MOBILE ROBOT PREDICTIVE TRAJECTORY TRACKING

Martin Seyr

Vienna University of Technology, Institute of Mechanics and Mechatronics
Gusshausstrasse 27-29, A-1040 Wien
martin.seyr@tuwien.ac.at

Stefan Jakubek

Vienna University of Technology, Institute of Mechanics and Mechatronics
Gusshausstrasse 27-29, A-1040 Wien
stefan.jakubek@tuwien.ac.at

Abstract: For a two-wheeled differentially driven mobile robot a trajectory tracking concept is developed. A trajectory is a time-indexed path in the plane, i.e. in the three-dimensional configuration space consisting of position and orientation. Due to the nonholonomic nature of a rolling wheel, the system cannot be stabilized by a continuous time-invariant feedback or by feedback linearization. A novel approach taken in this paper to solve the nonholonomic control problem consists of nonlinear predictive control in conjunction with linear state space control with integration of the control error. Based on a Gauss-Newton algorithm, predicted future position errors are minimized by numerical computation of an optimal sequence of control inputs using prespecified shape functions.

1 INTRODUCTION

The basic task in mobile robot motion control is to accurately follow a given trajectory. The error between the present posture $x(t) = [x(t)\ y(t)\ \varphi(t)]^T$ and the reference trajectory is to be minimized.

Considerable research has been done on trajectory tracking control of the unicycle-type mobile robot. Its kinematics are a classical example of a nonholonomic nonlinear control system, the nonholonomic integrator (NHI), in somewhat different form also known as Brockett- or Heisenberg-system.

It was first shown by (Brockett, 1983), that this system cannot be stabilized by continuous, time-invariant feedback, although it is controllable in a nonlinear sense.

Furthermore, it can be shown using a methodology by (Isidori, 1989), that the NHI cannot be feedback-linearized.

Therefore, various control concepts trying to circumvent the aforementioned limitations have been presented in recent years. Among the major groups of approaches are sliding-mode control, e.g. (Bloch and Drakunov, 1994), time-varying feedback laws, e.g. (Samson, 1995), hybrid control laws, e.g. (Hespanha and Morse, 1996) and dynamic feedback linearization, e.g. (Oriolo et al., 2002).

None of the mentioned publications deal with the problem of non-zero side-slip angle. Some do not take the dynamics of the system into account, they are only concerned with its kinematics.

A drawback inherent to many of the concepts present in the literature is a singularity in the control law occurring at zero velocity, e.g. (Oriolo et al., 2002).

In the present paper, a novel approach is presented employing numerical optimization of open loop control rather than any explicit feedback control law. Therefore, the aforementioned restrictions do not apply here.

This concept is made possible by the robot's outstanding on-board calculation capacity provided by a microcontroller and a digital signal processor.

2 THE PLANT: AN AUTONOMOUS TWO-WHEELED MOBILE ROBOT

The robot, Fig. 1, has two wheels with rubber tires and two felt shoes, one at the front and one at the rear to stabilize it around the pitch axis. It fits into a cuboid

with a 0.075m square footprint. The two wheels are supported by ball bearings and powered by two individual DC-motors. A microcontroller produces two pulse-width modulated (PWM) constant voltage signals, which are amplified by a dual full bridge driver. The amplified signals drive the two DC-motors.
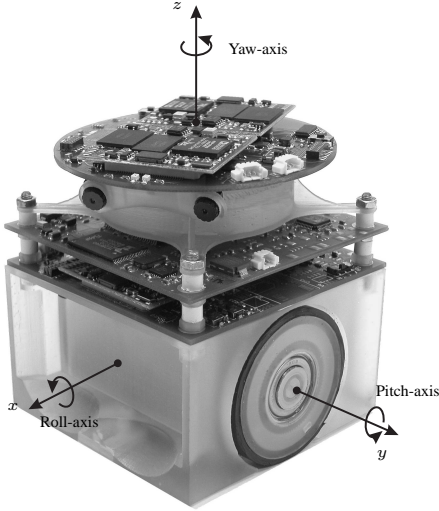

Figure 1: Autonomous mini-robot Tinyphoon, http://www.tinyphoon.com

Nonlinearities originate from variable switching times and dead zones in the amplifier circuit, friction characteristics of bearings and gearboxes and the wheel slip dynamics. The kinematics of a two-wheeled mobile robot are equivalent to those of a single rolling wheel. Therefore, this system is often referred to as the unicycle-type mobile robot.

The dynamics of the entire system will be partitioned into those of the path velocity and yaw angular velocity, which can be linearized to form a two-input-two-output linear state space system, and the nonlinear nonholonomic kinematics.

2.1 Velocity dynamics

To obtain a linearization of the velocity dynamics, a number of simplifications are made:

1. The slip-dynamics are omitted,

2. the side-slip velocity v_n and the tangential velocity v_t are combined to an effective track speed v, Fig. 2,

$$v = \sqrt{v_t^2 + v_n^2} \, \mathrm{sign} v_t = \frac{v_t}{\cos \alpha}, \qquad (1)$$

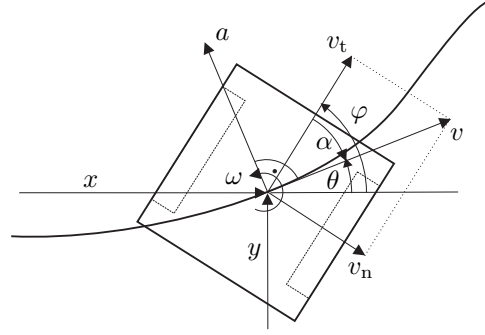3. the motor characteristic is linearized using least squares with a bilinear regressor function.

Figure 2: Kinematics of a two-wheeled mobile robot under consideration of side slip and tangential wheel slip

The result is a linear continuous-time state space system with state vector $\boldsymbol{v} := [v \; \omega]^T$ and input vector $\boldsymbol{u} := [r_{\mathrm{PWM,r}} \; r_{\mathrm{PWM,l}}]^T$,

$$\dot{\boldsymbol{v}} = \left[\begin{array}{cc} a & 0 \\ 0 & c \end{array} \right] \boldsymbol{v} + \left[\begin{array}{cc} b & b \\ d & -d \end{array} \right] \boldsymbol{u}, \qquad (2)$$

where the parameters have been collected to form the constants a, b, c and d, and $r_{\mathrm{PWM,r}}$ and $r_{\mathrm{PWM,l}}$ denote the PWM-duty cycles of the right and left motor input voltages.

2.2 Kinematics

The kinematics of the unicycle-type mobile robot are given by

$$\begin{array}{rcl} \dot{x} & = & v \cos(\varphi + \alpha) \\ \dot{y} & = & v \sin(\varphi + \alpha) \\ \dot{\varphi} & = & \omega, \end{array} \qquad (3)$$

where x and y denote the inertial coordinates and φ denotes the inertial attitude angle of the robot.

From Fig. 2 it can be seen that

$$\theta = \varphi + \alpha, \qquad (4)$$

where θ denotes the angle of the tangent to the actual path.

3 CONTROL CONCEPT

Since the velocity dynamics and the kinematics can be solved consecutively (i.e. the result of the velocity dynamics serves as input into the kinematics, but there is no influence of the positions on the velocities), the application of a cascading control scheme is straightforward.

The chosen scheme employs a linear state feedback

law with integration of the velocity errors to control the velocities v, in the following referred to as the inner loop, and a nonlinear predictive controller using the closed loop dynamics of the inner loop in the prediction of the positions x (in the following: outer loop), Fig. 3.
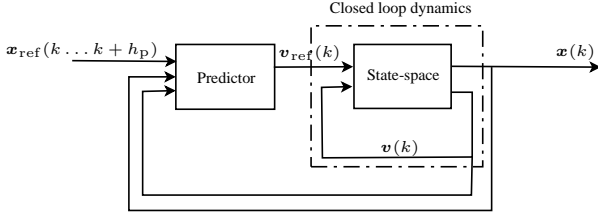

Figure 3: Cascaded control scheme, h_p denotes the prediction horizon

4 CONTROL OF THE INNER LOOP

4.1 Discretization

First, the continuous-time state space representation of the robot's dynamics is discretized assuming that a zero-order hold acts at the input. Theoretical background can be found e.g. in (Isermann, 1987).
The system (2) is then written as

$$v(k+1) = \underbrace{\begin{bmatrix} p & 0 \\ 0 & r \end{bmatrix}}_{A} v(k) + \underbrace{\begin{bmatrix} q & q \\ s & -s \end{bmatrix}}_{B} u(k), \quad (5)$$

where p, q, r, s are real-valued constants and k denotes the integer sampling instants.

4.2 Extended system

The control algorithm's calculation time must be expected to consume a considerable portion of the sampling interval, therefore the common assumption, that the output of a control algorithm is available instantaneously, does not hold.
On the contrary, it must be assumed, that the output of the controller can only be applied at the beginning of the next sampling interval. Basically, this would lead to a control law of the form

$$u(k) = f(v(k-1), v_\mathrm{ref}(k-1)), \quad (6)$$

i.e. a one-step deadtime in the control law itself unless a prediction for $v(k)$ is used.

Therefore, the order of the system is deliberately increased by using past values as additional states, so as to be able to use $v(k-1)$ in the control law for $u(k)$.

In the outer loop, a prediction for the future velocities is calculated anyway (indicated with a ˆ). Practical tests show, however, that it is still advisable to use the extended system for controller design, since a feedback law based entirely on predicted values tends to destabilize quickly.

After introducing the integrated control errors q as additional states, the extended system is written as

$$\begin{bmatrix} v(k+1) \\ v(k) \\ q(k) \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ I & 0 & 0 \\ 0 & -I & I \end{bmatrix} \begin{bmatrix} \hat{v}(k) \\ v(k-1) \\ q(k-1) \end{bmatrix} +$$
$$+ \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} v_\mathrm{ref}(k-1). \quad (7)$$

The static state feedback control law has the form

$$u(k) = \underbrace{K}_{2\times 6} \underbrace{\begin{bmatrix} \hat{e}(k) \\ e(k-1) \\ -q(k-1) \end{bmatrix}}_{6\times 1}, \quad (8)$$

where K denotes the feedback gain matrix, calculated using standard pole assignment procedures.

5 CONTROL OF THE OUTER LOOP

The principle of nonlinear predictive control using a Gauss-Newton-optimization algorithm is taken from (Norgaard et al., 1999), where this procedure is applied to SISO-systems (single-input-single-output), which are dynamically modeled by recurrent multi-layer perceptron networks (MLP).
The algorithm is adapted to use a nonlinear MIMO-state space representation instead of the MLP-network.
A similar procedure for a MIMO-neural network was used in (Seyr and Jakubek, 2005).

5.1 Discretization of the kinematic model

First, the continuous time nonlinear state space system (3) is discretized using a simple forwards difference approximation (explicit Euler) for the first order derivatives,

$$\dot{x}(k) = \frac{x(k+1) - x(k)}{T_{\mathrm{s}}},$$

analogously for y and φ.

Then the system (3) can be written as

$$
\begin{aligned}
x(k+1) &= T_{\mathrm{s}}v(k)\cos[\varphi(k) + \alpha(k)] + x(k) \\
y(k+1) &= T_{\mathrm{s}}v(k)\sin[\varphi(k) + \alpha(k)] + y(k) \\
\varphi(k+1) &= T_{\mathrm{s}}\omega(k) + \varphi(k).
\end{aligned}
\tag{9}
$$

5.2 Cost function

Predictive control is based on the minimization of a scalar quadratic cost function containing predicted future position errors P and future control variables V_{ref} during every sampling interval, here denoted for a system with three outputs ($x := [x\ y\ \varphi]^T$) and two control variables ($v_{\mathrm{ref}} := [v_{\mathrm{ref}}\ \omega_{\mathrm{ref}}]^T$). For the two control variables, shape functions are chosen. The shape of the future control variables is then adjusted using two parameters each. The cost as a function of the form-parameters c reads

$$V = \frac{1}{2h_{\mathrm{p}}}P^T L P + \frac{1}{2}c^T R c, \tag{10}$$

where

$$
P = \begin{bmatrix}
x_{\mathrm{ref}}(k+3) - x(k+3) \\
x_{\mathrm{ref}}(k+4) - x(k+4) \\
\vdots \\
y_{\mathrm{ref}}(k+3) - y(k+3) \\
\vdots \\
y_{\mathrm{ref}}(k+h_{\mathrm{p}}) - y(k+h_{\mathrm{p}}) \\
\vdots \\
\varphi_{\mathrm{ref}}(k+h_{\mathrm{p}}) - \varphi(k+h_{\mathrm{p}})
\end{bmatrix}.
$$

The structure of the shape functions for v_{ref} and ω_{ref} is identical and given by

$$v, \omega_{\mathrm{ref}}(k) = c_{1,3} + c_{2,4}\underbrace{(1 - \exp(-\kappa k T_{\mathrm{s}}))}_{f(k)}, \tag{11}$$

where the curvature of the shape function can be adjusted by the form factor κ: from almost linear ($\kappa \ll$) to steep at the beginning and flat at the end ($\kappa \gg$), which influences the bandwidth of the system.

Additionally, the second part $f(k)$ is scaled to ensure comparable influence of $c_{2,4}$ when using different values for κ.

- The cost function V is evaluated and minimized during each sampling interval.

- Future reference values x_{ref} up to the prediction horizon h_{p} must be known.

- The form-parameters c, shaping the v_{ref} up to the control horizon h_{u} (here: $h_{\mathrm{u}} = h_{\mathrm{p}} - 2$), are optimized using (10).

- The weight matrices $L \in \mathbb{R}^{3(h_{\mathrm{p}}-2)\times 3(h_{\mathrm{p}}-2)}$ and $R \in \mathbb{R}^{4\times 4}$ determine to what extent the future control variables and the future control errors are considered.

From measurement data, the current posture $[x(k)\ y(k)\ \varphi(k)]^T)$ and velocities $[v(k)\ \omega(k)]^T$ are calculated.

Using the first and the last line of the closed loop state space representation of the inner loop

$$v(k+1) =$$

$$= \begin{bmatrix} A - BK_1 & -BK_2 & -BK_{\mathrm{I}} \end{bmatrix} \begin{bmatrix} v(k) \\ v(k-1) \\ q(k-1) \end{bmatrix} +$$

$$+ \begin{bmatrix} BK_1 & BK_2 \end{bmatrix} \begin{bmatrix} v_{\mathrm{ref}}(k) \\ v_{\mathrm{ref}}(k-1) \end{bmatrix},$$

and

$$
\begin{aligned}
q(k) &= \begin{bmatrix} 0 & -I & I \end{bmatrix} \begin{bmatrix} v(k) \\ v(k-1) \\ q(k-1) \end{bmatrix} + \\
&\quad + \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} v_{\mathrm{ref}}(k) \\ v_{\mathrm{ref}}(k-1) \end{bmatrix},
\end{aligned}
\tag{12}
$$

estimated future velocities and velocity error integrals can be calculated recursively.

Next, the predicted positions are calculated by (9). Therefore, the current side-slip angle $\alpha(k)$ has to be determined. The current position $x_k = [x_k\ y_k]$ and the last two positions x_{k-1} and x_{k-2} are transformed to local coordinates, Fig. 4,

$$x_{k-i,\mathrm{loc}} = \begin{bmatrix} \cos\theta_{k-1} & \sin\theta_{k-1} \\ -\sin\theta_{k-1} & \cos\theta_{k-1} \end{bmatrix}[x_{k-i} - x_{k-1}], \tag{13}$$

where $i \in [0\ 1\ 2]$. Next, the parameters of a Spline-approximation of $x(k-i)$ and $y(k-i)$ are calculated using a quadratic regressor $[1\ t\ t^2]$ and dimensionless time $t \in [0; 2]$ according to

$$
\begin{bmatrix} a_{x,0} \\ a_{x,1} \\ a_{x,2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix}^{-1} \begin{bmatrix} x_{k-2} \\ x_{k-1} \\ x_k \end{bmatrix}.
\tag{14}
$$

The increment of the path angle $\Delta\theta$ is calculated by evaluating the derivatives of the Spline-approximations at time $t = 2$,
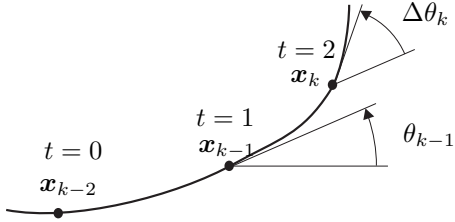
Figure 4: Estimation of the current path angle θ_k

$$\Delta\theta_k = \operatorname{atan}\frac{\mathrm{d}y}{\mathrm{d}x} = \operatorname{atan}\frac{\mathrm{d}x/\mathrm{d}t}{\mathrm{d}y/\mathrm{d}t} = \operatorname{atan}\frac{a_{y,1} + 4a_{y,2}}{a_{x,1} + 4a_{x,2}}. \tag{15}$$

Finally, the actual side-slip angle is given by

$$\alpha_k = \theta_k - \varphi_k = \theta_{k-1} + \Delta\theta_k - \varphi_k. \tag{16}$$

The incremental calculation of the path angle has the decisive advantage that possible transgressions of the interval $[-\pi; \pi]$ do not have to be accounted for.

Naturally, no meaningful results for the path angle are obtained for zero track speed v. Therefore, the path angle is set equal to the attitude angle for small values of v below a certain margin, followed by a short region of linear interpolation to ensure continuity and finally taken full beyond another threshold of v.

When using possibly noise corrupted measurement data, the estimation can also be performed using a greater number of previous positions, then employing a least squares estimator.

To control the inertial attitude angle, the current side-slip α is subtracted from the reference path angle θ_{ref}, thus obtaining a feasible reference attitude φ_{ref}.

After the computation of an initial estimate of the future positions x, under the assumption that the side-slip angle α remains constant over the prediction horizon, the position errors $p = x_{\mathrm{ref}} - x$ are calculated and concatenated in P.

5.3 Minimization

The position errors are now approximated in a first order Taylor series expansion,

$$\boldsymbol{P} \doteq \boldsymbol{P}_0 + \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{V}_{\mathrm{ref}}} \frac{\partial \boldsymbol{V}_{\mathrm{ref}}}{\partial \boldsymbol{c}} \boldsymbol{c} =$$

$$\boldsymbol{P}_0 - \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{V}_{\mathrm{ref}}} \frac{\partial \boldsymbol{V}_{\mathrm{ref}}}{\partial \boldsymbol{c}} \boldsymbol{c} := \boldsymbol{P}_0 - \boldsymbol{D}_X \boldsymbol{D}_V \boldsymbol{c}. \tag{17}$$

The matrix $\boldsymbol{D}_X \in \mathbb{R}^{3(h_{\mathrm{p}}-2) \times 2(h_{\mathrm{p}}-2)}$ can be written as

$$\boldsymbol{D}_X = \begin{bmatrix} \dfrac{\partial x(k+2+i)}{\partial v_{\mathrm{ref}}(k+j)} & \dfrac{\partial x(k+2+i)}{\partial \omega_{\mathrm{ref}}(k+j)} \\[2ex] \dfrac{\partial y(k+2+i)}{\partial v_{\mathrm{ref}}(k+j)} & \dfrac{\partial y(k+2+i)}{\partial \omega_{\mathrm{ref}}(k+j)} \\[2ex] \dfrac{\partial \varphi(k+2+i)}{\partial v_{\mathrm{ref}}(k+j)} & \dfrac{\partial \varphi(k+2+i)}{\partial \omega_{\mathrm{ref}}(k+j)} \end{bmatrix}, \tag{18}$$

with $i, j \in [1; h_{\mathrm{p}} - 2]$, and $\boldsymbol{D}_V \in \mathbb{R}^{2(h_{\mathrm{p}}-2) \times 4}$ reads

$$\boldsymbol{D}_V = \begin{bmatrix} 1 & f(1) & 0 & 0 \\ 1 & f(2) & 0 & 0 \\ \vdots & & \vdots & \\ 0 & 0 & 1 & f(1) \\ 0 & 0 & 1 & f(2) \\ \vdots & & \vdots & \end{bmatrix} \tag{19}$$

The total derivatives of the positions with respect to the reference velocities are calculated recursively. The total derivatives of the velocities $\boldsymbol{V}$ and the velocity error integrals $\boldsymbol{Q}$ (where $\boldsymbol{Q}$ is the vector of velocity error integrals concatenated of $\boldsymbol{q}$ in the exact same way as $\boldsymbol{P}$) with respect to the reference velocities $\boldsymbol{V}_{\mathrm{ref}}$ are needed during the calculation of the total derivatives of the positions $\boldsymbol{X}$.
The dependencies are

$$\boldsymbol{x}(k+2+i) = \boldsymbol{f}_1(\boldsymbol{x}(k+1+i), \boldsymbol{v}(k+1+i))$$

$$\boldsymbol{v}(k+1+i) = \boldsymbol{f}_2(\boldsymbol{v}(k+i), \boldsymbol{v}(k-1+i), \boldsymbol{q}(k-1+i),$$
$$\boldsymbol{v}_{\mathrm{ref}}(k+i), \boldsymbol{v}_{\mathrm{ref}}(k-1+i))$$

$$\boldsymbol{q}(k+i) = \boldsymbol{f}_3(\boldsymbol{v}(k-1+i), \boldsymbol{q}(k-1+i), \boldsymbol{v}_{\mathrm{ref}}(k-1+i)),$$

where the functions $\boldsymbol{f}_1$ through $\boldsymbol{f}_3$ are given by (9) and (12).

The cost function (10) now reads

$$V(\boldsymbol{c}) = \frac{1}{2}\boldsymbol{c}^T \boldsymbol{R} \boldsymbol{c} + \tag{20}$$

$$+ \frac{1}{2h_{\mathrm{p}}}(\boldsymbol{P}_0 - \boldsymbol{D}_X \boldsymbol{D}_V \boldsymbol{c})^T \boldsymbol{L}(\boldsymbol{P}_0 - \boldsymbol{D}_X \boldsymbol{D}_V \boldsymbol{c}).$$

To ensure closed loop stability of the inertial angle, which proved to be critical during testing, a terminal constraint for the inertial angle is introduced, (Mayne et al., 2000) and references therein.
To fulfill the terminal constraint, an additional term

with a Lagrange-multiplier is added to the cost function. The additional term reads

$$\lambda \left(\varphi(k + h_{\mathrm{p}}; \boldsymbol{c}) - \varphi_{\mathrm{ref}}(k + h_{\mathrm{p}}) \right) \doteq$$
$$\lambda \underbrace{\left(\varphi_0(k + h_{\mathrm{p}}) - \varphi_{\mathrm{ref}}(k + h_{\mathrm{p}}) \right)}_{\Delta\varphi_{\mathrm{end}}} + \lambda \boldsymbol{D}_\varphi \boldsymbol{c}, \quad (21)$$

where the 1×4 row vector $\boldsymbol{D}_\varphi$ is the last row of $\boldsymbol{D}_X \boldsymbol{D}_V$, i.e. the derivatives of $\varphi(k + h_{\mathrm{p}})$ with respect to the form-parameters, and λ is the Lagrange-multiplier. The second term in (21) is a first order Taylor approximation of an otherwise nonlinear constraint.

Minimization under fulfilment of the terminal constraint is then obtained by differentiating with respect to $\boldsymbol{c}$ and λ and equating the derivative with zero.

After some algebraic manipulations the linear system of equations with dimension 5

$$\begin{bmatrix} \frac{1}{h_{\mathrm{p}}} \boldsymbol{D}_V^T \boldsymbol{D}_X^{(\nu)T} \boldsymbol{L} \boldsymbol{D}_X^{(\nu)} \boldsymbol{D}_V + \boldsymbol{R} & \boldsymbol{D}_\varphi^{(\nu)T} \\ \boldsymbol{D}_\varphi^{(\nu)} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{c}^{(\nu)} \\ \lambda^{(\nu)} \end{bmatrix} =$$

$$= \begin{bmatrix} -\frac{1}{h_{\mathrm{p}}} \boldsymbol{D}_V^T \boldsymbol{D}_X^{(\nu)T} \boldsymbol{L} \boldsymbol{P}_0^{(\nu)} \\ \Delta\varphi_{\mathrm{end}}^{(\nu)} \end{bmatrix}, \quad (22)$$

written with index ν for the ν-th cycle of the iteration, is obtained.

With the calculated form-parameters $\boldsymbol{c}$, the future reference velocities $\boldsymbol{V}_{\mathrm{ref}}$ are updated.

$$v, \omega_{\mathrm{ref}}^{(\nu+1)}(k + 1 + i) = \quad (23)$$
$$= v, \omega_{\mathrm{ref}}^{(\nu)}(k + 1 + i) + c_{1,3}^{(\nu)} + c_{2,4}^{(\nu)} f(i)$$

Then, the updated prediction of the position errors $\boldsymbol{P}_0^{(\nu+1)}$ and the matrix of derivatives $\boldsymbol{D}_X^{(\nu+1)}$ at the new predicted positions are calculated.
After a specified number of iterations, the algorithm terminates. Usually, a few cycles are sufficient to achieve convergence.

The weight matrices $\boldsymbol{L}$ and $\boldsymbol{R}$, the prediction horizon h_{p}, the form factor κ, the sampling time T_{s} and the number of iterations performed are the design parameters and substantially influence the performance of the system.

The optimization of a few form-parameters instead of an entire reference velocity sequence reduces the calculation time significantly, since only a 5×5-system of equations has to be solved every iteration.

Moreover, the solution is more robust and the system's bandwidth can be adapted selectively.

Prediction and optimization using the estimated side-slip angle α leads in some situations to unstable oscillatory behaviour of the side-slip angle during tracking of stationary curves, while the reference positions are matched with high accuracy.
On the other hand, when omitting the side-slip angle in the optimization, position precision is deteriorated. The controller then attempts to match the attitude angle of the robot with the reference path angle, which makes it physically impossible to keep the reference position at large side-accelerations, because the side force is generated by the side-slip.
Therefore, a compromise between stability and accuracy is sought by reducing the estimated side-slip angle α by a relaxation factor μ. A value of about 0.9 to 0.95 provides stability throughout the entire feasible 2-dimensional velocity-curvature domain for stationary curves, while keeping position precision at a reasonable level.

5.4 Application of the control law of the inner loop

The velocity error $\hat{\boldsymbol{e}}(k + 1)$ can now be calculated, as mentioned before. The velocity error integral $\boldsymbol{q}(k)$ is known from the first step of the prediction.
Therefore, the control law of the inner loop (8), written for time $k + 1$

$$\boldsymbol{u}(k + 1) = \boldsymbol{K} \begin{bmatrix} \hat{\boldsymbol{e}}(k + 1) \\ \boldsymbol{e}(k) \\ -\boldsymbol{q}(k) \end{bmatrix} \quad (24)$$

is now used to compute the PWM-input signals to the DC-motors, which are applied at the end of the current sampling interval, i.e. at time $k + 1$.
This means that the algorithm can consume the entire sampling interval to calculate the output without any negative effect on the control performance. The calculation time is thus effectively compensated for, provided it does not exceed the duration of one sampling interval.

6 RESULTS

To test the tracking algorithm, simulations are carried out using the nonlinear model.
Changing ground conditions are modeled via low frequency noise or step changes acting on the respective parameters of the nonlinear model.
The effect of possible modeling inaccuracies is simulated by simply modifying various parameters

used in controller design.

To show the ability of the system to cope with changing ground conditions, a stationary curve (i.e. a circle) with a moderate centripetal acceleration of $2.9\mathrm{ms}^{-2}$, starting and ending with a sinusoidal curvature-over-arclength profile and linear acceleration and deceleration of the track speed, Fig. 5, is used as reference trajectory, fed to the control algorithm in terms of reference x-, y- and θ-sequences.
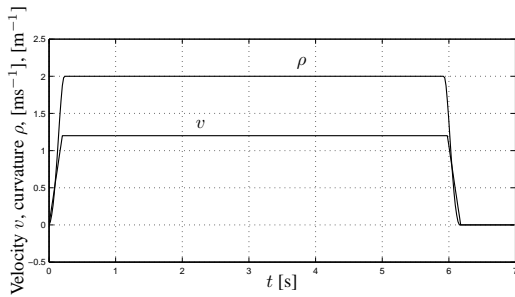

Figure 5: Velocity and curvature profile used to generate the reference trajectory

After 1.5s, the ground friction drops to 70% of the initial value and returns to 100% after 5s. The absolute of the side-slip angle increases drastically to peak values near $35°$, Fig. 6. Stability is maintained, the position error, however, increases almost proportionally with the absolute of the side-slip angle, reaching peak values of below 5cm, Fig. 7. Low frequency damped oscillations of the side-slip angle can be observed.

For comparison, the plots are underlayed with the results for the same trajectory without disturbances.
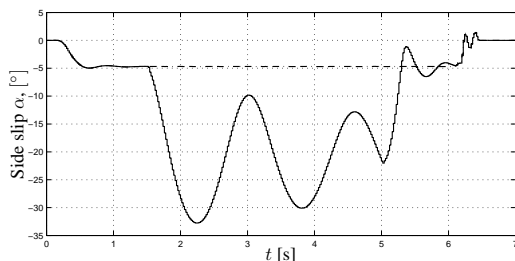

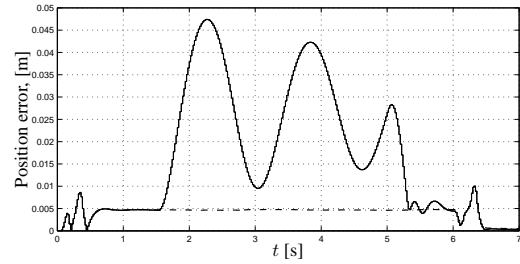Figure 6: Side-slip angle with and without disturbance (dashed)

Figure 7: Absolute position error with and without disturbance (dashed)

The corresponding input signals $r_{\mathrm{PWM,r}}$ and $r_{\mathrm{PWM,l}}$ are depicted in Fig. 8 and the calculated reference velocities v_{ref} and ω_{ref} and the true velocities v and ω are displayed in Fig. 9.
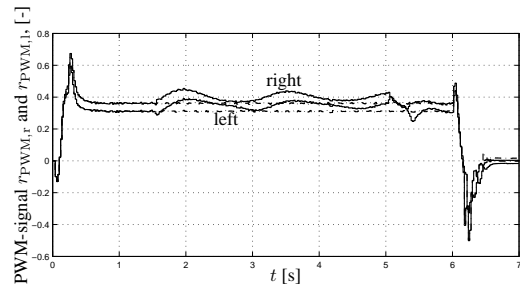

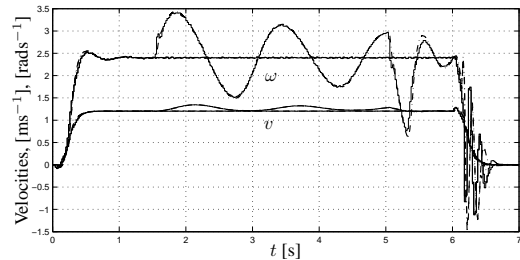Figure 8: Right and left PWM-input signal, with and without disturbance (dashed)

Figure 9: Velocities with and without disturbance, true: dashed, reference: solid

To show the performance of the algorithm at larger centripetal accelerations and larger side-slip angles, a sharp turn with a minimum radius of 0.29m at a velocity of $1.1\mathrm{ms}^{-2}$ is performed, leading to a peak side-slip angle of $45°$, Fig. 10.

A certain position error has to be accepted, but as soon as the side-acceleration (and with it the side-slip) diminishes, the error is compensated.
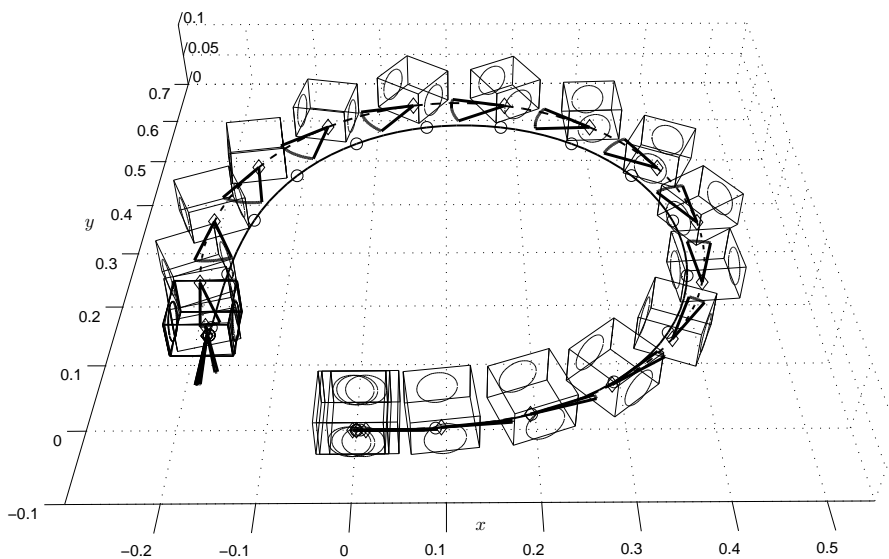
Figure 10: 3D-visualization of the robot sliding through a sharp turn

In a first practical test, the algorithm, written in C, was executed on a 600MHz-clock-rate Blackfin BF533 DSP. The calculation time consumed during each sampling interval is well below the duration of the interval.

7 CONCLUSION

A robust and universally applicable tracking control algorithm is presented, that suffers under none of the drawbacks inherent to the various approaches found in literature.

The presented tracking algorithm performs very well even under severe disturbances in a simulation.
The state feedback controller with integration of the control error of the inner loop makes the system very robust against modeling inaccuracies.

The optimization in the outer loop does not contain any model assumptions except for the kinematic equations, which are not subject to any uncertainties, and is therefore (depending on the tuning parameters) very robust against changing ground conditions.

Although the algorithm was developed for the class of unicycle-type mobile robots or 2WDD(two-wheel-differential-drive)-robots, it can easily be adapted to be used for four-wheelers with one steered axle. Only the velocity dynamics would have to be modified.

REFERENCES

Bloch, A. and Drakunov, S. (1994). Stabilization of a Non-holonomic System via Sliding Modes. *IEEE Conference on Decision and Control.*

Brockett, R. W. (1983). *Differential Geometric Control Theory*, chapter Asymptotic Stability and Feedback Stabilization. Birkhauser.

Hespanha, J. P. and Morse, A. S. (1996). Stabilization of Nonholonomic Integrators via Logic-Based Switching. *Automatica's Special Issue on Hybrid Systems, to appear.*

Isermann, R. (1987). *Digitale Regelsysteme*, volume 1. Springer.

Isidori, A. (1989). *Nonlinear Control Systems*. Springer.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M. (2000). Constrained Model Predictive Control: Stability and Optimality. *Automatica*, (36):789–814.

Norgaard, M., Ravn, O., and Poulsen, N. K. (1999). *Neural Networks for Modelling and Control of Dynamic Systems*. Springer.

Oriolo, G., De Luca, A., and Vendittelli, M. (2002). WMR Control Via Dynamic Feedback Linearization: Design, Implementation, and Experimental Validation. *IEEE Transactions on Control Systems Technology*, 10(6):835–851.

Samson, C. (1995). Control of Chained Systems. Application to Path Following and Time-Varying Point-Stabilization of Mobile Robots. *IEEE Transactions on Automatic Control*, 40(1):64–77.

Seyr, M. and Jakubek, S. (2005). Neural Network Predictive Trajectory Tracking of an Autonomous Two-Wheeled Mobile Robot. In *IFAC05 conference proceedings.*