots of each team play in the middle league on the playground sized 2.2 m × 1.8 m.

The paper is organised as follows: Section 2 presents the mathematical model of the robot and its limitations. A quick overview of curve synthesis and analysis is given in Section 3. Section 4 describes the proposed technique. A case study is presented in Section 5 and application aspects are discussed in Section 6. Section 7 gives the conclusions.

2. Robot model and limitations

The robot is of cubic shape with the side of 7.5 cm. It is driven with a differential drive, which is located at the geometric centre. This kind of drive allows zero turn-radius. The front and/or the back of the robot slide on the ground. For a more detailed description see Fig. 2. The commands that the computer sends to the robot are reference for linear and angular velocity. The microprocessor on the robot calculates the reference angular velocities of the left and right wheel. The motors that drive the wheels contain encoders so the microprocessor also knows the actual velocities. The PID controller in the microprocessor then calculates the needed voltage for both motors. The PID controller together with powerful motors causes sliding of the wheels if the desired velocity makes a step change. This knowledge is important when modelling the robot.

The movement of the robot can be modelled with the following equations:

$$\dot{x} = v_{\mathrm{real}} \cos(\varphi), \qquad \dot{y} = v_{\mathrm{real}} \sin(\varphi), \qquad \dot{\varphi} = \omega_{\mathrm{real}}, \tag{1}$$

where x, y and φ stand for position and orientation, respectively, v_{real} is the real linear velocity and ω_{real} the real angular velocity. If the wheels are not sliding, both velocities are very close to the reference velocities that have been sent to the robot. With these assumptions the real velocities from Eq. (1) can be substituted with the ones, which have been sent as commands.

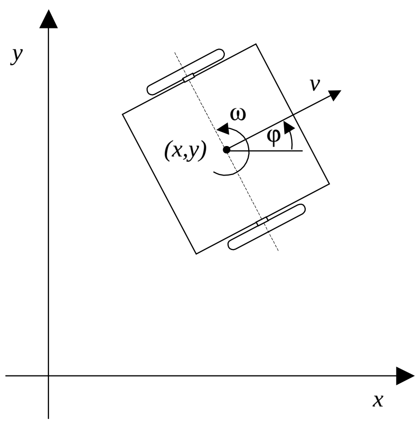

Fig. 2. The robot.

We get

$$\dot{x} = v\cos(\varphi), \qquad \dot{y} = v\sin(\varphi), \qquad \dot{\varphi} = \omega. \tag{2}$$

Only this simplified model will be used and all other dynamics will be neglected. It must not be forgotten, that this model is good only when the wheels do not slide. The force, which causes the acceleration of the robot, is the friction force. The size of friction force F_{friction} depends on the force that pushes object to the ground (the gravity) and the friction coefficient c_{friction}

$$F_{\text{friction}} = mgc_{\text{friction}}. \tag{3}$$

The friction force defines the maximal acceleration as

$$a_{\max} = \frac{F_{\text{friction}}}{m}. \tag{4}$$

With the combination of Eqs. (3) and (4), the following is obtained:

$$a_{\max} = \frac{F_{\text{friction}}}{m} = gc_{\text{friction}}. \tag{5}$$

From Eq. (5) it can be seen that only the friction coefficient limits the maximal acceleration. In case of mobile robot the force that wheels push to the ground is not the gravity. That happens, because the gravity centre of the robot is at a certain height above ground level. When accelerating in linear direction, the robot leans on the rear slider, which takes over a part of the robot weight. That means that the wheels of the robot press on the ground with a force that is smaller than gravity force. For that reason the limit of tangential acceleration differs from the limit of radial acceleration. Since the friction force is a product of the force orthogonal to the ground and the friction index. Comparing tangential acceleration to radial, the tangential acceleration is smaller.

The overall acceleration can be decomposed to tangential acceleration and radial acceleration. The tangential acceleration is the derivative of velocity with the respect to time and is caused with the intent to increase or decrease speed

$$a_{\text{tang}} = \frac{\mathrm{d}v}{\mathrm{d}t}. \tag{6}$$

The radial acceleration is caused by turning at certain speed and is the product of linear and angular velocity

$$a_{\text{rad}} = v \times \omega. \tag{7}$$

Since tangential and radial acceleration are orthogonal, the overall acceleration is the Pythagoras sum as follows:

$$a = \sqrt{a_{\text{tang}}^2 + a_{\text{rad}}^2}. \tag{8}$$

The overall acceleration is limited by the friction force. The acceleration limits have been measured in our case. To measure the radial acceleration limit, the angular velocity was set to a certain value and then the linear velocity was slowly increased. The slipping moment was determined visually. The maximal radial acceleration was then calculated from Eq. (7). Tangential acceleration limit measurement was a little more complicated. One of possibilities to measure it would be the experiment with an inclined plane. But we decided not to use it, because we wanted to get the measurement of maximal tangential acceleration in normal operating conditions, i.e. moving at normal speed. In this case slipping cannot be determined visually, so the computer vision system was used. Several experiments were made. During each experiment the robot was forced with a constant acceleration. The acceleration at each next experiment was slightly increased comparing to the previous experiment. Real acceleration of the robot was measured as second derivative of robot's position, which was obtained using the computer vision system. The measured maximum tangential acceleration was $2\,\text{m/s}^2$ and maximum radial acceleration $4\,\text{m/s}^2$, so the overall acceleration should be somewhere inside the ellipse as it is shown in Fig. 3.

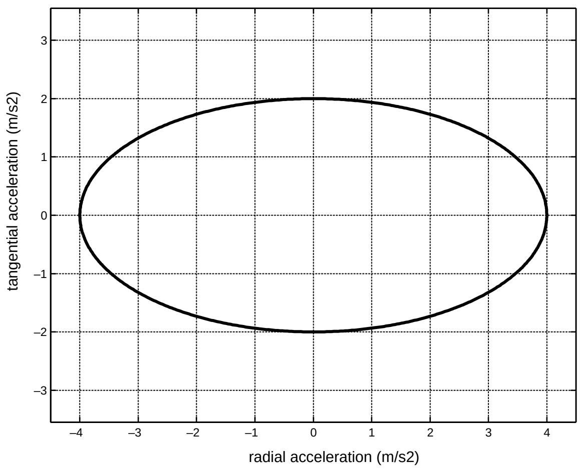

Fig. 3. Acceleration limits.

3. Curve design and analysis

There are many possible ways to describe the path. Spline curves are just one of them. The corresponding theory has been presented in a number of books and papers [1,7] so in this paper a quick overview will be given. The two-dimensional curve is got by combining two splines, $x(u)$ and $y(u)$, where u is the parameter along the curve. Each spline consists of one or more segments – polynomials. The point of tangency of two neighbour segments is called knot. The spline could be interpolated through desired points in the (u, x) or (u, y) domain, where also the derivative conditions can be fulfilled. When the knots are set, the spline parameters can be obtained by solving a linear equation system. If the pth order spline consists of m segments, then the number of parameters to determine is

$$m(p+1). \tag{9}$$

Number of linear equations is

$$n+(m-1)p, \tag{10}$$

where n is the number of explicitly defined points and derivative conditions at these points, $(m-1)$ the number of knots and p the number of continuous derivatives at the knots. The number of searched parameters should be equal to the number of linear equations, which leads to

$$m=n-p. \tag{11}$$

This equation presents the general spline condition, and if the constructor is not careful, some segments can be over- and others can be under-defined. To avoid this problem the knots were set to fit in the proposed interpolation points. These points are called control points (CP).

Fig. 4 shows the sample of set conditions to design the splines $x(u)$ and $y(u)$. Splines from Fig. 4 are joined to the curve $y(x)$ shown in Fig. 5. There are 7 conditions ($n = 7$) to define each of the splines and each of the splines consists of 4 segments ($m = 4$). According to Eq. (11) this leads to the cubic spline. A new inserted CP raises n and m for 1 and Eq. (11) remains fulfilled.

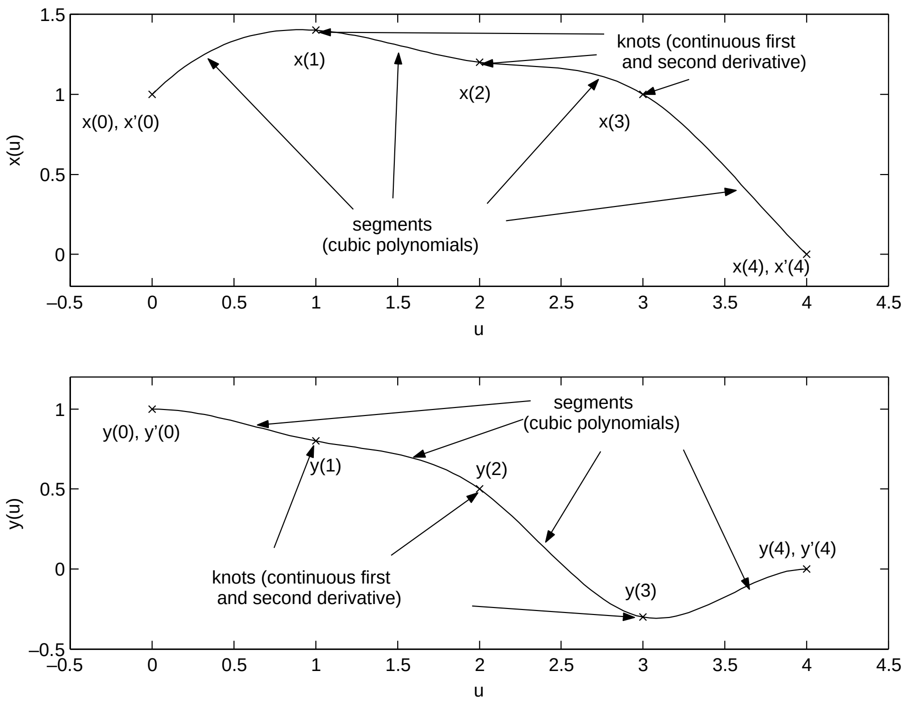

Fig. 4. The splines.

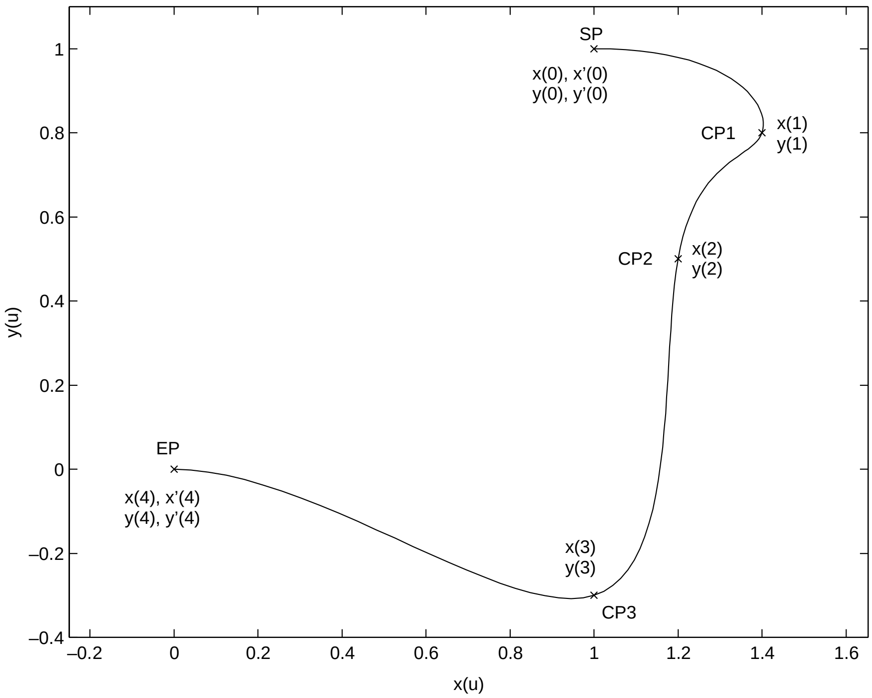

Fig. 5. The spline curve.

The orientation at the start and the end point (SP and EP) are given as angles, but should be transformed to the derivative conditions. The following can be written:

$$\varphi_{\mathrm{SP}} = \mathrm{arctg}\frac{y'(u_{\mathrm{SP}})}{x'(u_{\mathrm{SP}})}, \qquad \varphi_{\mathrm{EP}} = \mathrm{arctg}\frac{y'(u_{\mathrm{EP}})}{x'(u_{\mathrm{EP}})}, \quad (12)$$

where $x'(u_{\mathrm{SP}})$, $y'(u_{\mathrm{SP}})$, $x'(u_{\mathrm{EP}})$ and $y'(u_{\max\mathrm{EP}})$ are derivatives of splines $x(u)$ and $y(u)$ with the respect to parameter u at the start and the end point, and must be obtained knowing only the start and end direction. Eq. (12) determines only the quotients between derivatives $x'(u)$ and $y'(u)$. This leaves some free space to determine their absolute value. The idea is to determine them to stay at approximately the same value so the following was proposed:

$$\sqrt{x'(u_{\mathrm{SP}})^2 + y'(u_{\mathrm{SP}})^2} \approx \frac{\mathrm{dist}(\mathrm{SP}, \mathrm{first\,CP})}{u_{\mathrm{first\,CP}} - u_{\mathrm{SP}}},$$
$$\sqrt{x'(u_{\mathrm{EP}})^2 + y'(u_{\mathrm{EP}})^2} \approx \frac{\mathrm{dist}(\mathrm{last\,CP}, \mathrm{EP})}{u_{\mathrm{EP}} - u_{\mathrm{last\,CP}}}. \quad (13)$$

Time optimal path planning requires robots to drive with high speed. For driving with high speed a smooth path is necessary. The path smoothness is presented by the curvature κ. When dealing with spline curves in two dimensions κ is given as follows:

$$\kappa(u) = \frac{x'(u)y''(u) - y'(u)x''(u)}{(x'(u)^2 + y'(u)^2)^{3/2}}. \quad (14)$$

The geometrical meaning of the curvature is the inverted value of the circle radius at the particular point ($1/R$).

4. Finding the optimal path

In competition systems, such as robot soccer, the time needed by robots to get to desired points is most critical. So the problem to be solved is a minimum time problem where the time is calculated by integration of time differentials along the path

$$t = \int_{\mathrm{initial\ position}}^{\mathrm{target}} \frac{\mathrm{d}s}{v_{\mathrm{s}}(s)}, \quad (15)$$

where $v_s(s)$ is robot velocity as a function of s. Considering

$$ds = \sqrt{x'(u)^2 + y'(u)^2}\,du \tag{16}$$

and substituting $v_s(s)$ by $v(u)$, which is actually the same function running on different parameter, Eq. (15) can be written as

$$t = \int_{u_{SP}}^{u_{EP}} \frac{\sqrt{x'(u)^2 + y'(u)^2}}{v(u)}\,du. \tag{17}$$

To assure the real robot to follow the prescribed path, it must not slide, i.e. his accelerations must be within limits given in Fig. 3. It is well known that the time optimal systems operate on their limits, so the acceleration must be within the ellipse given in Fig. 3. The problem is solved by constrained numerical optimisation with control points as free parameters to be optimised. The optimisation procedure is as follows:

(1) Choose initial control points randomly and calculate the initial path. An example of this is shown in Fig. 6.
(2) For a given path the highest allowable overall velocity profile is calculated as follows:

- Its curvature is calculated according to Eq. (14) as shown in Fig. 7.
- The local extrema (local maxima of absolute value) of the curvature are determined and named turning points (TP). In these points the turning radius reaches local minimum. That means the velocity in this point should be locally the lowest. Maximum allowable speed of the robot at the TP is determined according radial acceleration limit. The tangential acceleration at that point is supposed to be 0. That is actually always true because there is a local minimum of turning radius.
- Before and after a TP, the robot can move faster, because the curve radius is bigger than at the TP. Before and after the TP the robot must tangentially decelerate and accelerate, respectively, as maximally allowed by the (de) acceleration constraint. In this way the maximum velocity profile is determined for each TP and has the shape of a ‘U’ (or ‘V’) as shown in Fig. 8. Minima of velocity profiles at Fig. 8 corresponds with the TP seen in Fig. 7. At some point the velocity profile becomes horizontal. The velocity there is so high, that

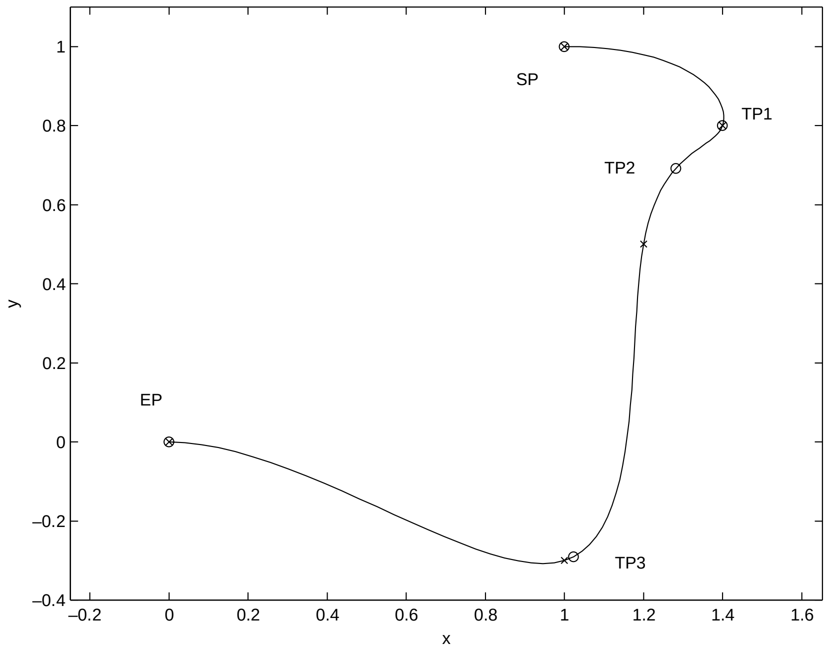

Fig. 6. Example of the curve.

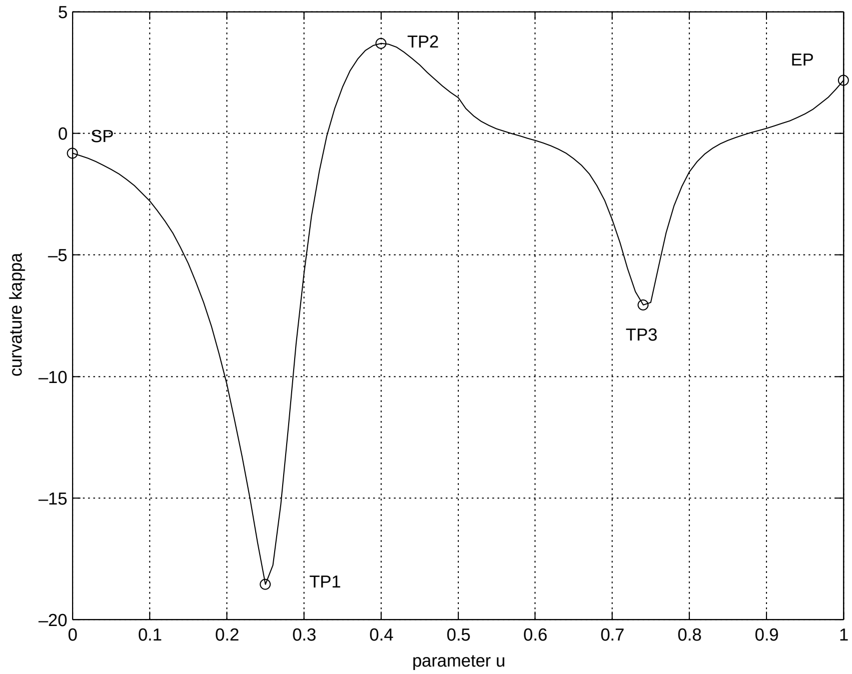

Fig. 7. The curvature.

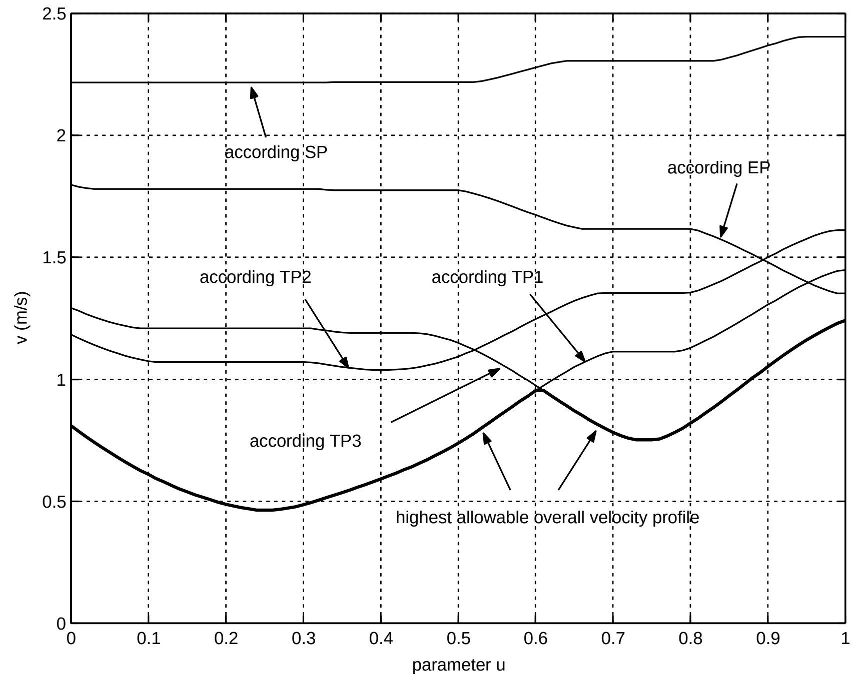

Fig. 8. The velocity profile.

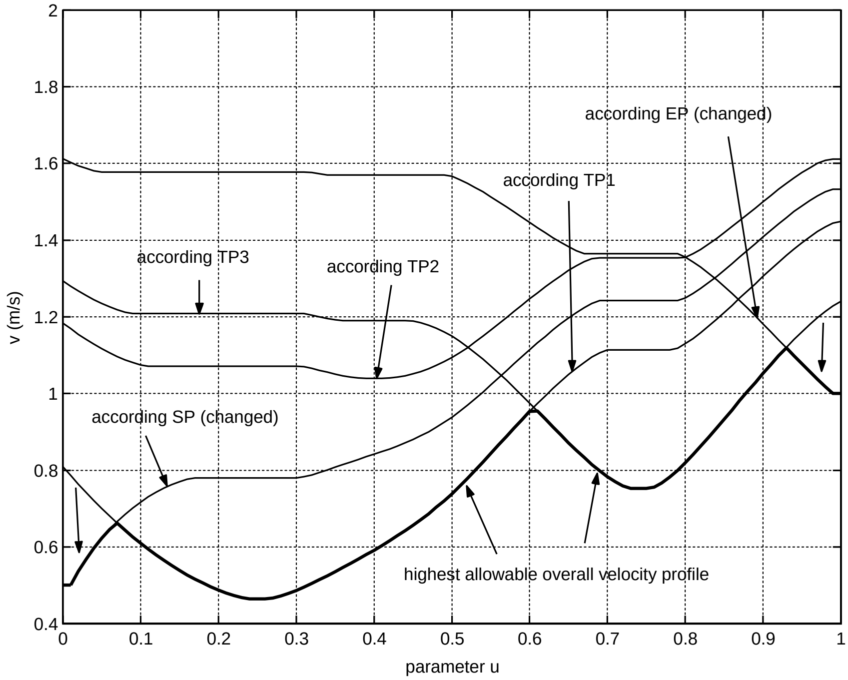

Fig. 9. The corrected velocity profile concerning initial and terminal velocity.

the radial acceleration is out of limits. The part of the curve after that point is useless. This happens because the curvature starts increasing (the influence of the neighbour TP). But that neighbour TP requires lower speed in that area so its lower limit must be observed.

- Similarly the maximum velocity profile (due to tangential acceleration/deceleration) is determined for the initial (SP) and final (FP) velocity, respectively.
- The highest allowable overall velocity profile is determined as the minimum of all velocity profiles, as indicated in Figs. 8 and 9 by bold curves.
- The initial and final velocities must be on the highest allowable overall velocity profile (as it is in Fig. 9). If not, the given path cannot be driven without violating acceleration constraints (this is Fig. 8).
- For given highest allowable velocity profile the cost function (time) is calculated according to Eq. (17).

(3) Optimise the problem with control points as optimising parameters using one of optimisation methods and time needed as a cost function. This is described in detail in Section 6.

5. Case study

The objective of this case study is to determine the number of points needed to find a good approximation of the time optimal path. Let us take a look at a case that is not very simple, but on the other hand it is not the most complicated. The robot starts at the point SP $(-0.5, 1)$ in direction 225° with the velocity of 1 m/s. The end point is at the origin of the system. The robot should pass it with the velocity of 1 m/s in the direction 180°. The question is how many control points are needed. Two points are needed to fulfil the conditions of initial and terminal velocity. Each one can be placed in the way to ensure some minimum distance from start or end point to the closer TP. The test was made with a varying number of CPs. The initial

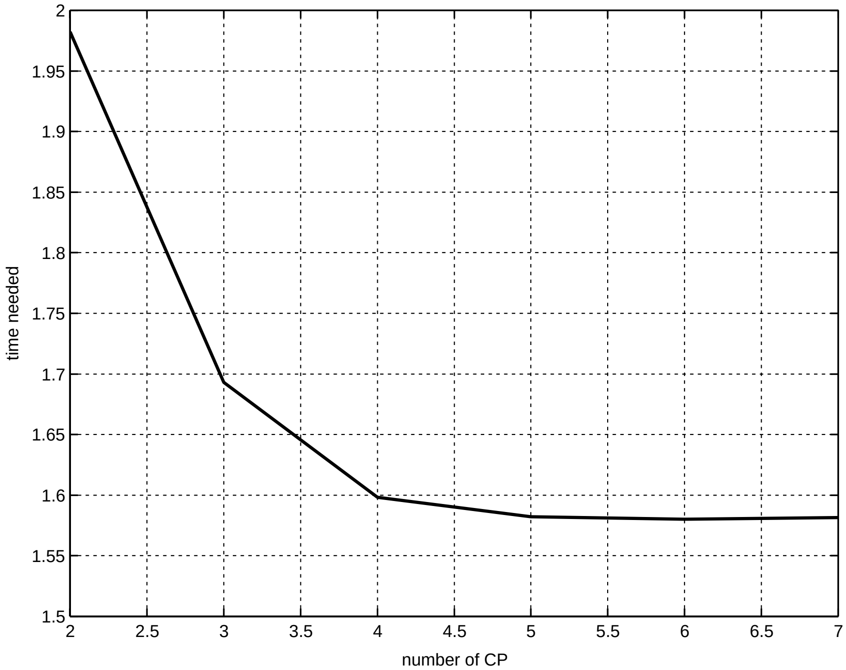

Fig. 10. The needed time with the respect to the number of CP.

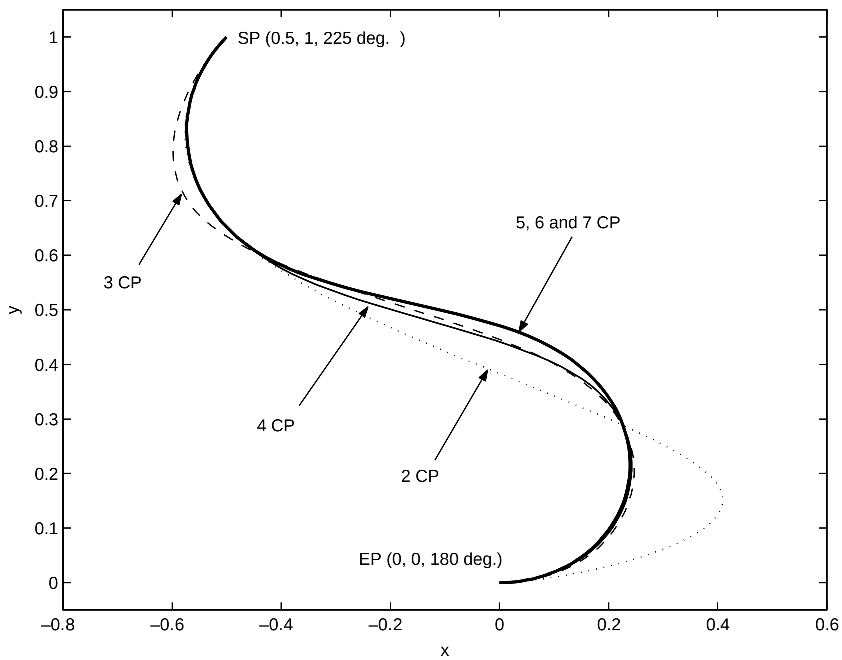

Fig. 11. Paths with different number of CPs.

number was 2 and was increased up to 7 CPs. Fig. 10 shows how the needed time depends on the number of CPs. It can be seen that the use of 4 CPs are optimum in our case. The 4th CP improves the time for a tenth of a second (more than 6%) and the 5th would improve it for only one-hundredth of a second. Also very important is the choice of optimisation method. In the presented example, good results were achieved using the Nelder–Mead simplex (direct search) method.

The resulting paths are shown in Fig. 11. The doted line presents a 2 CP path, the 3 CP path is shown with a dashed line and the 4 CP path with a continuous line. Five, 6 or 7 CP paths are practically the same and are presented with the thick line. Choosing the number of CPs used is problem oriented. As a matter of fact it depends on performance requirements and computation capabilities. In our case 4 CPs were sufficient. It can be seen where the 2 and 3 CP paths spend too much time because of suboptimal path. The 5 (or more) CP path is only slightly different from the 4 CP one, and the difference lies in the area where a large improvement cannot be obtained.

In some cases there would be more than 4 CPs needed to find a path close to optimal. But the problem of using only 4 CPs is not critical. In case of not using enough CPs the result is not so close to optimality (time needed would increase). If the number of playing robots is taken into account, we can say that the robot with such a complicated path would also need more time to reach the goal. The goal is usually to kick the ball and that is a task just for one robot. The supervisory algorithm, which controls the roles of the robots, would choose the robot with minimum time needed to do that and would probably not choose the robot with a complicated path.

6. Optimising the placement of control points

The proposed technique uses optimisation to find an optimal solution. As it is well known, optimisation is very time-consuming. The particular problem becomes burning when the realisation is taken into account. The robot's control algorithm acts in the following way. First the path is planned and then the control action is calculated from a planned path using the inverted model of the robot. This is repeated each time instant. The time allocated to the path planning is therefore shorter than the sample time. In a dynamically changing environment, as the robot soccer game is, a short sample time is required. Actually the camera defines it. Using a NTSC standard camera the sample time is 33 ms, and this is a far shorter time than time needed for optimisation. The idea that solves this problem is called multi-parametric programming. For a grid of initial relative positions of the robot regarding to the ball, the paths (CPs) are obtained in advance and are stored in a look-up table. Inputs are relative robot position, initial angle, initial and final velocity and outputs are the CPs. The table is determined for a certain quantisation. For the intermediate points, linear interpolation is used.

Moving ball is a problem that is not dealt in this paper. Generally, it could be solved in the following way. The presumption is that the ball is moving in the straight line, so depending on time the ball position is defined. First two points should be chosen on the ball track line and time optimal path to the both points should be determined. For both points it is known, the time when the ball reaches that particular point and the time when the robot reaches it. One point should be close to the present ball position so the ball reaches it before the robot and the other point should be far enough that the robot is there before the ball. The object is to find the point where the robot (driving time optimal) and the ball meet simultaneously. The bisection method solves mentioned problem.

The use of a look-up table also increases co-operating capabilities. Robots can very quickly determine which of them needs shorter time to perform an action. Shorter time if often closely related to the effectiveness. Such precision path planning offers a lot of support to the multi-agent decision-making algorithm that is in charge for robot co-operation.

7. Conclusions

A path finding algorithm for nonholonomic mobile robots was proposed. The case study concerned slippery conditions in robot soccer environment. The path is presented as a spline curve and was obtained with control points positioning. The control points were placed using an optimisation function where the criterion was time needed. The optimisation is a very time-consuming process and cannot be done online, so a look-up table was built. Due to well-defined movement of all robots, the co-operation between players also improved.

References

[1] The MathWorks Inc., Spline Toolbox User's Guide, Version 2, 1999.
[2] M. Asada, E. Uchibe, K. Hosoda, Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via vision-based reinforcement learning and development, Artificial Intelligence 110 (1999) 275–292.
[3] P. Borenszejn, J. Jacobo, M. Mejail, A. Stoliar, A. Katz, M. Cecowski, A. Ferrari, J. Santos, Design of the vision system for the UBA-Sot team, in: Proceedings of the 2002 FIRA Robot World Congress, vol. 1, Seoul, May 26–29, 2002, pp. 616–619.
[4] C. Candea, H. Hu, L. Iocchi, D. Nardi, M. Piaggio, Coordination in multi-agent RoboCup teams, Robotics and Autonomous Systems 36 (2001) 67–86.
[5] G. Desaulniers, On shortest paths for a car-like robot maneuvering around obstacles, Robotics and Autonomous Systems 17 (1996) 139–148.
[6] M. Egerstedt, X. Hu, A hybrid control approach to action coordination for mobile robots, Automatica 38 (2002) 125–130.
[7] A.I. Ginnis, P.D. Kaklis, Planar C^2 cubic spline interpolation under geometric boundary conditions, Computer Aided Geometric Design 19 (5) (2002) 345–363.
[8] G. Klančar, O. Orqueda, D. Matko, R. Karba, Robust and efficient vision system for mobile robots control—application to soccer robots, Electrotechnical Review, Journal for Electrical Engineering and Computer Science 68 (5) (2001) 306–312.
[9] F.M. Marchese, A directional diffusion algorithm on cellular automata for robot path-planning, Future Generation Computer Systems 18 (2002) 983–994.
[10] C.F. Martin, S. Sun, M. Egerstedt, Optimal control, statistics and path planning, Mathematical and Computer Modelling 33 (2001) 237–253.
[11] D. Matko, G. Klančar, M. Lepetič, A tool for the analysis of robot soccer game, in: Proceedings of the 2002 FIRA Robot World Congress, vol. 1, Seoul, May 26–29, 2002, pp. 743–748.
[12] L. Podsedkowski, J. Nowakowski, M. Idzikowski, I. Vizvary, A new solution for path planning in partially known or unknown environment for nonholonomic mobile robots, Robotics and Autonomous Systems 34 (2001) 145–152.
[13] I. Škrjanc, G. Klančar, M. Lepetič, Modeling and simulation of prediction kick in Robo-Football, in: Proceedings of the 2002 FIRA Robot World Congress, vol. 1, Seoul, May 26–29, 2002, pp. 616–619.
[14] P. Švestka, M.H. Overmars, Coordinated path planning for multiple robots, Robotics and Autonomous Systems 23 (1998) 125–152.
[15] Y. Ting, W.I. Lei, H.C. Jar, A path planning algorithm for industrial robots, Computers and Industrial Engineering 42 (2002) 299–308.
[16] W. Wu, H. Chen, P.Y. Woo, Time optimal path planning for a wheeled mobile robot, Journal of Robotic Systems 17 (11) (2000) 585–591.

Marko Lepetič in 2000 finished the study and got B.Sc. degree at the Faculty of Electrical Engineering, University of Ljubljana. Currently, he is employed at the same faculty as Ph.D. student. At the beginning his research was oriented in nonlinear predictive controllers. Later his research interests were in the control of multi-agent systems, especially application oriented.

Gregor Klančar received the B.Sc. degree in 1999 from the Faculty of Electrical Engineering, University of Ljubljana, Slovenia. His current research interests are in the area of fault diagnosis methods of technical processes and multiple vehicle co-ordinated control.

Igor Škrjanc received the B.Sc., the M.Sc. and the Ph.D. degrees in electrical engineering, in 1988, 1991 and 1996, respectively, from the Faculty of Electrical and Computer Engineering, University of Ljubljana, Slovenia. He is currently an Assistant Professor in the same faculty. His main research interests are in adaptive, predictive, fuzzy and fuzzy adaptive control systems what was also the title of his Ph.D. thesis.

Drago Matko received the B.Sc., the M.Sc. and the Ph.D. degrees in electrical engineering in 1971, 1973 and 1977, respectively, from the University of Ljubljana, Slovenia, for work in the field of adaptive control systems. He visited the Institute for Control Engineering Darmstadt, Germany, several times, from 1980 to 1982, 1984, 1985 and 1986 as a Humboldt fellow, from 1987 to 1991 in the frame of the project adaptive control systems sponsored by International Bureau KFA Jülich. In 1995–1996, he visited the Institute of Space and Astronautical Science in Sagamihara Kanagawa, Japan for 9 months as Foreign Research Fellow. He has published 24 journal articles, more than 200 conference papers and four student edition books (in Slovene) and he is also a co-author of two books published by Prentice-Hall. In 1989, he received the award of Slovenian ministry for research and technology for the work in the field of computer aided design of control systems.

Boštjan Potočnik received the B.Sc. and M. Sc. degrees in 1998 and 2001, respectively, from the Faculty of Electrical Engineering, University of Ljubljana. Currently, he work in the field of hybrid systems with a stress on the modelling and control.