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
marko.lepetic@fe.uni-lj.si

***Abstract*--Path planning technique is proposed in the paper. It was developed for robots with differential drive, but with minor modification could be used for all types of nonholonomic robots. The path was planned in the way to minimize the time of reaching 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 the spline curve and was optimised by placing the control points trough which the curve should take place.**

Index Terms: **mobile robots, path planning, acceleration limits, spline curve, velocity profile**

I. Introduction

Mobile, autonomous robots are about to become an important element of the "factory of the future" [12]. Their flexibility and their ability to react in different situations [9] 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. Well-planned path together with robot capabilities assure desired efficiency of the robot. The path could be optimised considering different aspects such as minimum time, minimum fuel, minimum length and others [4, 7, 10]. When the path is planned in details, the robot capabilities are exactly known and that makes an advantage when coordinating several mobile robots [3].

This paper deals with time optimal path planning considering acceleration limits. The proposed technique is 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 [4, 7, 10], obstacle avoidance [4], multi-agent cooperation [3, 11], autonomous vehicles, game strategy [2, 8], robotic vision [6], artificial intelligence and control. The robot soccer has also proven to be excellent approach in engineering education, because it is attractive and through the game the students get immediate feedback about the quality of their algorithms.

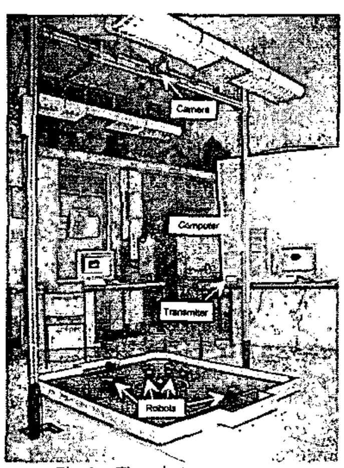

Fig. 1 - The robot-soccer system

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 with the cube of 7.5 cm side length. The navigation of the robots is provided with the 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. Small league is a game of 3 against 3 robots on the playground of 1.5 m x 1.3 m, while 5 robots of each team play middle league on the playground sized 2.2 m x 1.8 m.

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.

The paper is organized 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. Case study is presented in Section 5 and application aspects are discussed in Section 6. Section 7 gives the conclusions.

II. Robot Model and Limitations

The robot is cubic shape with the side of 7.5 cm. It is driven with the 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 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

0-7803-7763-X/03/$17.00 ©2003 IEEE

microprocessor also knows 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 step change. This knowledge is important when modelling the robot.

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

$$\begin{aligned} \dot{x} &= v_{real}\cos(\varphi) \\ \dot{y} &= v_{real}\sin(\varphi) \\ \dot{\varphi} &= \omega_{real} \end{aligned} \qquad (1)$$

where x, y and φ stand for position and orientation respectively, v_{real} is real linear velocity and ω_{real} is 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 has been sent as commands. We get:

$$\begin{aligned} \dot{x} &= v\cos(\varphi) \\ \dot{y} &= v\sin(\varphi) \\ \dot{\varphi} &= \omega \end{aligned} \qquad (2)$$

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 don't slide or in other words, when the robot is not forced with too large acceleration. 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 desire to increase or decrease speed.

$$a_{tang} = \frac{dv}{dt} \qquad (3)$$

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

$$a_{rad} = v \times \omega \qquad (4)$$

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

$$a = \sqrt{a_{tang}^2 + a_{rad}^2} \qquad (5)$$

The overall acceleration is limited with the friction force.

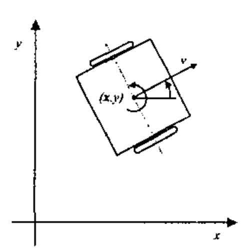

Fig. 2 - The robot

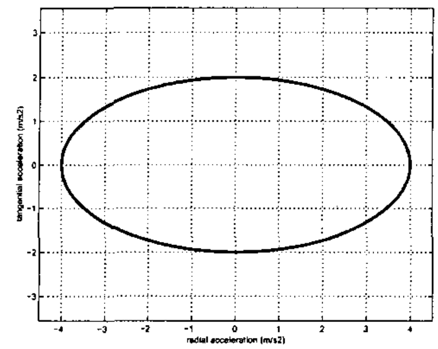

Fig. 3 - Acceleration limits

The limit of tangential acceleration differs from the limit of radial acceleration. That happens, because the gravity centre of the robot is on 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 the force that is smaller than gravity force. We know that the friction force is product of the force orthogonal to the ground and the friction index. Comparing tangential acceleration to the radial, the orthogonal force is smaller, which causes lower acceleration limit.

The acceleration limits have been measured in our case. To measure 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. (4). Tangential acceleration limit measurement was little more complicated. In this case slipping cannot be determined visually, so the vision system was used. Several experiments were made. During each experiment the robot was forced with the 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 derivation of robot's position, which was obtained using the vision system. Measured maximum tangential acceleration was 2 m/s^2 and maximum radial acceleration 4 m/s^2, so the overall acceleration should be somewhere inside the ellipse as it is shown on Fig. 3.

III. 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 number of books and papers [1,5] 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 (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

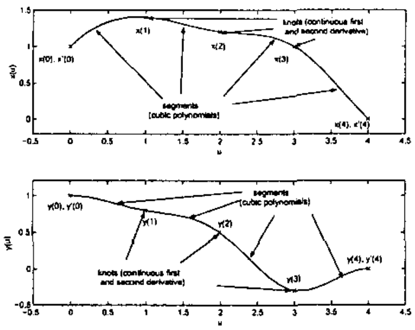

Fig. 4 - The splines

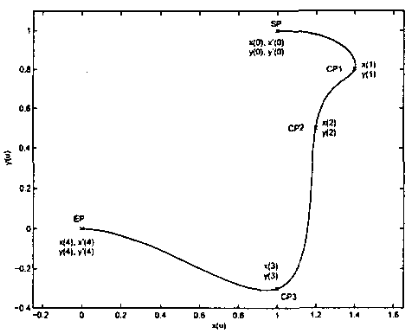

Fig 5 - The spline curve

solving a linear equation system. If the p-th order spline consists of m segments, than the number of parameters to determine is

$$m(p+1) \tag{6}$$

Number of linear equations is

$$n+(m-1)p \tag{7}$$

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

$$m=n-p \tag{8}$$

This equation presents the general spline condition, and if the constructor is not careful, some segments can be over- and other 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 joint to the curve $y(x)$ that is shown in Fig 5. There are 7 conditions (n=7) to define each of splines and each of splines consists of 4 segments (m=4). According to eq. (8) this leads to the cubic spline. New inserted CP raises n and m for 1 and eq.(8) remains fulfilled.

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

$$\varphi_{SP}=arctg\frac{y'(u_{\min})}{x'(u_{\min})} \quad , \quad \varphi_{EP}=arctg\frac{y'(u_{\max})}{x'(u_{\max})} \tag{9}$$

where $x'(u_{\min})$, $y'(u_{\min})$, $x'(u_{\max})$ and $y'(u_{\max})$ 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. This leaves some free space, so the following was proposed:

$$\begin{aligned}\sqrt{x'(u_{SP})^2+y'(u_{SP})^2} &\approx \frac{dist(SP, first\,CP)}{u_{firstCP}-u_{SP}}\\ \sqrt{x'(u_{EP})^2+y'(u_{EP})^2} &\approx \frac{dist(last\,CP, EP)}{u_{EP}-u_{lastCP}}\end{aligned} \tag{10}$$

Time optimal path planning requires robots to drive with high speed. For driving with high speed 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:

$$\kappa(u)=\frac{x'(u)y''(u)-y'(u)x''(u)}{\left(x'(u)^2+y'(u)^2\right)^{3/2}} \tag{11}$$

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

IV. 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

$$t=\int_{init.pos.}^{target}\frac{ds}{v} \tag{12}$$

Considering

$$ds=\sqrt{x'(u)^2+y'(u)^2}\,du\,, \tag{13}$$

Eq. (12) can be written as

$$t=\int_a^b\frac{\sqrt{x'(u)^2+y'(u)^2}}{v(u)}du \tag{14}$$

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 on the ellipse given in Fig. 3. The problem is solved by constraint numerical optimisation with control points as free parameters to be optimised. The optimisation procedure is as follows:

1. Choose initial control points and calculate the initial path. An example of this is shown in Fig.4.
2. For given path the highest allowable overall velocity profile is calculated as follows:
   - Its curvature is calculated according to Eq. (11) as shown in Fig 6.
   - The local extreme (local maximum of absolute value) of the curvature are determined and named turning points (TP). In these points the robot has to move with maximum allowable speed due to radial acceleration limit. Its tangential acceleration must be 0.
   - Before and after a TP, the robot can move faster, because the curve radius get bigger than in TP. Before and after the TP the robot can tangentially decelerate and accelerate respectively as max. allowed by (de) acceleration constraint. In this way the maximum velocity profile is determined for each TP and have the shape of "U" (or "V") as shown in Fig. 7. At some point the velocity profile becomes horizontal. The velocity there is so high, that 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 the described problem doesn't really have meaning.
   - Similarly the maximum velocity profile (due to tangential acceleration/deceleration) is determined for initial (SP) and final (FP) (if required) velocity respectively.
   - The highest allowable overall velocity profile is determined as the minimum of all velocity profiles, as indicated in Figs. 7 and 8 (bold curves)
   - The initial and final (if required) velocities must be on the highest allowable overall velocity profile (as it is in Fig 8). If not, the given path cannot be driven without violating acceleration constraints. (The case in Fig 7).
   - For given highest allowable velocity profile the cost function is calculated according Eq.(14).
3. Optimize the problem with control points as optimizing parameters.

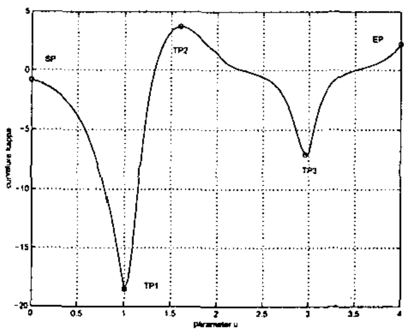

Fig. 6 - The curvature

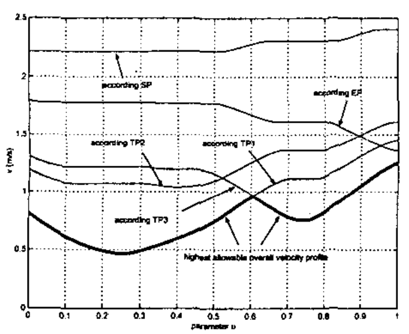

Fig. 7 - The velocity profile

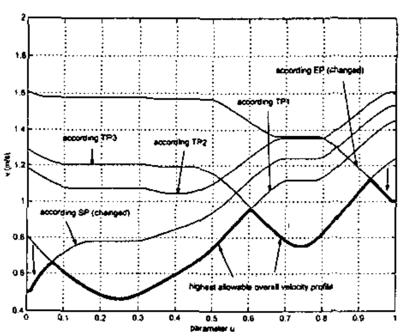

Fig. 8 - The corrected velocity profile concerning initial and terminal velocity

V. Case Study

The objective of this case study is to the number of points needed to find good approximation of time optimal path. Let us take a look to the case for which we can say it is not very simple, but on the other hand we cannot say it is 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 in 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 the various number of CPs. The initial number was 2 and was increased up to 7 CPs. Fig. 9 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 4^{th} CP improves the time for a tenth of a second (more than 6 %) and the 5^{th} would improve it for only one hundredth of a second.

The resulted paths are shown in Fig. 10. The doted line presents 2 CP path, 3 CP path is shown with dashed line and 4 CP path with continuous line. 5, 6 or 7 CP paths are practically the same and are presented with the thick line. It can be seen where 2 and 3 CP paths spend too much time because of not well-defined path. 5 (or more) CP path is slightly different from the 4 CP one and the difference lies in the area where a large improvement cannot be done.

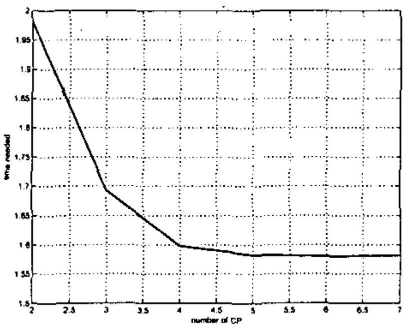

Fig. 9 - The needed time with the respect to the number of CP

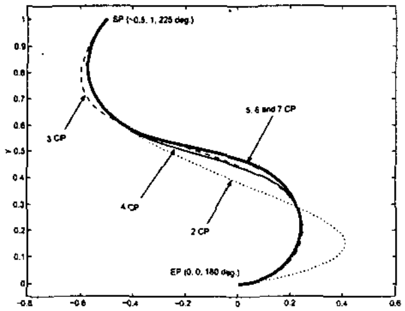

Fig. 10 - Paths with different number of CP

In some cases there would be more than 4 CPs needed to find 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 optimal (time needed would increase). If the number of playing robots is taken into account, we can say that the robot with such complicated path would also need more time to reach the goal. The goal is usually to kick the ball and that is job just for one robot. The supervisory algorithm who 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 complicated path.

VI. The Application Aspect

The proposed technique uses optimisation to find optimal solution. As it is well known, the 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, then the control action is calculated from 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 sample time. In the dynamically changing environment, like it is robot soccer game, short sample time is required. Actually it is defined with the camera. Using the NTSC standard camera the sample time is 33 ms, and this is 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 position of the robot regarding to the ball, the paths (CPs) are obtained in advance and are stored to the look-up table. Inputs are relative robot position, initial angle, initial and final velocity and outputs were the CPs. The table was determined for certain quantization. For the intermediate points, linear interpolation was used.

The use of look-up table also increases cooperating 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 cooperation.

VII. Conclusions

The 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 got with the control points positioning. The control points were placed using the optimisation function where the criterion was needed time. The optimisation is very time-consuming process and cannot be done online, so the look-up table was built. Due to well defined future moving of all robots, the cooperation between players also improved.

References

[1] The MathWorks Inc., *Spline Toolbox User's Guide*, Version 2, 1999.

[2] Asada, M., Uchibe, E., Hosoda, K., Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via vision-based reinforcement learning and development. *Artificial Intelligence*, Vol. 110, pp. 275-292, 1999.

[3] Candea, C., Hu, H., Iocchi, L., Nardi, D., Piaggio, M., Coordination in multi-agent RoboCup teams. *Robotics and Autonomous Systems*, Vol.36, pp. 67–86, 2001.

[4] Desaulniers, G., On Shortest paths for a car-like robot maneuvering around obstacles. *Robotics and Autonomous Systems*, Vol. 17, pp. 139–148, 1996.

[5] Ginnis, A. I., Kaklis, P. D., Planar C^2 cubic spline interpolation under geometric boundary conditions, *Computer Aided Geometric Design*, Vol. 19, Iss. 5, pp. 345-363, 2002.

[6] Klančar, G., Orqueda, O., Matko, D., Karba, R., Robust and efficient vision system for mobile robots control-application to soccer robots. *Electrotechnical Review, Journal for Electrical Engineering and Computer Science*, Vol 68, No 5, pp 306-312, 2001.

[7] Martin, C. F., Sun, S., Egerstedt, M., Optimal Control, Statistics and Path Planning. *Mathematical and Computer Modelling*, Vol. 33, pp. 237-253, 2001.

[8] Matko, D., Klančar, G. Lepetič, M., A Tool For the Analysis of Robot Soccer Game, *Proceedings of the 2002 FIRA Robot World Congress*, Vol. 1, pp. 743-748, Seoul, May 26-29 2002.

[9] Podsedkowski, L., Nowakowski, J., Idzikowski, M., Vizvary, I., A new solution for path planning in partially known or unknown environment for nonholonomic mobile robots, *Robotics and Autonomous Systems*, Vol. 34, pp. 145–152, 2001.

[10] Škrjanc, I., Klančar, G., Lepetič, M., Modeling and Simulation of Prediction Kick in Robo-Football, *Proceedings of the 2002 FIRA Robot World Congress*, Vol. 1, pp. 616-619, Seoul, May 26-29 2002.

[11] Švestka, P., Overmars, M. H., Coordinated path planning for multiple robots. *Robotics and Autonomous Systems*, Vol. 23, pp. 125-152, 1998.

[12] Ting, Y., Lei, W. I., Jar, H. C., A Path Planning Algorithm for Industrial Robots. *Computers & Industrial Engineering*, Vol. 42, pp. 299-308, 2002