Adaptive optimal path planning and nonlinear model predictive control for a nonholonomic ultraflat overrunable mobile robot

Georg Berndl, Klemens Springer and Hubert Gattringer  
Institute for Robotics  
Johannes Kepler University Linz  
Linz, Austria  
Email: georgberndl@gmail.com, {klemens.springer, hubert.gattringer}@jku.at

Julian Simader  
Dr. Steffan Datentechnik GmbH  
Linz, Austria  
Email: simader@dsd.at

Abstract—In this paper an adaptive optimal path planning in combination with nonlinear model predictive control is presented for trajectory tracking of an ultraflat overrunable mobile robot. As a basis for nonlinear control, the dynamical model of this nonholonomic robotic system, that is used particularly for testing purposes of advanced driver assistance systems, is derived. The aim of this work is to apply model predictive control to minimize the lateral distance between a desired path and the current robot position with respect to physical constraints of the robot. In order to smoothly approach to the desired track, in particular at high velocities, a new path is planned at each sampling instant which heads the mobile robot back on a desired path. This new reference trajectory for the model predictive controller is based on B-Splines, whereby the corresponding control vertices are calculated by means of a quadratic program with respect to minimal curvature. Implementation details as well as sufficient simulation results are shown.

I. INTRODUCTION

In the last years the application of wheeled mobile robots (WMR) in the area of automation industry and service robotics was considerably increasing. Hence trajectory tracking plays an important role in order to improve positioning accuracy and performance. As a basis for control, kinematical and dynamical modeling of nonholonomic WMRs have been exhaustively investigated, see [1], [2]. In [3] a quasi-static feedback controller, that is based on differential flatness, see [4], for nonholonomic mobile platforms is derived. Among many control concepts, there was strong research towards sliding mode control (SMC) as well, see [5], [6], for the use in the field of nonholonomic mobile platforms. In [7] the trajectory tracking problem was solved based on the exact discrete-time model. Besides the conventional PD controller, model predictive control (MPC), also known as receding horizon control, became an important control strategy in petrochemical industry in the 1980s. Therewith the trajectory tracking problem for a finite horizon can be solved in an optimal manner in consideration of state and input constraints. Due to its high computational burden it was mainly used for applications with slow sampling times. Development in processor technology and of efficient numerical algorithms, see [8] for a comprehensive overview, allows the usage of nonlinear MPC (NMPC) for plants with time constants in the lower millisecond range nowadays. In [9] the trajectory tracking problem for a WMR is compared for a linear as well as a nonlinear MPC approach using a kinematical model without slipping conditions. An NMPC approach based on a dynamical model of a unicycle-like mobile robot was proposed in [10]. The main drawback of such MPC strategies shows up at high velocities in combination with small curve radii. Therefore in this paper an adaptive optimal path planning strategy in combination with NMPC for an ultraflat overrunable (UFO) nonholonomic mobile robot is proposed. In order to approach fast but smoothly to the desired trajectory, a new curve based on B-Splines serves as a reference for the NMPC. This new reference trajectory is optimized with respect to curvature and fast approximation. Nonlinear model predictive control is solved using an efficient real-time iteration algorithm implemented in C-code in the software package ACADO, presented in [11]. The paper is organized as follows: Section II describes the robots application scenarios and the main drawback of the current control approach. Section III explains the dynamical model of the mobile robot in detail. In IV the NMPC strategy is shown. Section V introduces the implemented path planning method based on B-Splines. The proposed strategy is applied to the ultraflat mobile robot in VI and results are shown in section VII.

II. PROBLEM FORMULATION

The UFO mobile robot is a drive system used for Advanced Driver Assistance Systems (ADAS) testing. It is a platform able to move different objects on the road like car or pedestrian dummies without the need of having the same kinematic constraints, as shown in Fig. 1. The robot is driven by two electric drives on the rear axle and one for the steering and has four wheels. Furthermore, the robot has a settle-down function which makes it possible to unload the wheels in case of a run-over. These functions allow testing of pre-crash, crash and post-crash real world scenarios. In case of an impact, the UFO can be overrun and the target is pushed away. In Fig. 2, a typical junction test scenario is shown, whereby the UFO moves a dummy car. Further scenarios are lane changing or distance keeping. Before such a maneuver starts, the operator specifies a desired path, the dummy car...
III. Modeling & Setup

The overall system of the mobile robot is divided into two parts. The control unit responsible for path planning, position control and data processing on the one hand and the mobile robot (Fig. 3) on the other hand. In order to ensure real-time capability of the implemented algorithms, a powerful microprocessing system is used for the control unit. The robot is divided into five bodies. Four out of five consist of a wheel and an axle mounting, whereby the steering of the front wheels is mechanically coupled. The fifth body describes the chassis of the robot. The two drives for the rear wheels that are responsible for the required torque $M_f$ to move the WMR are mounted on the rear axle. The steering angle $\gamma$ is actuated by an additional drive generating the torque $M_s$. For controlling the wheel and steering positions, encoders and a potentiometer are used for measurement.

The mathematical model of the considered mobile robot in Fig. 3 with $N = 5$ bodies is obtained with the Projection Equation

$$\sum_{i=1}^{N} \begin{cases} \left( \frac{\partial \mathbf{v}_c}{\partial \mathbf{s}} \right)^T & \mathbf{r} \mathbf{p} + \mathbf{R} \mathbf{\omega}_IR \mathbf{r} \mathbf{p} - \mathbf{r} \mathbf{F}^i \mathbf{L} + \mathbf{R} \mathbf{\omega}_IR \mathbf{L} - \mathbf{r} \mathbf{M}^e_i \end{cases} = 0,$$

(1)

see [12]. The $\{\}$ operator denotes the skew-symmetric matrix representing the cross product ($\hat{a}b = a \times b$). This method projects the linear $\mathbf{p}$ and angular momenta $\mathbf{L}$, formulated in an arbitrary reference frame $\mathbf{R}$, with the Jacobian matrices $(\partial \mathbf{R} \mathbf{v}_c / \partial \mathbf{s})^T$ and $(\partial \mathbf{R} \mathbf{\omega}_s / \partial \mathbf{s})^T$ into the direction of unconstrained motion. In this formulation $\mathbf{R} \mathbf{v}_c$ and $\mathbf{R} \mathbf{\omega}_s$ denote the linear and angular velocities of the center of gravity of each body $i$. The variables $\mathbf{r} \mathbf{F}^i$ and $\mathbf{r} \mathbf{M}^e_i$ denote impressed forces and moments. Based on that, the equations of motion in minimal description are obtained

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{s}} + \mathbf{g}(\mathbf{q}, \dot{\mathbf{s}}) = \mathbf{B} \mathbf{u}$$

$$\mathbf{u} = \mathbf{M}_{\text{mot}}$$

(2)

with minimal coordinates $\mathbf{q} = [x_{ch} \ y_{ch} \ \gamma_{ch} \ \psi]_T$ and minimal velocities $\dot{\mathbf{s}} = [v_{ch} \ \dot{\gamma}_{ch}]_T$. The variables $x_{ch}, \gamma_{ch}$ and $\psi_{ch}$ describe the position and orientation of the center $C$ on the rear axle. The speed of the mobile robot is given by $v_{ch}$. Terms due to gravity and all other nonlinear effects like Coriolis and centrifugal forces can be found in $\mathbf{g}(\mathbf{q}, \dot{\mathbf{s}})$. Due to the fact that dissipative terms like friction were disregarded, the mobile robot moves with the assumption of pure rolling (without slipping) between wheels and ground. The constraints for the rear wheels are

$$\omega_{\text{wheel}}R_{\text{wheel}} = v_{ch} \pm \frac{\gamma_{ch}}{2},$$

(3)

whereby $R_{\text{wheel}}$ denotes the radius and $\omega_{\text{wheel}}$ the angular velocity of the wheel. $\mathbf{M}$ represents the positive definite mass matrix and $\mathbf{B}$ contains the gear ratios and maps the actuator torques $\mathbf{M}_{\text{mot}} = [M_d \ M_r]_T$ into minimal space.
Using the kinematic model

$$\mathbf{q} = \mathbf{H}^+(\mathbf{q}) \mathbf{s} = \begin{bmatrix} \cos(\gamma_h) & 0 \\ \sin(\gamma_h) & 0 \\ \\frac{1}{2} \tan(\gamma_r) & 0 \\ 0 & 1 \end{bmatrix} \left( v_{ch} \right)$$

(4)

with the Moore-Penrose pseudoinverse \( \mathbf{H}^+ \), the equations of motion from (2) can be formulated in state space representation to

$$\dot{\mathbf{x}} = \mathbf{H}^+(\mathbf{q}) \dot{\mathbf{s}} = \left( \mathbf{M}(\mathbf{q})^{-1}(\mathbf{Bu} - \mathbf{g}(\mathbf{q}, \dot{\mathbf{s}})) \right)$$

(5)

$$\mathbf{y} = \mathbf{x}$$

with \( \mathbf{x} = [\mathbf{q}^T \dot{\mathbf{s}}^T]^T \). The parameter \( a \) denotes the axial distance.

IV. MODEL PREDICTIVE CONTROL

The main idea of MPC is to minimize the control error over a finite horizon using an optimization algorithm subject to state and input constraints. Written in a discretized form, in each sampling step \( k \) an MPC computes the optimal control solution as a result of the minimization of a quadratic cost function

$$\min_{\mathbf{u}} \mathbf{J} = \min_{\mathbf{u}} \left[ \sum_{j=1}^{n_{ph}} \left( \mathbf{r}_{k+j} - \mathbf{\hat{y}}_{k+j|k} \right)^T \mathbf{Q} \left( \mathbf{r}_{k+j} - \mathbf{\hat{y}}_{k+j|k} \right) + \sum_{j=0}^{n_{ch}-1} \left( \mathbf{\hat{u}}_{k+j|k} \right)^T \mathbf{R} \left( \mathbf{\hat{u}}_{k+j|k} \right) \right]$$

(6)

dependent on the optimization variable \( \mathbf{u} = [\mathbf{\hat{u}}_{k|k}^T, \ldots, \mathbf{\hat{u}}_{k+n_{ch}-1|k}^T]^T \) with \( \mathbf{\hat{u}}_{k+j|k} = [\mathbf{M}_{d,k+j|k}, \mathbf{M}_{s,k+j|k}]^T \). The notation \( \mathbf{M}_{d,k+j} \) describes the predicted value \( \mathbf{M}_d \) at time \( t_{k+j} \) in sampling instant \( k \). For the evaluation of the objective function in (6), the future system outputs \( \mathbf{\hat{y}}_{k+j|k} = [\mathbf{\hat{y}}_{x, k+j|k}, \mathbf{\hat{y}}_{x, k+j|k}]^T \) are predicted based on the current measured state \( \mathbf{x}_k \), the future system inputs \( \mathbf{\hat{u}}_{k+j|k} \), the finite prediction horizon \( n_{ph} \), the finite control horizon \( n_{ch} \) and the dynamical model of the system from section III. Then the error between the predicted values and the reference trajectory \( \mathbf{r}_{k+j} \) as well as system inputs are penalized. The variables \( \mathbf{Q} \) and \( \mathbf{R} \) denote weighting matrices, whereby high values in \( \mathbf{Q} \) entail lower tracking errors as well as high values in \( \mathbf{R} \) result in lower input values. The first of the \( n_{ch} \) optimal inputs is applied to the system. Then the same procedure starts again at the next sampling instant \( t_{k+1} \) with a new measured state \( \mathbf{x}_{k+1} \) and the prediction horizon is shifted forward. Hence MPC is also known as receding horizon control. In contrast to PID control, MPC is able to anticipate future changes and to react accordingly. In the application of trajectory tracking with a WMR, this allows the mobile robot at higher velocities to better approach to curve segments with big curvatures. A further advantage of MPC is to consider restrictions on inputs and states

$$\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max}$$

(7)

$$\mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}$$

A main drawback of MPC strategies is the computational burden of the optimization process. The higher the number of optimization variables, the longer it takes to find a solution of (6).

In order to set up this optimization problem, a reference trajectory \( \mathbf{r}_{k+j} \) is required. Therefore, in section V an adaptive path planning which provides the necessary data is explained in detail.

V. PATH PLANNING

Due to unsatisfactory results by using the desired path as reference path, an adaptive path is implemented which heads the mobile robot back on the desired path. This shortcoming is due to the fact, that the minimization of the orientation error and the lateral distance of the nonholonomic WMR are inherently contradicting.

Hence, at each sampling time \( t_k \) a new path is planned which has to satisfy the following constraints:

- Boundary condition of start \( \mathbf{p}_{\text{start}} \) and end point \( \mathbf{p}_{\text{end}} \) of the path.
- Boundary conditions of first derivative and curvature \( \mathbf{p}_{\text{start}}, \kappa_{\text{start}} \) and \( \mathbf{p}_{\text{end}}, \kappa_{\text{end}} \).
- Continuity of first derivative and curvature over the whole path.
- Path should be composed of a sequence of segments.

This leads to better path control (more inputs to steer the path).

- Ability to describe line and curve segments.

Cubic basis splines (B-Splines) are able to consider these definitions and are chosen as suitable functions.

A. Derivation of cubic B-Spline basis functions

A cubic B-Spline curve is composed of local shape functions (basis functions) of degree \( m = 3 \) each. Hence, one single point on the two-dimensional curve

$$\mathbf{p}_i(u) = \begin{bmatrix} p_{x,i}(u) \\ p_{y,i}(u) \end{bmatrix} = \sum_{j=0}^{3} b_j(u) \begin{bmatrix} k_{x,j+i-1} \\ k_{y,j+i-1} \end{bmatrix} \quad 0 \leq u \leq 1$$

(8)

is completely controlled by just \( m + 1 = 4 \) control points for every direction \((x \text{ and } y)\), called vertices. Each vertex \( k_{x,y,j+i-1} \) is associated with a scalar-valued parametric function

$$b_3(u) = (-u^3 + 3u^2 - 3u + 1)/6$$

(9)

$$b_2(u) = (3u^3 - 6u^2 + 4)/6$$

(10)

$$b_1(u) = (-3u^3 + 3u^2 + 3u + 1)/6$$

(11)

$$b_0(u) = u^3/6$$

(12)

that define a basis of cubic curves. A more detailed description to determine these coefficients is shown in [13] and leads to basis functions of degree \( m = 3 \). As shown in Fig. 4, each curve segment is controlled by 4 vertices and the associated B-Spline. In Fig. 5 the main idea of the path planning scheme is sketched. The B-Spline based path (black, dashed) guides the mobile robot back on the desired path (black, solid). This dashed path is composed of several curve segments.
In order to determine the path, it requires the calculation of \(v_c\), as shown later. The number of path segments \(n_{\text{path}}\) of the WMR, see Fig. 5, depends on the desired velocity \(v_c\) and the prediction horizon \(n_{\text{ph}}\). In order to determine the path, it requires the calculation of \(n_v = 2n_{\text{path}} + 6\) vertices. This is solved by means of an optimization problem which can be rewritten into a standard QP problem

\[
\min_{k \in \mathbb{R}^n} \frac{1}{2} k^T H k + k^T g
\]

s.t. \(A k \leq b\)

with the bound \(b\) as well as inequality and equality constraints determined in \(A\). The matrix \(H\) is defined as the Hessian matrix and \(g\) represents the gradient vector. The reasons for determining the control vertices by means of an optimization problem are motivated as follows:

- More equations \(n_q = 2(n_{\text{path}} - 1) + 10\) (5 at start and end, respectively) than unknown control vertices \(n_v = 2n_{\text{path}} + 6\) (over-determined system of equations) are given.
- It allows to influence the curvature \(\kappa\) on specified trajectory points.

The linear constraints in (14) contain equations to fulfill the continuity constraints \(P_{\text{start}}, \dot{P}_{\text{start}}, \kappa_{\text{start}}, P_{\text{end}}, \dot{P}_{\text{end}}\) and \(\kappa_{\text{end}}\). The Hessian matrix for the optimization problem in (14) to minimize (13) is given exemplarily for the corresponding path segment \(p_i(u)\) with

\[
H_d, i = \begin{pmatrix}
1/36 & 0 & 1/9 & 0 & 1/36 & 0 \\
0 & 1/36 & 0 & 1/9 & 0 & 1/36 \\
1/9 & 0 & 4/9 & 0 & 1/9 & 0 \\
0 & 1/9 & 0 & 4/9 & 0 & 1/9 \\
1/36 & 0 & 1/9 & 0 & 1/36 & 0 \\
0 & 1/36 & 0 & 1/9 & 0 & 1/36
\end{pmatrix}
\]

The linear part is defined as the gradient

\[
g_{d, i} = \begin{pmatrix}
-x_{fi}/3 \\
y_{fi}/3 \\
-x_{fi}/3 \\
-4x_{fi}/3 \\
-4y_{fi}/3 \\
-x_{fi}/3 \\
-y_{fi}/3
\end{pmatrix}
\]

In order to avoid oscillations in the resulting paths, that may lead to bad tracking behavior, curvature is minimized additionally. Curvature \(\kappa\) of a two-dimensional parametric curve is described by the nonlinear equation

\[
\kappa_i = \frac{p_{x,i}'(u)p_{y,i}''(u) - p_{x,i}''(u)p_{y,i}'(u)}{(p_{x,i}'(u)^2 + p_{y,i}'(u)^2)^{3/2}}.
\]

Due to the fact that this can not be reformulated in quadratic form, a direct implementation in (14) is not possible. The choice of \(d_f = 1\) (distance between future points) and \(u = 1\) (end of a path segment) limits the denominator of (15) approximately to one and the slope terms between \(-1 \leq p_{x,i}'(u), p_{y,i}'(u) \leq 1\). From the nominator, only the terms of the second derivatives \(p_{x,i}''(u), p_{y,i}''(u)\) are used for minimization because all other terms are already known in fixed areas. This is equivalent to a decoupled bending minimization of the \(x\)- and \(y\)-curve. Then the additional requirement to minimize curvature is formulated

\[
\min_{k \in \mathbb{R}^n} \sum_{i=1}^{n_{\text{path}}-1} p_{x,i}'(u)^2 + p_{y,i}'(u)^2.
\]
Again, the Hessian matrix is given exemplarily for the corresponding path segment $p_i(u)$ with

$$H_{k,i} = \begin{pmatrix} 1 & 0 & -2 & 0 & 1 & 0 \\ 0 & 1 & 0 & -2 & 0 & 1 \\ -2 & 0 & 4 & 0 & -2 & 0 \\ 0 & -2 & 0 & 4 & 0 & -2 \\ 1 & 0 & -2 & 0 & 1 & 0 \\ 0 & 1 & 0 & -2 & 0 & 1 \end{pmatrix}.$$  

The overall Hessian matrix $H$ and gradient vector $g$ are represented by

$$H = f(H_{d,i}, H_{x,i})$$

$$g = f(g_{d,i}).$$

The resulting optimization problem can be solved with an appropriate QP solver as mentioned in section VI. The obtained geometric path $p_i(u)$ is represented in dependence of the parameter $u$. The only time dependence which influences the path is given through the length of the prediction horizon $T_p = n_{ph} T_s$ with sample time $T_s$.

### C. Reference trajectory

After computation of the geometric path $p_i(u)$, the necessary time dependence to obtain the reference trajectory $r_{k+j}$ can be deduced. Based on the desired velocity and acceleration of the original trajectory (remember the assumption of constant desired speed), the distances

$$L(u_j) = v_{ch} T_i j \quad j = 1, 2, ..., n_{ph}$$

(18)

to the $n_{ph}$ reference trajectory points are found. With the desired robot speed $v_{ch,d}$, the equation

$$v_{ch,d} T_i j = \sum_{i=1}^{n_{ph}} \int_0^{u_j} \| p_i''(u) \| du_j$$

(19)

must hold and defines a relationship between the path parameter $u$ and time. Unfortunately, this expression can not be solved analytically with respect to $u$. Therefore, an advanced numerical integration method like e.g. the Simpson’s rule is required. Determining $n_{ph}$ curve parameters $u_j$ leads to the necessary reference trajectory $x_j = p_s(u_j)$, $y_j = p_y(u_j)$, $\gamma_ch = \arctan(p_y''(u_j)/p_x''(u_j))$ and $\gamma_d = \arctan(a_k(u_j))$.

Due to the minimization of $\gamma$ within the optimization problem, the change of the curvature is slower. Therewith the performance of trajectory tracking is increased.

### VI. APPLICATION

#### A. Used software packages

The software package ACADO Toolkit [15] was used for the realization of the NMPC. ACADO is an open source software environment and algorithm collection. In order to solve NMPC problems, it contains all necessary software like

- Hessian approximation methods
- Discretization methods (single- or multiple shooting)
- Different integrator types (Runge Kutta, Euler method, Dormand-Prince)

Furthermore, it exports optimized, highly efficient C-code. This allows simple implementation on embedded boards. The optimization is based on a sequential quadratic programming (SQP) method. This iterative method reduces the nonlinear optimization problem to a quadratic problem (QP) which can be solved with the software package qpOASES, see [16]. Furthermore, qpOASES is utilized to solve the quadratic program defined in (13) and (16).

#### B. Simulation flow

Figure 6 shows the control scheme. In order to check the behavior of the closed loop, disturbed parameter values are used for the NMPC model. At each sampling interval, the simulation process can be divided into the following steps.

1) Project the mobile robot onto the desired path
2) Compute future points $p_{f,i}$
3) Set up the optimization problem
4) Compute the geometric path $p$
5) Compute the reference trajectory $r_{k+j}$
6) Set up the NMPC
7) Solve the nonlinear optimization problem
8) Start again from 1)

Steps 1 to 5 are computed within the optimal path planning block in Fig. 6 while steps 6 to 7 describe the procedures done by the NMPC block.

### VII. RESULTS

The reference path starts at the origin $x_r = 0$ m and $y_r = 0$ m. The initial condition of the mobile robot is $x_0 = [0 \ 0.2 \ 0 \ 0 \ 10 \ 0]^T$, whereby the lateral distance at $t = 0$ s is 0.2 m and the start speed 10 m/s. The weighting matrices are defined as $Q = \text{diag}[10^2, 10^2, 10^3, 10^2, 10^2, 1]$ and $R = \text{diag}[1, 1]$. The prediction horizon is set to $n_{ph} = 10$ and the sampling time to $T_s = 0.05$ s. The steering angle is mechanically limited to

$$-\frac{14}{180} \pi \leq \gamma \leq \frac{14}{180} \pi.$$

Figure 7 shows tracking results of the WMR when following a desired path. For the first circle segment a higher steering angle is necessary than limited in (20). Hence it is not possible to follow the path exactly, but the NMPC chooses the steering torque $M_t$ appropriately to minimize the tracking error over the whole horizon with respect to the limits of $\gamma_r$, see Fig. 8. As shown in Fig. 9, the initial lateral deviation to the desired path is minimized immediately. During the first circle segment a higher error occurs due to active constraints. For the following trackable segments the error vanishes. Due to numerical integration and the fact that cubic B-Splines are not able to represent curve segments exactly, a neglectable
error always occurs on curve segments. On line segments at the end of the path in Fig. 7 the error convergences to zero, see Fig. 9.

VIII. CONCLUSIONS

In this paper an adaptive optimal path planning approach to lead a mobile robot on a defined reference path using NMPC is introduced. Therefore a geometric path is planned based on B-Splines and approached optimally to the desired trajectory with respect to minimal curvature. This path is transformed to a time dependent trajectory solving an integral condition using the Simpson’s rule. The resulting trajectory serves as reference for the cascaded nonlinear model predictive control, that steers the mobile robot back to the desired path in an optimal manner considering state constraints. In a last step the concept was applied to a nonholonomic mobile platform, that demonstrates the good performance of the proposed path planning concept as well as the benefit from MPC with respect to prediction and constraint consideration. Future work will focus on implementing this strategy to the real robot.

ACKNOWLEDGMENT

This work was thankfully supported by Dr. Steffan Daten-technik Ges.m.b.H, Salzburgerstr. 34, 4020 Linz, Austria - see http://www.dsd.at.

REFERENCES