Proceedings of the 13th
Mediterranean Conference on Control and Automation
Limassol, Cyprus, June 27-29, 2005

WeM05-6

Mobile Robot Control on a Reference Path

Gregor Klančar, Drago Matko, Sašo Blažič

Abstract— In this paper a control design of a nonholonomic mobile robot with a differential drive is presented. On the basis of robot kinematics equations a robot control is designed where the robot is controlled to follow the arbitrary path reference with a predefined velocity profile. The designed control algorithm proved stabile and robust to the errors in robot initial positions, to input and output noises and to other disturbances. The obtained control law is demonstrated on a simple trajectory example, however, for a more general applicability a time-optimal motion planning algorithm considering acceleration constraints is presented as well.

I. INTRODUCTION

The control of a mobile robot with nonholonomic constraints on a reference path is an important task in mobile robotics [3], [4], [6], [9]. Nonholonomic systems have motion limitations emerging from their kinematics model. Therefore some directions of motion are not possible. This paper deals with a mobile robot with differential drive which cannot move in the direction lateral to the wheels. The control of this robot is solved by considering its first order kinematics model. The obtained motion can later be upgraded to include the dynamics properties also (inertia an mass) where the robot control problem is transferred to a control of a second order kinematics model ([9], [6]). The control of mobile robots considering only first order kinematics is very common in literature ([2], [4], [8], [1]) as well as in practice. This is mainly because the control problem is easier to solve, the system dynamics can usually be neglected (fast and strong motors), especially at moderate speeds and because the robot design sometimes does not allow torque or acceleration to be forced at robot input (only reference speed). The basic control of a mobile robot in obstacle-free environment can be solved by point-to-point control with classic control where the intermediate course of the states between start and end state is not important. The other possibility is to control the robot to follow the reference path from the start to the end position. When controlling nonholonomic systems, it is usually more appropriate to control the system (by feedback) to follow the reference path ([6], [9]). When controlling the robot to the end point (classic control), the control law usually has to be discontinuous and time variable ([2], [3], [6]). Further on the robot has to consider

G. Klančar is with Faculty of Electrical Engineering, University of Ljubljana Tržaška 25, SI-1000 Ljubljana, Slovenia gregor.klancar@fe.uni-lj.si
D. Matko is with Faculty of Electrical Engineering, University of Ljubljana Tržaška 25, SI-1000 Ljubljana, Slovenia drago.matko@fe.uni-lj.si
S. Blažič is with Faculty of Electrical Engineering, University of Ljubljana Tržaška 25, SI-1000 Ljubljana, Slovenia saso.blazic@fe.uni-lj.si
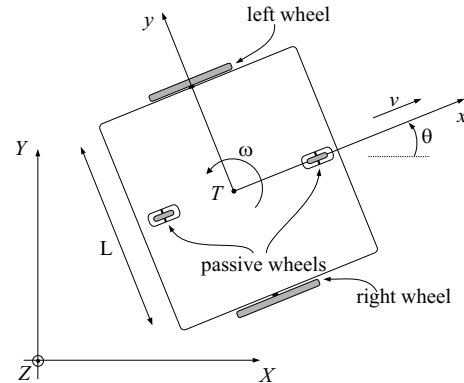
Fig. 1.   Robot architecture and symbols

nonholonomic constraints so its path cannot be arbitrary. The robot usually moves in the environment with obstacles, limitations and other demands which all somehow defines desired robot path. All these facts give advantage to control on a reference path which should follow all kinematic constraints. Nonholonomic systems usually have feedforward control ([2], [9], [6], [8]) where system inputs are calculated from the known trajectory and all kinematics constraints are implicitly considered by trajectory design. However, the use of the open-loop control only (just feedforward) is practically useless because it is not robust to errors in initial system states and other disturbances during operation. Closed loop is therefore added for practical use. The above mentioned combination (feedforward control) is intuitive and suitable for most nonholonomic mechanical systems. The paper is organised as follows. First the mobile robot motion modelling is revealed, followed by the control law derivation, the stability analysis and an example of the used control. Then an idea of time-optimal motion planning algorithm considering acceleration constraints is presented. The paper ends with conclusions.

II. MOBILE ROBOT MODELLING

In the sequel the direct and the inverse kinematics for the mobile robot with differential drive are determined. The robot's architecture together with its symbols is shown in Fig. 1 where it is supposed that its point of geometric centre T and the centre of gravity coincide.

The motion equations are as follows

$$
\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1}
$$

where v and ω are tangential and angular velocities of the platform in Fig. 1. Right and left velocities of the robot

wheels are then expressed as $v_R = v + \frac{\omega L}{2}$ and $v_L = v - \frac{\omega L}{2}$.

For a given reference trajectory $(x_r(t), y_r(t))$ defined in a time interval $t \in [0,T]$ the feedforward control law can be derived. From the obtained inverse kinematics the robot inputs are calculated which drive the robot on a desired path only if there are no disturbances and no initial state errors. The needed robot inputs, tangential velocity v_r and angular velocity ω_r are calculated from the reference path. The tangential velocity reads

$$v_r(t) = \pm\sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \qquad (2)$$

where the sign depends on the desired drive direction (+ for forward and – for reverse). The tangent angle of each point on the path is defined as

$$\theta_r(t) = \arctan2(\dot{y}_r(t), \ \dot{x}_r(t)) + k\pi \qquad (3)$$

where $k=0,1$ defines the desired drive direction (0 for forward and 1 for reverse) and function arctan2 is the inverse tangens function which returns the correct angles in all situations. By calculating the time derivative of Eq. (3) the robot angular velocity is obtained

$$\omega_r(t) = \frac{\dot{x}_r(t)\ddot{y}_r(t) - \dot{y}_r(t)\ddot{x}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)} = v_r(t)\kappa(t) \qquad (4)$$

where $\kappa(t)$ is path curvature. By following relations from (1) to (4) and the defined reference robot path $\mathbf{q}_r(t) = [x_r(t), y_r(t), \theta_r(t)]^T$ robot inputs $v_r(t)$ and $\omega_r(t)$ are calculated. The necessary condition in the path design procedure is twice differentiable path and nonzero tangential velocity $v_r(t) \neq 0$. If for some time t tangential velocity is $v_r(t)=0$, the robot rotates at a fixed point with the angular velocity $\omega_r(t)$. Angle $\theta_r(t)$ cannot be determined from Eq. (2) and therefore $\theta_r(t)$ must be given explicitly.

III. CONTROL DESIGN

When the robot is controlled to drive on a reference path, it usually has some following error. The following error expressed in the frame of the real robot, as shown in Fig. 2, reads

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \qquad (5)$$

In Fig. 2 the reference robot is an imaginary robot which ideally follows the reference path. On the contrary the real robot (when compared to the reference robot) has some error when following the reference path. Therefore the control algorithm should be designed to force the robot to follow the reference path precisely.

Considering the robot kinematics (1) and deriving relations (5) the following kinematics model is obtained

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \cos e_3 & 0 \\ \sin e_3 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_{r1} \\ u_{r2} \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad (6)$$
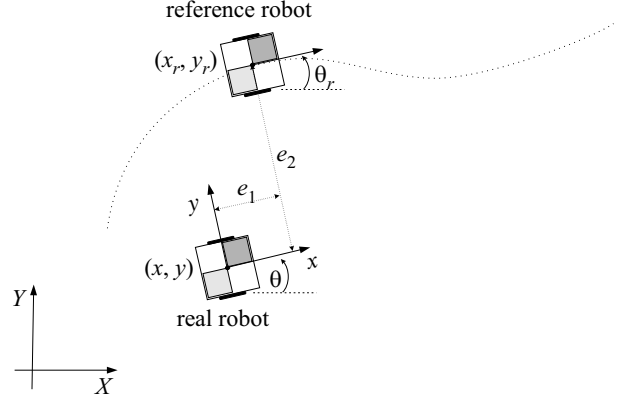

Fig. 2. Robot following error transformation

where u_{r1} is feedforward tangential velocity (2) and u_{r2} is feedforward angular velocity (4). Robot inputs, regarding relation (6), can be expressed in the following form

$$\begin{aligned} u_1 &= u_{r1}\cos e_3 - v_1 \\ u_2 &= u_{r2} - v_2 \end{aligned} \qquad (7)$$

where the first parts on the right side of the equal sign are the feedforward inputs, while v_1 and v_2 are the inputs from the closed loop. From (7) expressing the closed-loop inputs and rewriting Eq. (6) results in

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} 0 & u_2 & 0 \\ -u_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \\ + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} \cdot u_{r1} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \qquad (8)$$

Further on linearizing (8) around the operating point OP (OP: $e_1 = e_2 = e_3 = 0$, $v_1 = v_2 = 0$) results in the following linear model

$$\Delta\dot{\mathbf{e}} = \begin{bmatrix} 0 & u_{r2} & 0 \\ -u_{r2} & 0 & u_{r1} \\ 0 & 0 & 0 \end{bmatrix} \cdot \Delta\mathbf{e} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \cdot \Delta\mathbf{v} \quad (9)$$

which is in the state space form $\Delta\dot{\mathbf{q}} = \mathbf{A} \cdot \Delta\mathbf{q} + \mathbf{B} \cdot \Delta\mathbf{u}$. Kalman controllability matrix has full rank $rank(\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B})=3$ if either u_{r1} or u_{r2} is nonzero which is sufficient condition for controllability only when reference inputs u_{r1} and u_{r2} are constant. However this is only true in case of linear and circular paths. The controllability of a driftless system can be ascertained by Chow's theorem if the system is completely nonholonomic (all constraints are nonholonomic). The system (1) is completely nonholonomic as it has only one nonholonomic constraint $\dot{y}_c\cos\theta - \dot{x}_c\sin\theta = 0$ (robot cannot move in lateral direction to its wheels) and therefore the system is controllable.

Let us further on define the linear state space controller for the closed loop. The system has three states and two inputs, thus
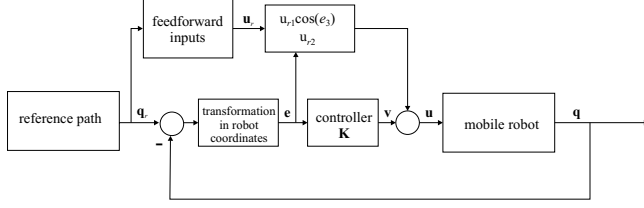
Fig. 3. Mobile robot control schematic

$$\mathbf{v} = \mathbf{K} \cdot \mathbf{e} \tag{10}$$

where gain matrix $\mathbf{K}$ has dimensions 2×3. The schematic of the obtained control is explained in Fig. 3.

The controller structure can be derived from observing Fig. 2. To reduce error in the driving direction e_1 the tangential robot velocity should be changed correspondingly. Similarly, the orientation error e_3 can be efficiently manipulated by robot angular speed. Finally, the error orthogonal to the driving direction could be reduced changing the angular velocity. At the same time also the robot drive direction (forward or backward) should be considered. From the above conclusions the control law is

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} -k_1 & 0 & 0 \\ 0 & -sign(u_{r1})k_2 & -k_3 \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \tag{11}$$

The next question is how to determine the suitable controller gains. This can always be done by trying or by optimization of some cost function. In this paper the controller gains are determined by comparing the real and the desired characteristic polynoms. The polynomial takes the following form

$$(s + 2\zeta\omega_n)(s^2 + 2\zeta\omega_n s + \omega_n^2) \tag{12}$$

Similarly as for the second order systems the desired dumping coefficient $\zeta \in (0,1)$ and the characteristic frequency $\omega_n > 0$, are selected. An extra pole at $s = -2\zeta\omega_n$ increases the rising time and decreases the system overshot. A characteristic polynomial of a closed loop with a state space controller (11) is

$$\det(s\mathbf{I} - \mathbf{A} + \mathbf{BK}) = s^3 + (k_1 + k_3)s^2 + \\ + (k_1 k_3 + k_2 u_{r1} + u_{r2}^2)s + k_1 k_2 e u_{r1} + k_3 u_{r2}^2 \tag{13}$$

Comparing coefficients at the same power of s in Eqs. (12) and (13) result

$$\begin{aligned} k_1 + k_3 &= 4\zeta\omega_n \\ k_1 k_3 + k_2 u_{r1} + u_{r2}^2 &= 4\zeta^2\omega_n^2 + \omega_n^2 \\ k_1 k_2 u_{r1} + k_3 u_{r2}^2 &= 2\zeta\omega_n^3 \end{aligned} \tag{14}$$

Further on let us find the solution of equation (14) in the form suggested in [7]

$$k_1 = k_3 = 2\zeta\omega_n \tag{15}$$

k_2 is then determined as

$$k_2 = \frac{\omega_n^2 - u_{r2}^2}{|u_{r1}|} \tag{16}$$

The obtained system poles are constant

$$\begin{aligned} s_1 &= -2\zeta\omega_n \\ s_2 &= -\zeta\omega_n + \omega_n\sqrt{(\zeta^2 - 1)} \\ s_3 &= -\zeta\omega_n - \omega_n\sqrt{(\zeta^2 - 1)} \end{aligned} \tag{17}$$

and ω_n should be larger than the maximum allowed robot angular velocity, $\omega_n \geq u_{r2MAX}$. When u_{r1} is close to zero, k_2 goes to infinity and therefore a gain scheduling [6] should be chosen for k_2 as $k_2 = g \cdot |u_{r1}(t)|$. System characteristic frequency becomes

$$\omega_n(t) = \sqrt{u_{r2}^2(t) + gu_{r1}^2(t)} \tag{18}$$

and the controller gains are

$$\begin{aligned} k_1 &= k_3 = 2\zeta\omega_n(t) \\ k_2 &= g \cdot |u_{r1}(t)| \end{aligned} \tag{19}$$

The parameter $g > 0$ gives an additional freedom in controller design [7]. The controller gains approaches zero when the robot stops at which time the robot is not controllable. If the controller gains are chosen from Eqs. (15) and (16) a nonlinear, time varying controller is obtained which gives stable and constant poles of the closed-loop system. Although the poles are stable and constant, the local asymptotic stability is not guaranteed as system is still time varying ([7] [8]). System stability is therefore checked by Lyapunov stability analysis discussed bellow.

IV. STABILITY ANALYSIS

For the presented controller (7) and (11), the system stability in the sense of Lyapunov is shown next. Lyapunov energy function $V(\mathbf{q})$ should be positive definite and continuous with continuous derivatives on all states $\mathbf{q}$. In the equilibrium point $\mathbf{q}=\mathbf{0}$ the energy function must be $V(\mathbf{q}) = 0$.

The origin point $\mathbf{q}=\mathbf{0}$ is stable equilibrium point if there exists such a energy function $V(\mathbf{q})$ that $\dot{V}(\mathbf{q})$ is negative semidefinite over all the trajectory. If $\dot{V}(\mathbf{q})$ is negative definite, then the equilibrium point is asymptotically stable.

If the controller from (11) is rewritten considering (19), discontinuous part $k_2 sign(u_{r1})e_2$ is replaced by $\bar{k}_2 u_{r1} e_2$ where $\bar{k}_2$ is a positive constant ($\bar{k}_2 > 0$). The controller becomes

$$\begin{aligned} v_1 &= -k_1 e_1 \\ v_2 &= -\bar{k}_2 u_{r1} e_2 - k_3 e_3 \end{aligned} \tag{20}$$

Defining Lyapunov function as

$$V(\mathbf{e}) = \frac{1}{2}\left(e_1^2 + e_2^2 + e_3^2\right) \tag{21}$$

and its derivative as

$$\dot{V}(\mathbf{e}) = e_1\dot{e}_1 + e_2\dot{e}_2 + e_3\dot{e}_3 \tag{22}$$

Considering model (9) and controller (20) the following relation is obtained

$$\dot{V}(\mathbf{e}) = -e_1^2 k_1 + e_2 e_3 u_{r1} - e_2 e_3 u_{r1} \bar{k}_2 - e_3^2 k_3 \qquad (23)$$

from where the stability (negative semidefiniteness) cannot be concluded. A more appropriate energy function reads

$$V(\mathbf{e}) = \frac{\bar{k}_2}{2} \left(e_1^2 + e_2^2 \right) + \frac{e_3^2}{2} \qquad (24)$$

and its first derivative is

$$\dot{V}(\mathbf{e}) = -k_1 \bar{k}_2 e_1^2 - k_3 e_3^2 \qquad (25)$$

which is negative semidefinite function (note that it is not negative definite with respect to e_2) and the origin $\mathbf{e} = \mathbf{0}$ is stable. Asymptotic stability is therefore not shown because $\dot{V}(\mathbf{e})$ is not negative definite. To check the asymptotic stability Barbalat's lemma is used. For energy function $V(t)$ holds

$$V(t) - V(0) = \int_o^t \dot{V}(\tau) d\tau =$$
$$= -k_1 \bar{k}_2 \int_0^t e_1^2(\tau) d\tau - k_3 \int_0^t e_3^2(\tau) d\tau \qquad (26)$$

The limit of the above integral as t tends to infinity is

$$\lim_{t \to \infty} \left(k_1 \bar{k}_2 \int_0^t e_1^2(\tau) d\tau + k_3 \int_0^t e_3^2(\tau) d\tau \right) \le V(0) < \infty \qquad (27)$$

The limit (27) exists and is a finite number; therefore $e_1, e_3 \in L_2$. Supposing the reference trajectory is smooth continuous function with bounded continuous derivatives then from (9) follows that u_{r1} and u_{r2} as well as $\dot{e}_1$, $\dot{e}_2$ and $\dot{e}_3$ are bounded. From $e_1, e_3 \in L_2$ and bounded derivatives $\dot{e}_1$, $\dot{e}_2$ and $\dot{e}_3$ (using Barbalat's lemma) follows that $e_1 \to 0$ and $e_3 \to 0$ when $t \to \infty$. The derived control law from Eq. (20) forces e_1 and e_3 to zero while this is not guaranteed for e_2. An example of such a situation is when the robot is in parallel with the reference line trajectory, $e_1 = e_3 = 0$, $e_2 \ne 0$ and the reference tangential speed $u_{r1}=0$. In this situation the controller (20) will not force e_2 to zero. However, since the $\dot{V}(\mathbf{e})$ is negative semidefinite, the error e_2 is always bounded. When the trajectory is designed so that $u_{r1}(t)$ is nonzero the controller (20) tries to lower the error e_2.

V. CONTROL EXAMPLE

As already concluded the open-loop control only (feed-forward) does not give satisfactory results because it is not robust to the initial state errors, noise and other disturbances during the operation. For successful control the closed loop must be implemented. In the sequel examples of closed-loop control of simulated and real robot according to the schematic in Fig. 3 are shown.

A. Simulated Robot

Fig. 4 shows simulated results of closed-loop control where robot has initial states error (it does not start with the correct orientation and position) and the robot inputs and outputs are contaminated by noise at inputs and at outputs.
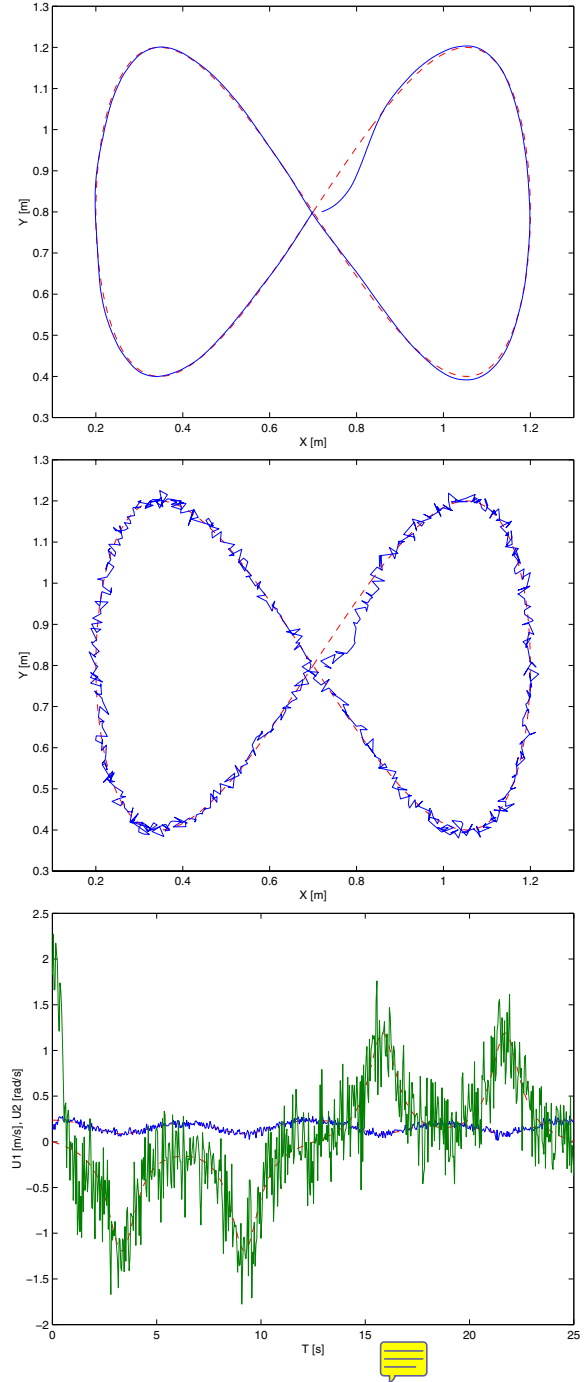

Fig. 4. Closed-loop control ($\zeta = 0.6$, $\omega_n = 2$) of a simulated robot (reference: - -, robot path: -). The robot path, outputs with noise and inputs (feedforward: - -, feedforward and closed-loop: -).

In Fig. 4 the robot follows the reference path despite noise and initial state error in robot orientation. The controller successfully suppresses the initial state error and follows

the reference path with satisfactory accuracy.

B. Real Robot

Results of real robot closed-loop control are shown in Fig. 5. The inputs to the robot are linear (u_1) and angular velocity (u_2). The robot is driven by two powerful DC motors, one for each wheel, where on board logic calculates reference wheel speeds (from u_1 and u_2) and by the use of signals obtained from encoders and implemented PID control with anti-wind up protection assures the robot to achieve requested input speeds (u_1 and u_2).
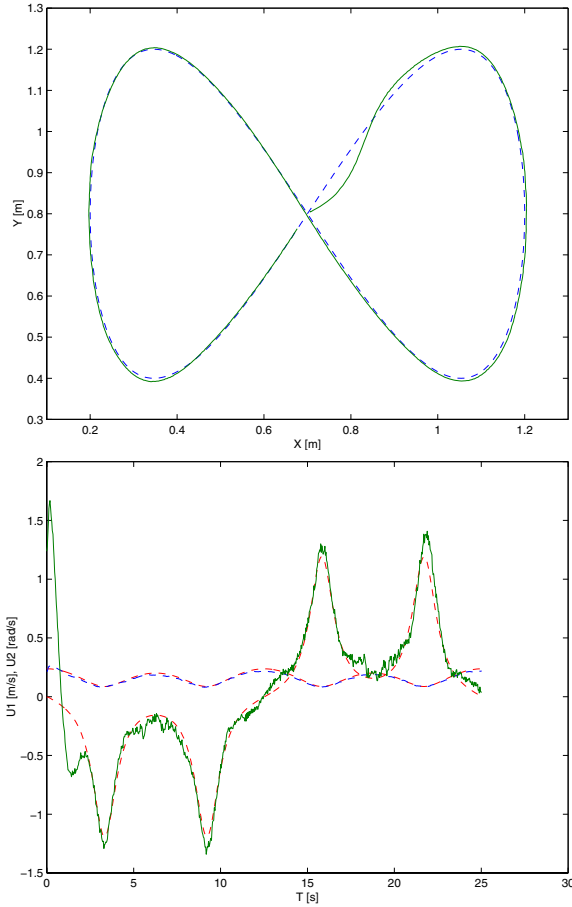

Fig. 5. Closed-loop control ($\zeta = 0.6$, $\omega_n = 2$) of real robot (reference: - -, robot path: -). The robot path and inputs (feedforward: - -, feedforward and closed-loop: -).

In Fig. 5 the robot follows the reference path despite noise and initial state errors. Smaller errors during following following are result of disturbances at command reception, quantization, sensors noise (wheel encoders, camera) and because controller in not asymptotically stable with respect to e_2.

VI. TRAJECTORY PLANNING

The above part of the paper deals with robot control on already known path where motion planning is not considered. However, motion planning, i.e. how the robot path is obtained, is a very important task in mobile robotics.
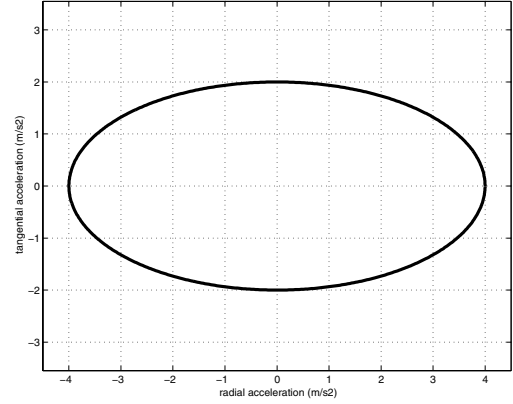

Fig. 6. Acceleration limits

A short introduction to time optimal path planning based on spline curves is given in the sequel. A detail account of the presented algorithm can be found in [5].

If pure rolling condition of the robot wheels is supposed than the robot real velocities suit command velocities according to relation (1). This condition can be achieved by limitation of the allowed acceleration which is decomposed to the tangential acceleration a_{tang} and to the radial acceleration a_{rad}.

$$a_{tang}(t) = \frac{dv(t)}{dt}$$
$$a_{rad}(t) = v(t) \times \omega(t) = v(t)^2 \kappa(t) \qquad (28)$$

where $\kappa(t)$ is curvature defined by Eq. 4. Since a_{tang} and a_{rad} are orthogonal the overall acceleration is

$$a = \sqrt{a_{tang}^2 + a_{rad}^2} \qquad (29)$$

The overall acceleration is limited with the friction force. Because the gravity centre of the robot is on certain height above ground level, the limit of the tangential acceleration differs from the limit of the radial acceleration When accelerating in linear direction, a part of the robot weight is carried by rear or front slider. This means that the wheels of the robot press on the ground with the force smaller than the robot gravity force so the limit for the tangential acceleration is therefore lower than the limit for radial acceleration.

In our case the acceleration limits were measured (for more details see [5]. Maximum tangential acceleration was 2 m/s^2 and maximum radial acceleration 4 m/s^2. The overall acceleration should be somewhere inside the ellipse as shown in Fig. 6. The real robot follows the prescribed path if it does not slide, i.e. its accelerations must be within the limits given in Fig. 6. If control is time-optimal, then the acceleration is always on the ellipse given in Fig. 6.

The reference path is obtained using spline curves. The two-dimensional curve is obtained by combining two splines, $x(t)$ and $y(t)$. The spline curve is then fully determined by a start point (SP), end an point (EP) and knots

(points connecting spline segments - cubic polinomials). Further information on spline design for this particular case can be found in [5].

Time-optimal path planning requires robots to drive with high speed and therefore a smooth path is necessary. The measure of path smoothness is presented by the curvature $\kappa(t)$ (Eq. (4)).

To find the time-minimum path, an optimisation algorithm is used which places the knots to the optimal positions (most of the problems can be solved with up to 3 knots).

As already mentioned, the criterion of optimality is time needed for the path. It is calculated in each iteration of optimisation by integration of velocity profile:

$$t_{crit} = \int_{SP}^{EP} \frac{1}{v(s)} \cdot ds \qquad (30)$$

The velocity profile is obtained using the following algorithm:

- Path curvature $\kappa(t)$ is calculated according to equation Eq. (4).
- 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 allowed speed due to radial acceleration limit $v(t) = \sqrt{a_{radMAX}/\kappa(t)}$. Its tangential acceleration a_{tang} must be 0.
- Before and after a TP, the robot can move faster, because the curve radius gets bigger than in the 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 has the shape of "U" (or "V"). Similarly the maximum velocity profile (due to tangential acceleration/deceleration) is determined for initial (SP) and final (EP) (if required) velocity respectively.
- The highest allowable overall velocity profile is determined as the minimum of all the velocity profiles.
- The initial and final (if required) velocities must be on the highest allowable overall velocity profile. If not, the given path cannot be driven without violating the acceleration constraints.
- For the highest given allowable velocity profile the cost function is calculated according to Eq. (30).

A typical example of the highest overall velocity profile determination including start and end point limitations is seen in Fig. 7.

There are five curves each belonging to one of the five points: start and end point and three turning points (TP) corresponding to the local curvature maximums along the current curve.

VII. Conclusion

The control of a mobile robot along the reference curve is presented where the controller consists of from two
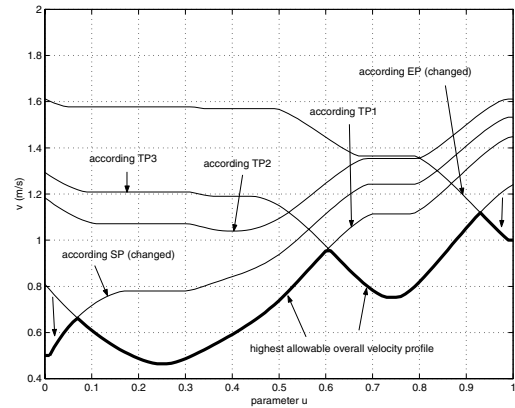

Fig. 7. Highest allowable velocity profile including start and end point

parts: the feedforward control and the closed-loop control. The former uses robot inverse kinematics to calculate the feedforward inputs from the reference curve, while the latter cancels the effects of noise, disturbances and initial state errors. The presented example as well as the real applications prove that a mobile robot can follow the desired reference path according to the prescribed velocity profile with a satisfactory accuracy. A motion planning optimisation algorithm that produces time optimal path for a given robot is also presented. The optimisation is time consuming and therefore in the future applications a solution with the predefined optimal paths for a grid of starting positions will be used instead.

References

[1] A. Balluchi, A. Bicchi, A. Balestrino and G. Casalino, "Path Tracking Control for Dubin's Cars," in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, 1996, pp. 3123-3128.

[2] C. Canudas de Wit and O. J. Sordalen, "Exponential Stabilization of Mobile Robots with Nonholonomic Constraints," *IEEE Transactions on Automatic Control*, vol. 37, no. 11, pp. 1791-1797, 1992.

[3] I. Kolmanovsky and N. H. McClamroch, "Developments in Nonholonomic Control Problems," *IEEE Control Systems*, vol. 15, no. 6, pp. 20-36, 1995.

[4] J. P. Laumond, *Robot Motion Planning and Control*. Lecture Notes in Control and Information Science, Springer-Verlag, 1998, vol. 229.

[5] M. Lepetič, G. Klančar, I. Škrjanc, D. Matko and B. Potočnik, "Time optimal planning considering acceleration limits," *Robotics and Autonomous Systems*, vol. 45, pp. 199-210, 2003.

[6] A. Luca and G. Oriolo, "Modelling and control of nonholonomic mechanical systems," in *Kinematics and Dynamics of Multi-Body Systems*, J. Angeles and A. Kecskemethy, Eds. Wien: Springer-Verlag, 1995.

[7] A. Luca, G. Oriolo and M. Vendittelli, "Control of wheeled mobile robots: An experimental overview," in *RAMSETE - Articulated and Mobile Robotics for Services and Technologies*, S. Nicosia, B. Siciliano, A. Bicchi and P. Valigi, Eds. Springer-Verlag, 2001.

[8] G. Oriolo, A. Luca and M. Vandittelli, "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.

[9] N. Sarkar, X. Yun and V. Kumar, "Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robot," *The International Journal of Robotic Research*, vol. 13, no. 1, pp. 55-69, 1994.