

*Chapter 10*

## **PATH PLANNING AND PATH TRACKING FOR NONHOLONOMIC ROBOTS**

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

Faculty of Electrical Engineering, University of Ljubljana,  
Tržaška 25, SI-1000 Ljubljana, Slovenia  
Tel +386 (1) 476 84 17, Fax +386 (1) 426 46 31

### **Abstract**

Chapter deals with most common problems of mobile robotic system. Those are path planning and path tracking. Path planning technique that is proposed in the chapter is based on a spline curve design. It was developed for nonholonomic robots with differential drive, but with minor modification could be used for all types of nonholonomic robots. The path is planned in the way to minimize the time needed for 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 is the grip of the tires that results in the acceleration limits. Further on a state space controller is proposed for tracking planned paths and its stability is proven by Lyapunov theorem.

**Keywords:** mobile robots, path planning, acceleration limits, spline curve, velocity profile, path tracking

## **1 Introduction**

Mobile, autonomous robots are about to become an important element of the “factory of the future” [19]. Their flexibility and their ability to react in different situations [15] 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-

---

\* Email address: marko.lepetic@fe.uni-lj.si

planned path together with robot capabilities assure desired efficiency of the robot. The path could be optimized considering different aspects such as minimum time, minimum fuel, minimum length and others [6, 10, 11, 17]. When the path is planned in details, the robot capabilities are exactly known and that makes an advantage when coordinating several mobile robots [4, 7]. Also high performance path tracking is essential. Efficiency is higher when the robot can drive faster and at that time stays more accurate on the planned path.

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 [6, 10, 11, 17], obstacle avoidance [6], multi-agent cooperation [4, 7, 18], autonomous vehicles, game strategy [1, 12], robotic vision [3, 8], 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.

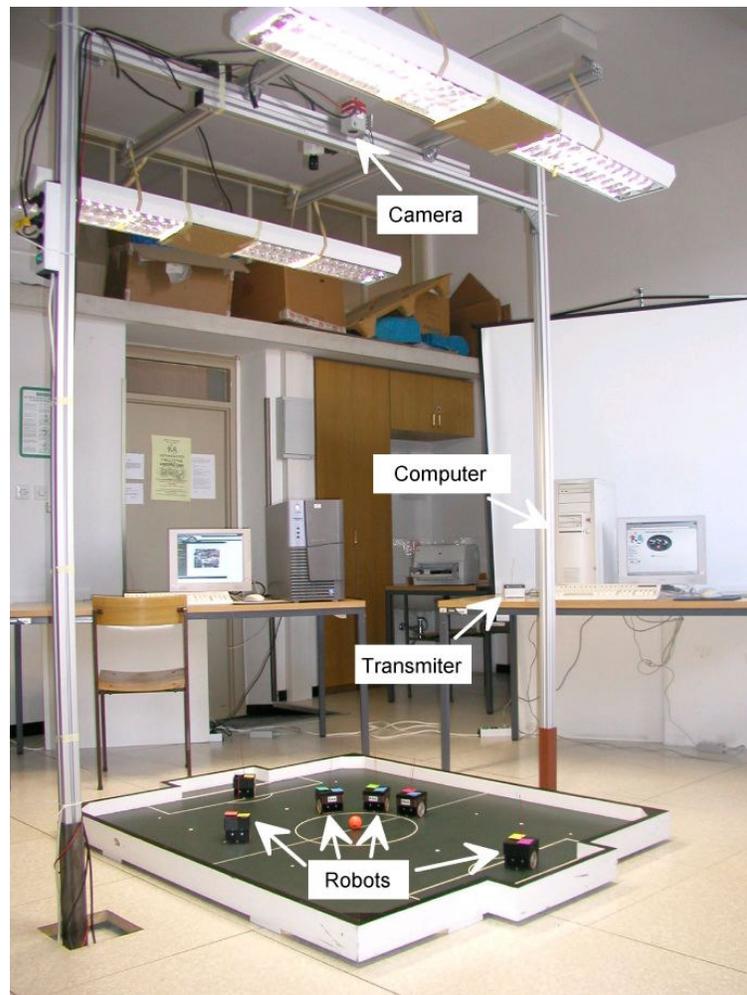


Fig. 1 - The robot-soccer system

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 that the paper presents an anti-skid path design. After that a controller with feed forward is presented.

## 2 Nonholonomic Systems

Holonomic and nonholonomic constraints are dealing with the kinematical model. Let us imagine a system with  $n$  generalized coordinates  $\mathbf{q} = [q_1, \dots, q_n]^T$ . We have to deal with holonomic constraints when the following can be written:

$$F(\mathbf{q}) = F(q_1, \dots, q_n) = 0 \quad (1)$$

Equation express dependence at least one of coordinates from the other and decreases number of the degrees of freedom of the system. If the equation (1) include also the time derivatives of the generalized coordinates we get:

$$F(\mathbf{q}, \dot{\mathbf{q}}) = F(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) = 0 \quad (2)$$

Constraints of this type are called nonholonomic constraints and the system is called nonholonomic system. Equation (2) comparing to (1) does not mean reciprocal dependencies, but expresses dependencies of coordinates derivatives from the coordinates.

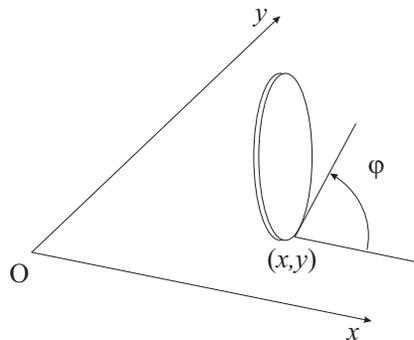


Fig. 2 - Rolling wheel

We have for example a wheel that rolls without sliding. The rolling wheel is chosen because it is very simple to understand, but its kinematical model can be generalized to a lot of types of mobile robots or even cars. It is shown in Fig. 2. Its kinematical state can be presented with its position in cartesian system of coordinates  $x$  and  $y$  and its orientation  $\varphi$ . For the rolling wheel it can be written:

$$\dot{x} = \omega r \cos(\varphi) \quad (3)$$

$$\dot{y} = \omega r \sin(\varphi) \quad (4)$$

where  $\omega$  is angular speed and  $r$  is radius of the wheel. Furthermore, from both equations the product  $\omega r$  can be expressed and equalized. The result is:

$$\dot{x} \sin(\varphi) - \dot{y} \cos(\varphi) = 0 \quad (5)$$

Equation (5) expresses dependence between coordinates and its derivatives in the sense of equation (2). Therefore the rolling wheel is nonholonomic system. Nonholonomic constraint itself does not restrict the system in reaching any desired point. But the nonholonomic system should follow the right sequence of operations to reach the goal. To control such systems it is necessary to plan a path that makes reaching the desired goal possible. Generally this is a curve that fulfils certain conditions like minimum turning radius, minimum distance to the obstacles for example.

### 3 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 center. 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. 3. 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 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 modeling the robot.

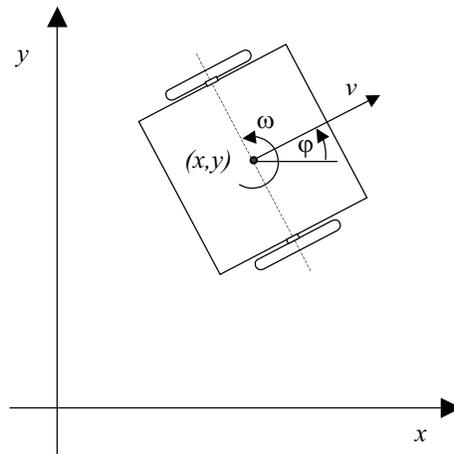


Fig. 3 - The robot

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

$$\begin{aligned}\dot{x} &= v_{real} \cos(\varphi) \\ \dot{y} &= v_{real} \sin(\varphi) \\ \dot{\varphi} &= \omega_{real}\end{aligned}\quad (6)$$

where  $x$ ,  $y$  and  $\varphi$  stand for position and orientation respectively,  $v_{real}$  is real linear velocity and  $\omega_{real}$  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. (6) can be substituted with the ones, which has been sent as commands. We get:

$$\begin{aligned}\dot{x} &= v \cos(\varphi) \\ \dot{y} &= v \sin(\varphi) \\ \dot{\varphi} &= \omega\end{aligned}\quad (7)$$

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. This model will be used for path planning and also for the control. For more detailed look into some situations that happen when the robot is driving along reference path we shall write more detailed model. At this point the robot is considered as a rigid body that is moving on flat surface. Let there be a cartesian system pinned fastened to the robot. Its  $x$ -axis is looking forward,  $y$ -axis to the left side and  $z$ -axis up. Following equations describes moving in forward direction and rotation around vertical axis:

$$F_X = F_{LX} + F_{RX} = ma_{tang} \quad (8)$$

$$M_Z = F_{RX}r - F_{LX}r = J\alpha_Z \quad (9)$$

Letters  $F$  and  $M$  stand for forces and torques,  $m$  and  $J$  are mass and moment of inertia and  $r$  is half the distance between both wheels. Indices  $X$ ,  $Y$  (not used here) and  $Z$  defines the direction. Indices  $L$  and  $R$  tells whether corresponding force is on the left or on the right wheel. Linear or also called tangential acceleration  $a_{tang}$  and angular acceleration  $\alpha$  are time derivatives of corresponding velocities.

$$a_{tang} = \frac{dv}{dt} \quad (10)$$

$$\alpha = \frac{d\omega}{dt} \quad (11)$$

From both equations can be seen that the presence of the grip on both wheels is needed to change linear or angular velocity. Two conditions must be fulfilled to have sufficient grip. First one is proper friction coefficient and second is proper vertical forces  $F_{LZ}$  and  $F_{RZ}$  on both wheels. Friction coefficient depends on surface and wheel tyre quality. In most cases only wheel tires can be improved. To see what happen with the vertical wheel forces when the robot is driving the third dimension of the robot should be studied.

The external forces to the rigid body can be presented with common force and common torque. Both of them have 3 components. Two of them have already been written in eqs. (8) and (9). The remaining for are the following:

$$F_{LZ} + F_{RZ} + F_{1Z} + F_{2Z} - mg = 0 \quad (12)$$

$$F_{LY} + F_{RY} = ma_{tang} \quad (13)$$

$$(F_{LZ} - F_{RZ})r = ma_{rad}h_{cg} \quad (14)$$

$$mgb - (F_{LZ} + F_{RZ})b = m|a_{tang}|h_{cg} \quad (15)$$

Center of gravity height is presented as  $h_{cg}$  and  $b$  is  $x$ -axis component of distance from wheels to the slider. Index 1 and 2 stands for front and rear slider respectively (Fig. 4) and  $a_{rad}$  is radial acceleration:

$$a_{rad} = v\omega \quad (16)$$

Other indices have already been presented together with equations (8) and(9).

Two examples will be studied. First example will show us the equilibrium of forces and moments in linear direction. The situation is shown in Fig. 4. It can be seen that a great part of vertical force is on the rear slider ( $F_{2Z}$ ). For the same part vertical forces on main wheels are smaller as can be also find out from equation (12). Vertical force on the slider is bigger if the center of gravity is higher. So in the case of linear acceleration lower gravity center is better.

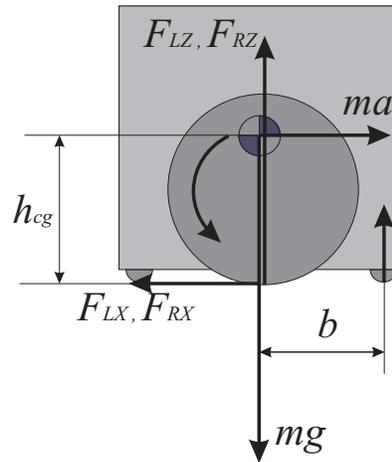


Fig. 4 - Forces on the robot during acceleration

Second example will show us, how the gravity center height affect driving in the circle. The situation when no linear acceleration is present is shown in Fig. 5. All weight of the robot is on the main wheels, so the grip is at it's maximum. But inner wheel of the robot takes less vertical force so inner wheel grip is lower than outer wheel grip and the difference is proportional to the center of gravity height. When the robot is forced with  $x$ -direction force on inner wheel it is again appropriate that the robot's center of gravity is as low as possible.

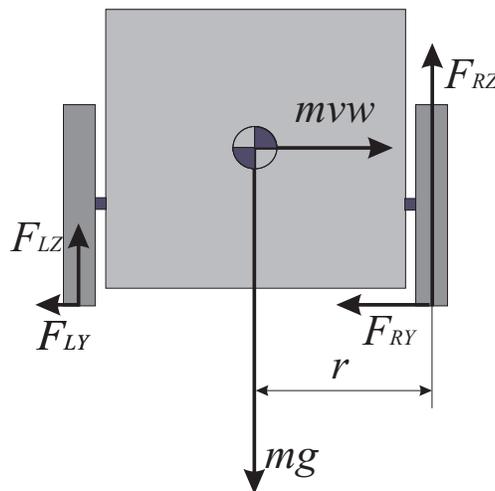


Fig. 5 - Forces during circling with constant speed

Most of situations are something in between of both situations. There exist both the tangential and radial acceleration at the same time. The given explanation is purely theoretical. In practice there are a lot of unknown factors that affect exact calculations, so in the sequel the problem is simplified. The robot is compressed into a moving mass point.

Due to the fact that tangential and radial acceleration are geometrically orthogonal, the overall acceleration that the grip must handle is their Pythagorean sum:

$$a = \sqrt{a_{tang}^2 + a_{rad}^2} \quad (17)$$

The theoretical background about sliding was given. We determined that maximum tangential acceleration is lower than radial 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. The radial acceleration is caused by turning at certain speed and is the product of linear and angular velocity

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 equation (16). 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 that was obtained using the vision system. 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 on Fig. 6.

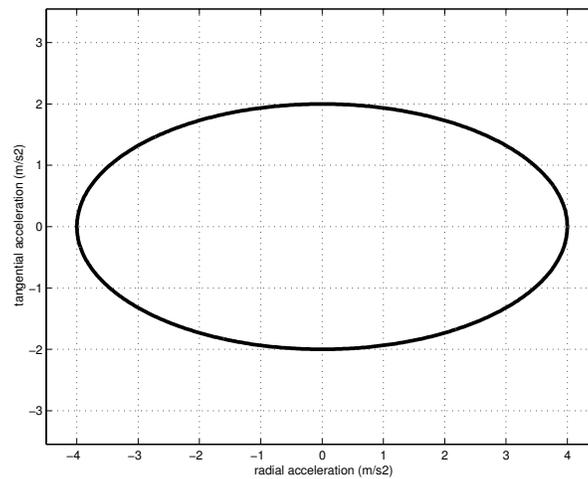


Fig. 6 - Acceleration limitations

## 4 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 so in this paper a quick overview will be given. The two dimensional curve is obtained 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 neighbor 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 solving a linear equation system. If the  $p$ -th order spline consists of  $m$  segments, than the number of parameters to determine is

$$m(p+1) \quad (18)$$

Number of linear equations is

$$n + (m-1)p \quad (19)$$

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:

$$m = n - p \quad (20)$$

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. 7 shows the sample of set conditions to design the spline.

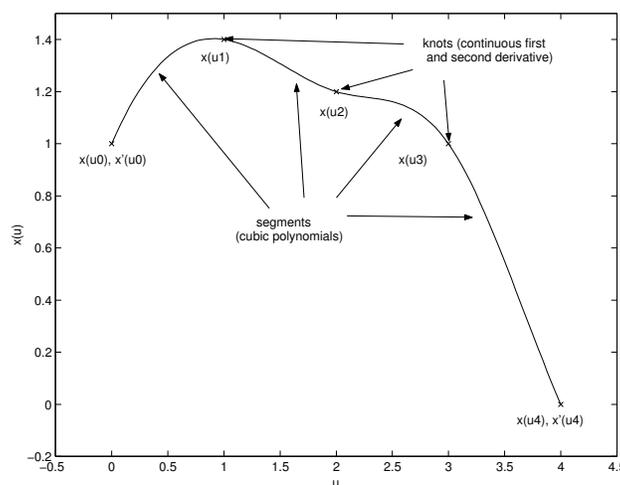


Fig. 7 - Spline

There are 6 conditions ( $n=6$ ) to define each of splines, which consist of 3 segments ( $m=3$ ). According to eq. (20) this leads to the cubic spline. New inserted CP raises  $n$  and  $m$  for 1 and eq.(20) remains fulfilled.

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

$$\varphi_{SP} = \arctg \frac{y'(u_{\min})}{x'(u_{\min})} , \quad \varphi_{EP} = \arctg \frac{y'(u_{\max})}{x'(u_{\max})} \quad (21)$$

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 is proposed:

$$\begin{aligned} \sqrt{x'(u_{SP})^2 + y'(u_{SP})^2} &\approx \frac{\text{dist}(SP, \text{first CP})}{u_{\text{firstCP}} - u_{SP}} \\ \sqrt{x'(u_{EP})^2 + y'(u_{EP})^2} &\approx \frac{\text{dist}(\text{last CP}, EP)}{u_{EP} - u_{\text{lastCP}}} \end{aligned} \quad (22)$$

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  $\kappa$ . When dealing with spline curves in two dimensions  $\kappa$  is given as follows:

$$\kappa(u) = \frac{x'(u)y''(u) - y'(u)x''(u)}{(x'(u)^2 + y'(u)^2)^{3/2}} \quad (23)$$

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

## 5 Finding 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

$$t = \int_{\substack{\text{initial} \\ \text{position}}}^{\text{target}} \frac{ds}{v} \quad (24)$$

Considering

$$ds = \sqrt{x'(u)^2 + y'(u)^2} du, \quad (25)$$

Eq. (24) can be written as

$$t = \int_a^b \frac{\sqrt{x'(u)^2 + y'(u)^2}}{v(u)} du \quad (26)$$

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 optimization with control points as free parameters to be optimized. The optimization procedure is as follows:

- 1.) Choose initial control points and calculate the initial path. An example of this is shown in Fig. 8.
- 2.) For given path the highest allowable overall velocity profile is calculated as follows:
  - Its curvature is calculated according to Eq. (23) as shown in Fig 9.
  - 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 is bigger than in TP. Before and after the TP the robot must tangentially decelerate and accelerate respectively as maximally 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. 10. 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 neighbor TP). But that neighbor 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. 10 and 11 by bold curves.
  - The initial and final (if required) velocities must be on the highest allowable overall velocity profile (as it is in Fig 11). If not, the given path cannot be driven without violating acceleration constraints. (The case in Fig 10)

- For given highest allowable velocity profile the cost function (time) is calculated according to Eq.(26).

3.) Optimize the problem with control points as optimizing parameters.

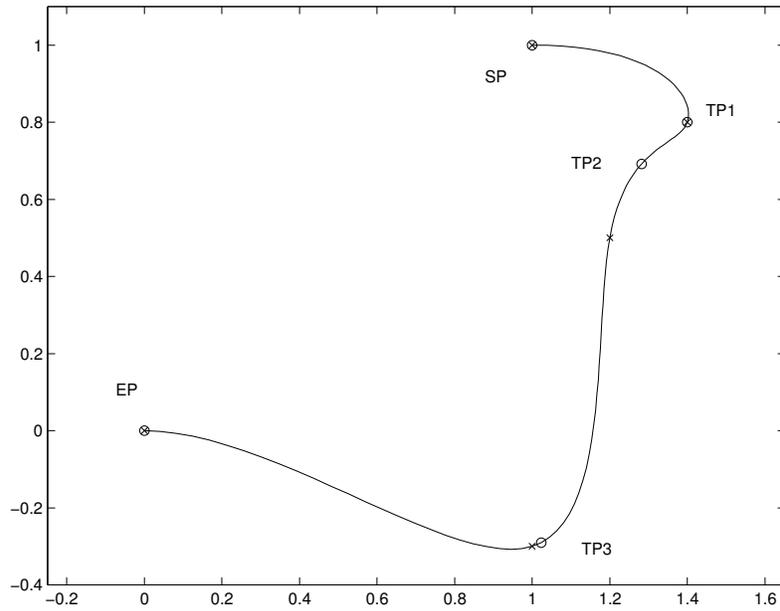


Fig. 8 - Example of the curve

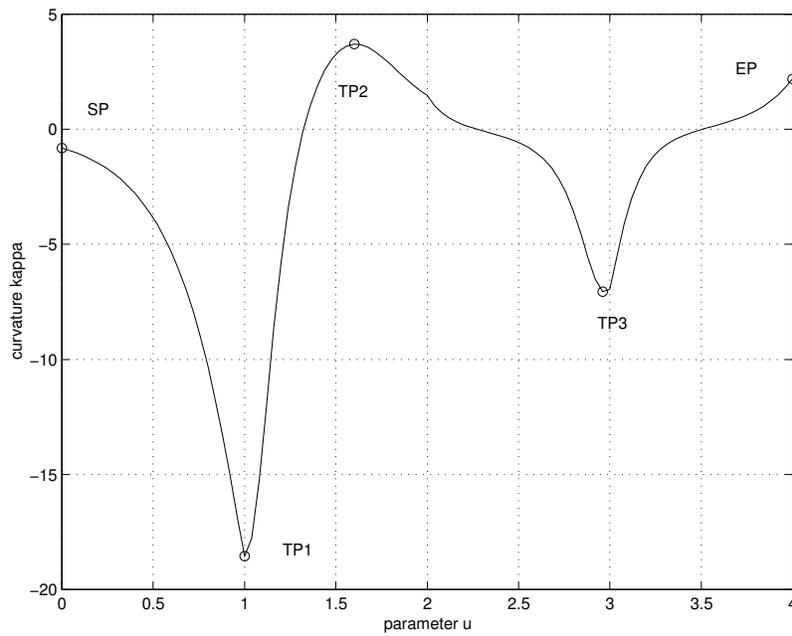


Fig. 9 - The curvature

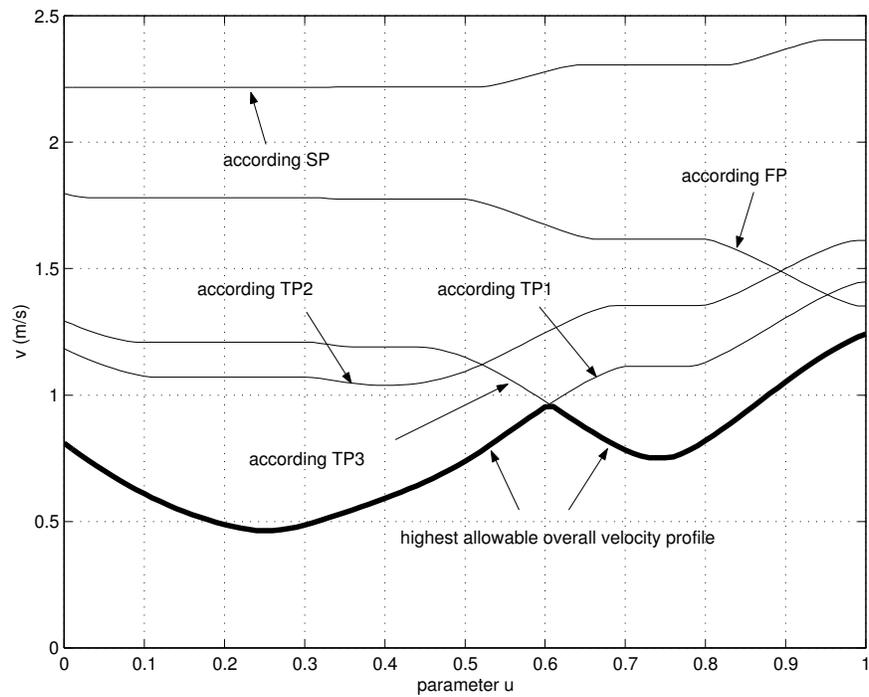


Fig. 10 - The velocity profile

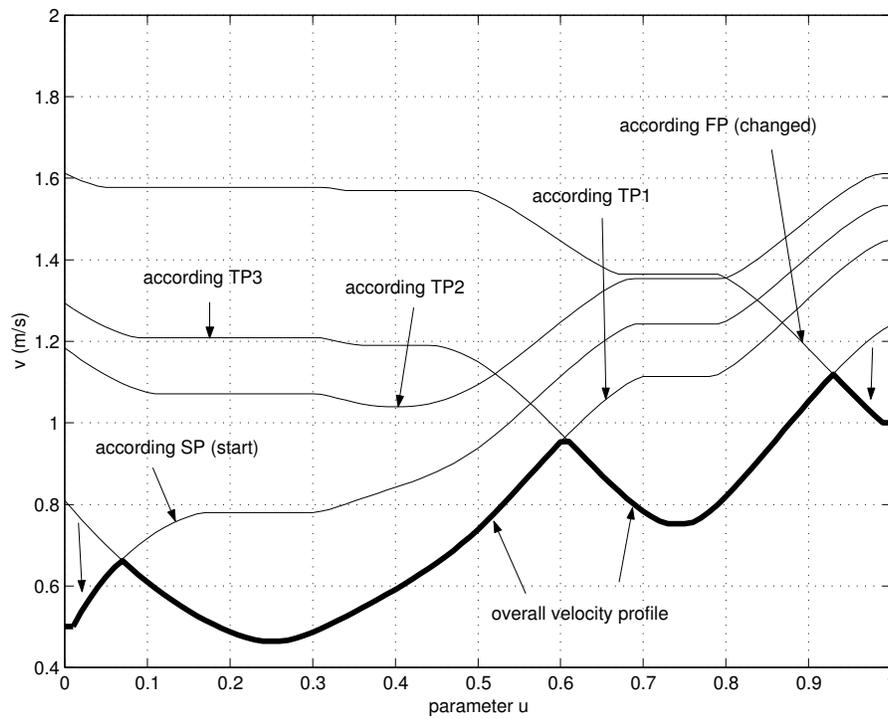


Fig. 11 - The corrected velocity profile concerning initial and terminal velocity

## 5.1 Case Study

The objective of this case study is 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  $225^\circ$  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  $180^\circ$ . The question is how many control points are needed. Two points are needed to fulfill 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. 12 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<sup>th</sup> CP improves the time for a tenth of a second (more than 6 %) and the 5<sup>th</sup> would improve it for only one hundredth of a second.

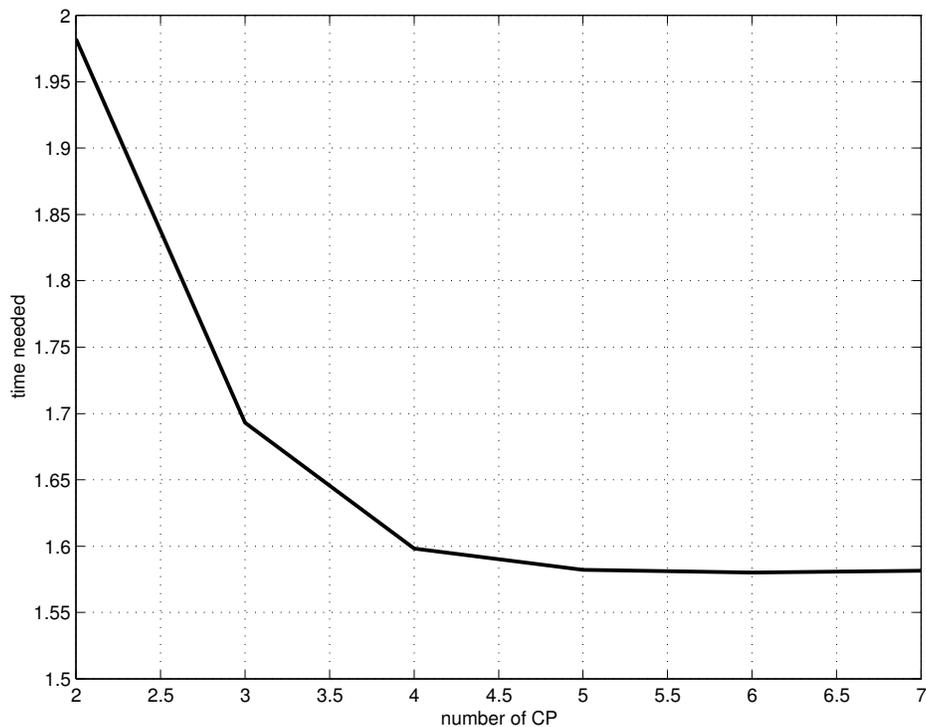


Fig. 12 - The needed time with the respect to the number of CP

The result paths are shown in Fig. 13. Thick line presents the path constructed from 4 CPs. On the same figure are also presented paths constructed from 2, 3 and 5 CPs, all presented with thin black line. It can be seen where 2 and 3 CP paths spend too much time because of not well-defined path. The 5 CP path is slightly different from the 4 CP one and the difference lies in the area where a large improvement cannot be done. The 6 and 7 CP paths are nearly the same as 5 CP-path and are not shown in the figure.

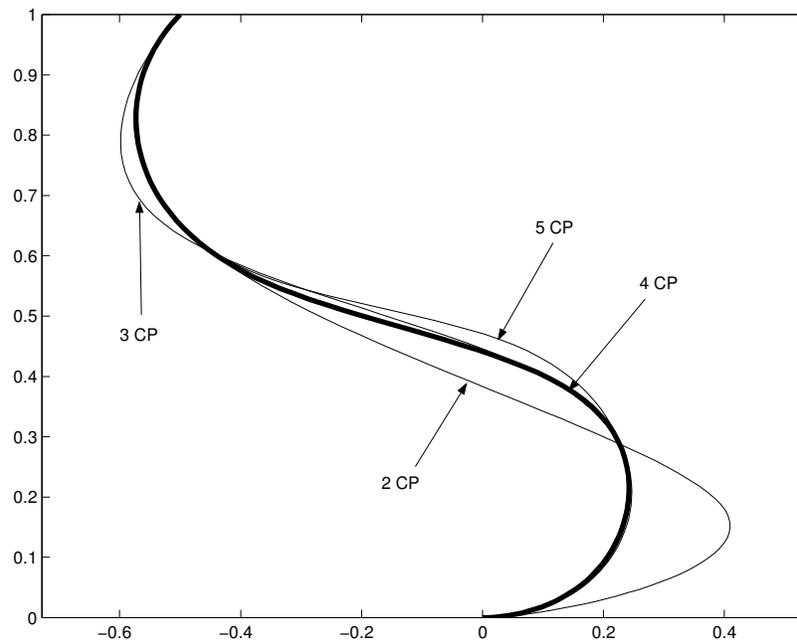


Fig. 13 - Paths with different number of CP

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.

## 6 Path Tracking

Path tracking requires high quality control. Ordinary closed loop control systems generally compare output signal to the reference signal and act with the respect to the difference between them. That works well when the reference signal is more or less constant. The path-tracking problem is particularly not that case. Since the path is generally not straight and has a lot of turns requires frequent and continuous control actions. On the other hand the reference i.e. the planned path is well defined and can be used for calculation of the control action using the inverse model of the robot.

The tracking will be based on the cinematic model of the robot. Its dynamics will be neglected. There are at least three reasons for this:

- Simplification of the problem
- Neglecting of the dynamics at moderate speed does not affect the result

- The construction of most of the robots. Its input is usually the reference for the velocity and angular velocity. The user cannot force the robot directly with the torque or acceleration.

The feed-forward command is in case of proposed path tracking combined with the controller for nonholonomic systems. The tracking error defined in coordinates relative to the controlled robot is:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \varphi_r - \varphi \end{bmatrix} \quad (27)$$

Where  $x_r$ ,  $y_r$ ,  $\varphi_r$  are coordinates of the reference robot which is moving ideally on the reference path, and  $x$ ,  $y$ ,  $\varphi$  are coordinates of the controlled robot. Considering the cinematic relations in equation (7) and the derivation the following equation is got:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \cos e_3 & 0 \\ \sin e_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{r1} \\ u_{r2} \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (28)$$

The control signal is with the respect to the equation (28) determined as:

$$\begin{aligned} u_1 &= u_{r1} \cos e_3 - v_1 \\ u_2 &= u_{r2} - v_2 \end{aligned} \quad (29)$$

where first parts of the right side represents feed forward signal and  $v_1$  and  $v_2$  control loop signals. They can be expressed as:

$$\begin{aligned} v_1 &= u_{r1} \cos e_3 - u_1 \\ v_2 &= u_{r2} - u_2 \end{aligned} \quad (30)$$

and inserted in equation (28). The result is the following:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} 0 & u_2 & 0 \\ -u_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} \cdot u_{r1} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (31)$$

Model can be linearized in the point  $e_1 = e_2 = e_3 = 0$ ,  $v_1 = v_2 = 0$ .

$$\Delta \dot{\mathbf{e}} = \begin{bmatrix} 0 & u_{r2} & 0 \\ -u_{r2} & 0 & u_{r1} \\ 0 & 0 & 0 \end{bmatrix} \cdot \Delta \mathbf{e} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \cdot \Delta \mathbf{v} \quad (32)$$

Let us define linear closed loop controller of the states as:

$$\mathbf{v} = \mathbf{K} \cdot \mathbf{e} \quad (33)$$

Matrix  $\mathbf{K}$  is the size of  $2 \times 3$  due to the fact there are 2 inputs for the control of 3 states. Schematically the control of the robot is presented in the Fig: 14

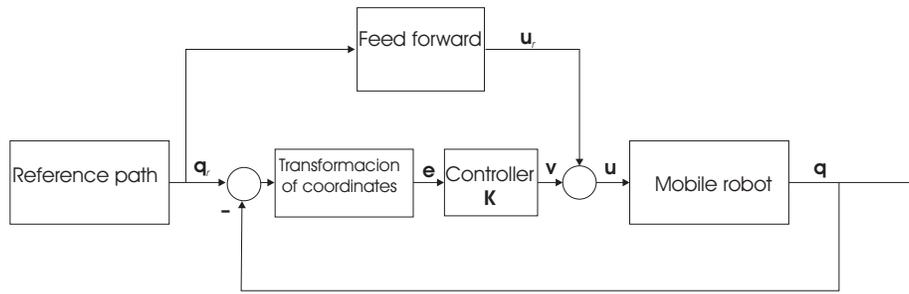


Fig. 14 - Control loop

Considering Fig. 14, we can find the desired structure of the controller. To decrease the error  $e_1$ , the action should be on the first input (the velocity) and similarly to decrease the error  $e_3$  the action should be only on the second input (the angular velocity). The error  $e_2$  can also be decreased by action on the second input. So we get:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} -k_1 & 0 & 0 \\ 0 & -k_2 & -k_3 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad (34)$$

To determine parameters of the controller  $k_1, k_2, k_3$  all procedures from the control theory can be used. In our case the parameters will be obtained by comparison of the actual and desired characteristic polynomial. For the third order system, the desired polynomial can be written:

$$(s + 2\xi\omega_n)(s^2 + 2\xi\omega_n s + \omega_n^2) \quad (35)$$

With the analogy to the second order system the damping parameter should be  $\xi \in (0,1)$  and natural frequency should be positive  $\omega_n > 0$ . Additional pole at  $s = -2\xi\omega_n$  causes increase of rise time and decrease of the overshoot [20].

Actual polynomial of the closed loop can be determined [21]:

$$\det(s\mathbf{I} - \mathbf{A} + \mathbf{BK}) = s^3 + (k_1 + k_3)s^2 + (k_1k_3 + k_2u_{r1} + u_{r2}^2)s + k_1k_2u_{r1} + k_3u_{r2}^2 \quad (36)$$

The result of the comparison of equations (35) and (36) is

$$\begin{aligned} k_1 + k_3 &= 4\xi\omega_n \\ k_1k_3 + k_2u_{r1} + u_{r2}^2 &= 4\xi^2\omega_n^2 + \omega_n^2 \\ k_1k_2u_{r1} + k_3u_{r2}^2 &= 2\xi\omega_n^3 \end{aligned} \quad (37)$$

We can choose:

$$k_1 = k_3 = 2\xi\omega_n \quad (38)$$

as proposed in [9] and get:

$$k_2 = \frac{\omega_n^2 - u_{r2}^2}{|u_{r1}|} \quad (39)$$

It can be seen, that natural frequency should be higher than maximal angular speed  $\omega_n > u_{r2MAX}$ . Parameter limits to infinity when  $u_{r1}$  limits towards 0, so it is appropriate to choose scheduled parameter  $k_2 = g \cdot u_{r1}$ . In that case natural frequency is

$$\omega_n = \sqrt{u_{r2}^2(t) + gu_{r1}^2(t)} \quad (40)$$

and the parameters of the controller are

$$\begin{aligned} k_1 &= k_3 = 2\xi\omega_n \\ k_2 &= g \cdot u_{r1} \end{aligned} \quad (41)$$

When the velocity decreases, also the parameters  $k_1, k_2, k_3$  limits towards 0. In that case the system is not controllable. If the controller parameters are obtained from equations (38) and (39) than the nonlinear and time variant controller assures constant and stable poles:

$$\begin{aligned}
s_1 &= -2\xi\omega_n \\
s_2 &= -2\xi\omega_n + \omega_n\sqrt{\xi^2 - 1} \\
s_3 &= -2\xi\omega_n - \omega_n\sqrt{\xi^2 - 1}
\end{aligned} \tag{42}$$

Although the poles are constant and stable, the local stability is not assured, because the system is time variant[9,14]. Stability can be checked by the Lyapunov analysis.

## 6.1 Stability Analysis

Lyapunov function  $V(\mathbf{q})$  is treated like energy, so it should be defined as positive definite and continuous with continuous derivatives on all components of  $\mathbf{q}$ . At the equilibrium  $\mathbf{q} = \mathbf{0}$  should be  $V(\mathbf{q}) = 0$ . If the first derivative is negatively semi-definite  $\dot{V}(\mathbf{q}) < 0$ , than equilibrium  $\mathbf{q} = \mathbf{0}$  is asymptotically stable point.

Let us present the controller from the equation (34)

$$\begin{aligned}
v_1 &= -k_1 e_1 \\
v_2 &= -k_2 u_{r1} e_2 - k_3 e_3
\end{aligned} \tag{43}$$

and determine the Lyapunov function as:

$$V(\mathbf{e}) = \frac{k_2}{2}(e_1^2 + e_2^2) + \frac{e_3^2}{2} \tag{44}$$

Considering the linearized robot model (32) and the controller (43) we get the following time derivative of the Lyapunov function:

$$\dot{V}(\mathbf{e}) = -k_1 k_2 e_1^2 - k_3 e_3^2 \tag{45}$$

which obviously is negatively semi definite.

## 6.2 Case Study

At first let us present an example when the robot is controlled just with feed-forward signal. As already said, this type of control is sensitive to wrong initial state and also to the noise of input signals. An example with initial state error can be seen in Fig. 15 and in Fig. 16 influence of noised input signal. Because there is no feedback the robot is unable to cancel an error.

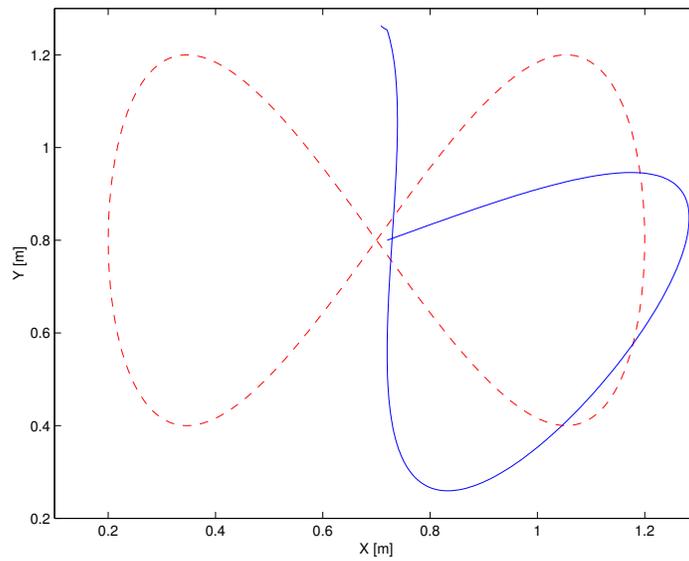


Fig. 15 - Open loop (only feed forward) control with initial state error (red – reference, blue - robot)

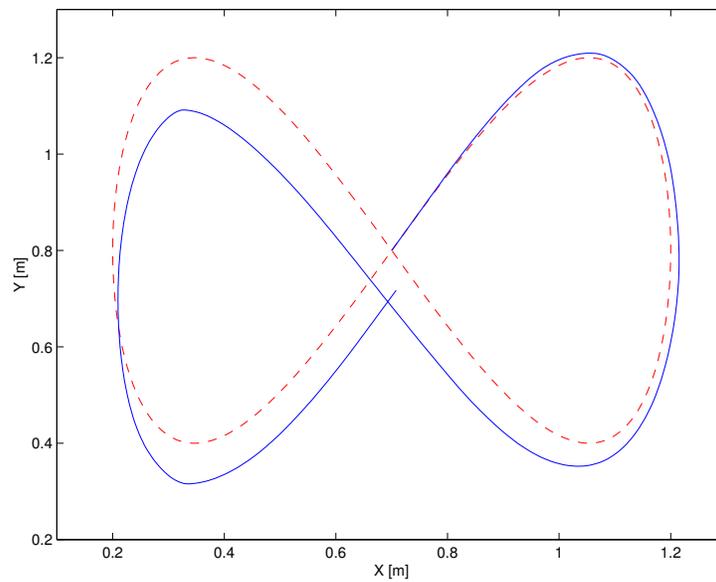


Fig. 16 - Open loop (only feed forward) control with noised input signal (red – reference, blue - robot)

To cancel an error the feedback should be added like it is presented in Fig. 14 and with equation (41). The parameter  $k_2$  is calculated using equation (39). In Fig. 17 the control performance is presented in the case of initial state error and noised input signal. The controller is in that case tuned to  $\omega_n = 2$  and  $\xi = 0.6$ . In Fig. 18 a similar case is presented. The controller was tuned to  $\omega_n = 4$  and  $\xi = 0.5$ .

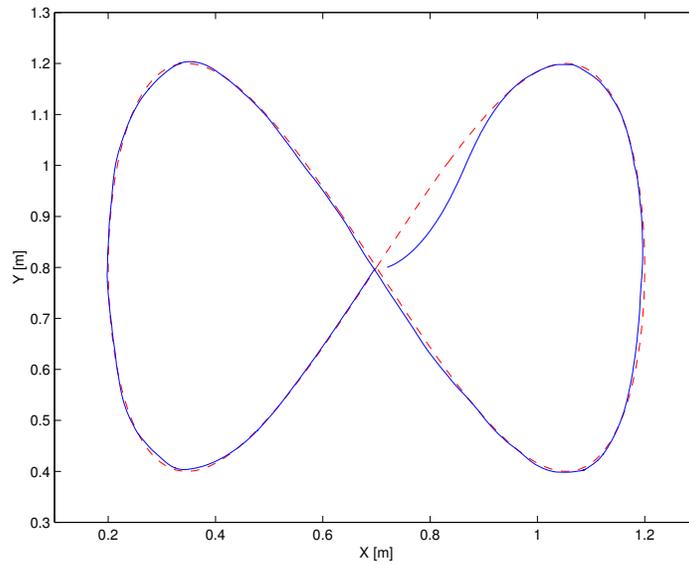


Fig. 17 - Closed loop control ( $\omega_n = 2$ ,  $\xi = 0.6$ ) (red – reference, blue - robot)

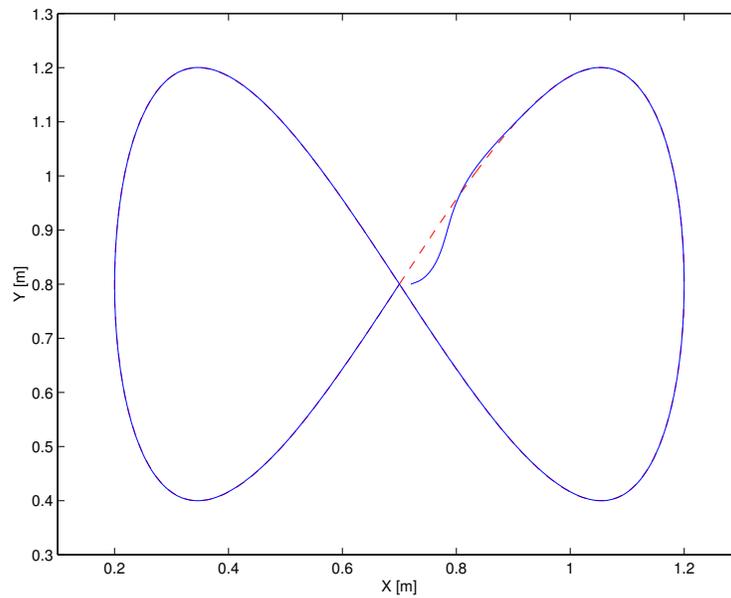


Fig. 18 - Closed loop control ( $\omega_n = 4$ ,  $\xi = 0.5$ ) (red – reference, blue - robot)

The controller was also tested on more complex reference path. The result is presented in Fig.19.

Comparison between to the controller that is using constant parameter  $k_2$  has been made. The simulation results using constant  $k_2$  is shown in Fig. 20.

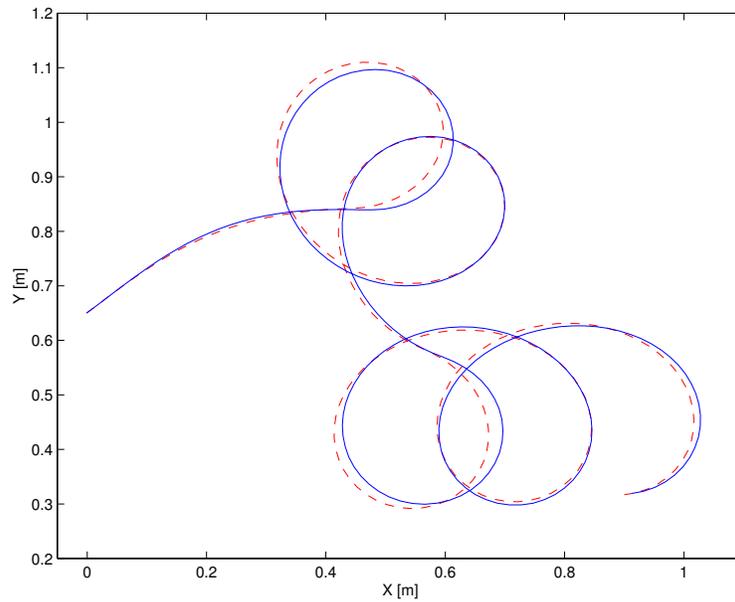


Fig. 19 - Closed loop control on more complex path ( $\omega_n = 4.5$ ,  $\xi = 0.1$ ) (red – reference, blue - robot)

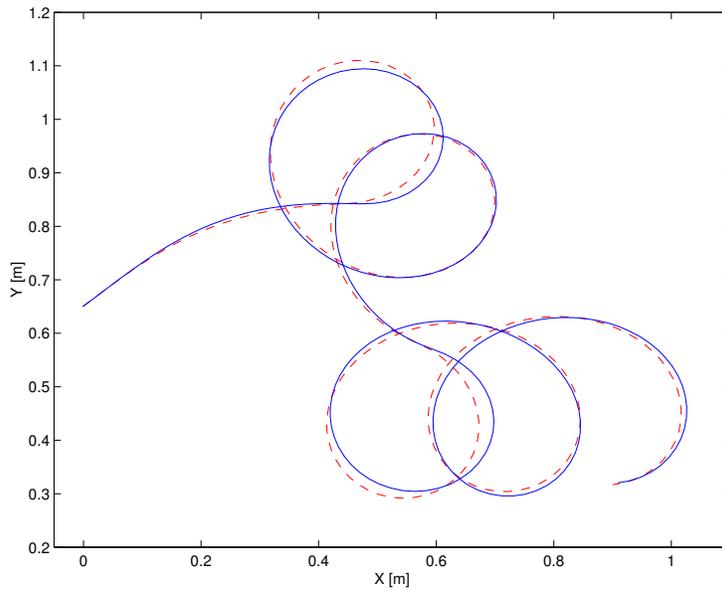


Fig. 20 - Closed loop control ( $\omega_n = 4.5$ ,  $\xi = 0.1$ ,  $k_2 = 25$ ) (red – reference, blue - robot)

The results are quite similar. Slightly better result is got with scheduled parameter  $k_2$  as it can be seen from the comparison of input signals. The comparison is more objective if the criterion function is used. The ISE (Integral of Square Error) for our case can be written:

$$J = \int_{t=0}^T (e_1^2(t) + e_2^2(t) + e_3^2(t)) dt$$

In the Table 1 the values of the criterion function are presented. According to the Table 1 the conclusion can be made that the controller with scheduled parameter  $k_2$  is slightly better. Also open loop control (only feed-forward signal) is added. It starts without initial state error and is simulated without noised signals.

Table 1 - Comparison of different controllers

Type of controller	Criterion $J$
Open loop	0.30795
Closed loop, variable $k_2$	0.11006
Closed loop, constant $k_2$	0.11815

## 7 Conclusion

The path finding algorithm for nonholonomic mobile robots is proposed. The case study concerned slippery conditions in robot soccer environment. The path is presented as a spline curve and is obtained with positioning of the control points positioning. The control points were placed using the optimization function where the criterion was needed time. The state space controller is proposed for high quality path tracking. Also the feed forward signal was used to minimize error. Otherwise the controller cannot react before the error appears. That is not acceptable in case of path tracking. Its stability is proven with Lyapunov stability theorem.

## References

- [1] 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.
- [2] Bennewitz, M., Burgard, W., Thrun, S. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and Autonomous Systems*, Vol.41, pp. 89-99, 2002.
- [3] Borenszejn, P., Jacobo, J., Mejail, M., Stoliar, A., Katz, A., Cecowski, M., Ferrari, A., Santos, J., Design of the Vision System for the UBA-Sot team. *Proceedings of the 2002 FIRA Robot World Congress*, Vol. 1, pp. 616-619, Seoul, May 26-29 2002.
- [4] 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.
- [5] Cha, Y. Y., Gweon, D. G. Local Path Planning of a Free Ranging Mobile Robot Using the Directional Weighting Method. *Mechatronics*, Vol. 6, No.1, pp. 53-80, 1996.

- 
- [6] Desaulniers, G. On Shortest paths for a car-like robot maneuvering around obstacles. *Robotics and Autonomous Systems*, Vol. 17, pp. 139–148, 1996.
  - [7] Egerstedt, M., Hu, X. A hybrid control approach to action coordination for mobile robots. *Automatica*, Vol. 38, pp. 125-130, 2002.
  - [8] 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.
  - [9] Luca A., Oriolo G., Vendittelli M.: Control of wheeled robots: An experimental overview, *RAMSETE – Articulated and Mobile Robotics for Services and Technologies*, S. Nicosia, B. Siciliano, A. Bicchi, P. Valigi, Eds., Springer – Verlag, 2001.
  - [10] Marchese, F. M. A directional diffusion algorithm on cellular automata for robot path-planning. *Future Generation Computer Systems*, Vol. 18, pp. 983–994, 2002.
  - [11] Martin, C. F., Sun, S., Egerstedt, M.. Optimal Control, Statistics and Path Planning. *Mathematical and Computer Modelling*, Vol. 33, pp. 237-253, 2001.
  - [12] 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.
  - [13] Minoru, A.; D’Andrea; R., Birk; A., Kitano, H., Veloso, M. Robotics in Edutainment. *Proceedings of the 2000 IEEE International Conference on Robotics & Automation (ICRA’2000)*, pp. 795-800, San Francisco, USA, Apr. 2000.
  - [14] Oriolo G., Luca A., Vandittelli M.: WMR Control Via Dynamic Feedback Linearization: Design Implementation and Experimental Validation, *IEEE Transactions on Control Systems Technology*, Vol. 10, No. 6, pp 835-852, 2002.
  - [15] 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.
  - [16] Stone, P., Veloso, M. Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robots*, Vol. 8, No. 3, pp. 345-383, 2000.
  - [17] 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.
  - [18] Švestka, P., Overmars, M. H. Coordinated path planning for multiple robots. *Robotics and Autonomous Systems*, Vol. 23, pp. 125-152, 1998.
  - [19] Ting, Y., Lei, W. I., Jar, H. C. A Path Planning Algorithm for Industrial Robots. *Computers & Industrial Engineering*, Vol. 42, pp. 299-308, 2002.
  - [20] Zupančič B.: Zvezni regulacijski sistemi I. del, Fakulteta za elektrotehniko in računalništvo, Univerza v Ljubljani, Ljubljana, 1996
  - [21] Zupančič B.: Zvezni regulacijski sistemi II. del, Fakulteta za elektrotehniko in računalništvo, Univerza v Ljubljani, Ljubljana, 1996
  - [22] Weber, H. A motion planning and execution system for mobile robots driven by stepping motors. *Robotics and Autonomous Systems*, Vol. 33, pp. 207–221, 2000.