Path Following Mobile Robot in the Presence of Velocity Constraints

Article

3 authors, including:

Niels Poulsen
Technical University of Denmark
236 PUBLICATIONS **4,629** CITATIONS

SEE PROFILE

Ole Ravn
Technical University of Denmark
121 PUBLICATIONS **2,582** CITATIONS

SEE PROFILE

Path Following Mobile Robot in the Presence of Velocity Constraints

Martin Bak*, Niels Kjølstad Poulsen** and Ole Ravn*

*Ørsted•DTU, Automation,
Building 326, Elektrovej
Technical University of Denmark
DK-2800 Kongens Lyngby, Denmark
{mba,or}@iau.dtu.dk

**Informatics and Mathematical Modelling,
Building 321, Richard Petersens Plads
Technical University of Denmark
DK-2800 Kongens Lyngby, Denmark
nkp@imm.dtu.dk

Abstract

This paper focuses on path following algorithms for mobile robots with velocity constraints on the wheels. The path considered consists of straight lines intersected with given angles. We present a fast real-time receding horizon controller which anticipates the intersections and smoothly controls the robot through the turnings while fulfilling the velocity constraints.

Keywords: Mobile Robots, predictive control, robot motion control, robot applications, autonomous vehicles

1 Introduction

Path following is a useful motion control approach when maneuvering mobile robots from one area to another (Samson 1995). In this paper, we consider a path consisting of straight-line sections intersected with given angles. Particularly in an indoor environment, where rooms often are rectangular, this kind of paths are often encountered, possibly with perpendicular intersections. Furthermore, the specification and actual implementation of such a path is easy. An appropriate path following algorithm should therefore be capable of controlling the robot through such intersections.

Sharp turnings raise a problem. The vehicle velocities, heading and angular, must be constrained such that the turning is appropriately restrained and smooth. A large heading velocity together with a large angular velocity will jeopardize the stability and safety of the robot or cause saturation in the motors which again will cause overshooting and long settling time. The velocity constraints can either be *self-imposed* due to desired behavior and safety concerns or *physical* due to actual limitations such as currents and voltages in the motors.

To avoid excessive overshooting and to have time to decelerate when turning, the presented controller is based on a strategy that forecasts the turning using a receding horizon approach where the controller predicts the posture of the robot and together with knowledge of an upcoming intersection compensates the control signals, appropriately. Predictive path planning was discussed in (Normey-Rico, Gómez-Ortega & Camacho 1999, Ferruz & Ollero 1998).

The general path following problem is characterized by the forward velocity not being part of the control problem opposed to the *path tracking problem* where typically a virtual reference cart is tracked (de Wit, Siciliano & Bastin 1996, Koh & Cho 1999, Samson & Ait-Abderrahim 1991) and both the forward and angular velocity are controlled. Hence, path following has an extra degree of freedom (but only controls two degrees) which allows handling of constraints by scaling the forward velocity. This has been exploited in (Bemporad, Marco & Tesi 1997) for a wall-following mobile robot. In (Koh & Cho 1999) the constrained path tracking problem is discussed.

The paper is organized as follows. Section 2 states the problem of path following and models the constraints acting on the system. In Section 3 a velocity scaling algorithm for a straight line path is presented that respects the constraints. Section 4 introduces the receding horizon approach capable of smoothly following sharp turnings. Some experimental results are reported in Section 5 and finally are conclusions given in Section 6.

2 Problem Formulation

The mobile robot under consideration is a so-called *uni-cycle robot* with two differential-drive wheels on the same axle and one castor wheel, see Figure 1. A robot with this kind of wheel configuration has an underlying *nonholonomic* property that restricts the mobility of the robot in the sideways direction. This adds to the complexity of the motion control problem. The posture

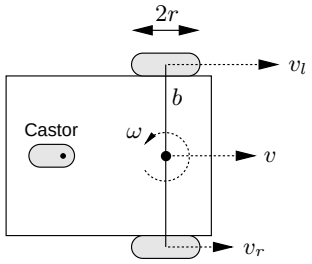


Figure 1. The mobile robot.

(position and orientation) of the mobile robot is given by the kinematic equations

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega, \end{aligned} \tag{1}$$

where (x, y) indicate the position of the robot center in the Cartesian space and θ is the heading angle of the robot (angle between the x-axis and the axis of the robot). The inputs are the heading or forward velocity v (defined as $v = \dot{x}\cos\theta + \dot{y}\sin\theta$) and the angular velocity ω. Combined, the triplet $(x, y, \theta)^T$ defines the posture of the robot.

2.1 Path Following

The path following problem is illustrated in Figure 2 where P is the orthogonal projection of the robot point R onto the path. The signed distance between P and R is denoted d. A turning is placed at C and the signed distance from C to P along the path is denoted s. The orientation error is defined as $\tilde{\theta} = \theta - \psi$, where ψ is the orientation reference. The two path sections have orientation ψ_1 and ψ_2, respectively, with $|\psi_2 - \psi_1| < \pi$ assumed. Shown are also two bisection lines defined by the angles $\alpha = \frac{\psi_2 - \psi_1}{2}$ and $\alpha + \frac{\pi}{2}$. These lines will later be used to determine at what point the reference should change.

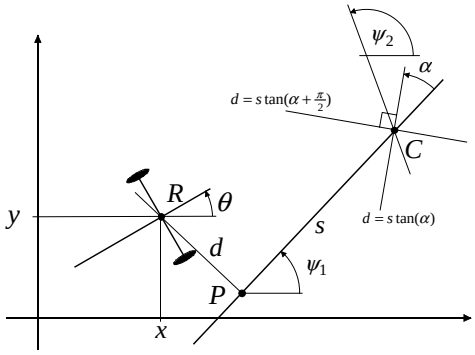


Figure 2. The path following problem. The path consists of two straight-line sections.

Given such a path and the mobile robot's forward velocity $v(t)$ the *path following problem* consists in finding a feedback control law $\omega(t)$ such that the distance to the path and the orientation error tend to zero.

For a piece-wise straight path, the path following problem is parameterized as

$$\begin{aligned} \dot{s} &= v\cos(\theta - \psi) \\ \dot{d} &= v\sin(\theta - \psi) \\ \dot{\theta} &= \omega, \end{aligned} \tag{2}$$

where s is measured with respect to the upcoming intersection.

2.2 Constraints

Constraints exist at different levels of a control system for a mobile robot. At the motor level, voltages and currents are magnitude limited, and at trajectory level, the same goes for velocities and accelerations. Since the path following algorithm is a velocity trajectory generator that generates references to the underlying motor controllers, only constraints on the velocities of the robot are considered. This is partly justified by the fact that, typically, the motors of a mobile robot are capable of delivering larger velocities than desirable during normal operation. Hence, velocity constraints are often imposed and magnitude saturation in the actual actuators (the motors) are generally not of concern, except when fast high performance trajectory generators are designed. Furthermore, only magnitude saturations are considered.

Let $u = (v, \omega)^T$ and $u_w = (v_r, v_l)^T$, where v_r and v_l denote the right and left wheel velocities, respectively. Discarding wheel slippage and uneven floors, the heading and angular velocities relate to the left and right wheel velocities in the following way

$$u = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix} = F_w u_w, \tag{3}$$

where b is the length of the wheel base of the robot. Let $v^w_{\max}$ and $v^w_{\min}$ be the maximum and minimum allowable velocities for the left and right wheels (we assume equal constraints on the two wheels), i.e.

$$v^w_{\min} \leq v_r \leq v^w_{\max}, \quad v^w_{\min} \leq v_l \leq v^w_{\max}. \tag{4}$$

Besides the wheel constraints it is convenient to be able to put a limit directly on the forward and angular velocities. For example, for $v = 0$, the constraints on the wheels alone may allow an undesirable maximum angular velocity. The constraints are

$$v_{\min} \leq v \leq v_{\max}, \quad \omega_{\min} \leq \omega \leq \omega_{\max}. \tag{5}$$

Figure 3 illustrates how the constraints relate. It is assumed that zero belongs to the set of valid velocities. In combined compact form the velocity constraints are

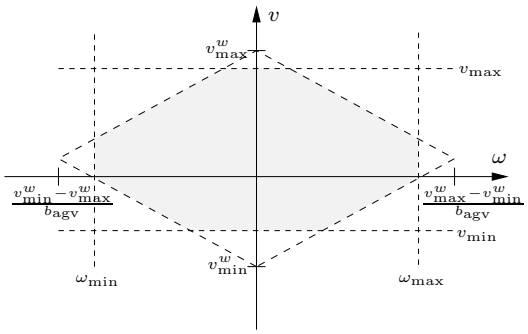


Figure 3. The velocity constraints.

$$\begin{bmatrix} F_w^{-1} \\ -F_w^{-1} \\ \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix} \end{bmatrix} u \leq \begin{bmatrix} \mathbf{1}_2(v^w_{\max}) \\ -\mathbf{1}_2(v^w_{\min}) \\ \omega_{\max} \\ -\omega_{\min} \\ v_{\max} \\ -v_{\min} \end{bmatrix}, \tag{6}$$

or

$$Pu \leq q, \tag{7}$$

where $\mathbf{1}_m(x) = (x, x, \ldots, x)^T$ is a vector of length m.

3 Velocity Scaling

In this section the path is assumed perfectly straight and infinitely long, that is, the corner is not under consideration. We consider a velocity scaling approach to a standard linear controller such that the constraints are satisfied. In Section 4 the method is applied to the receding horizon controller. The velocity scaling was implicitly introduced in (Dickmanns & Zapp 1987) and further explored in the control context in (Sampei, , Tamura, Itoh & Nakamichi 1991, Samson 1992)

3.1 Linear Controller

In the neighborhood of the origin ($d = 0, \tilde{\theta} = 0$), the linearization of (2) gives

$$\begin{aligned} \dot{d} &= v\tilde{\theta} \\ \dot{\tilde{\theta}} &= \omega. \end{aligned} \tag{8}$$

Assuming that v is different from zero (but not necessarily constant), this system (8) is controllable and stabilizable when using a linear state feedback controller of the form

$$\omega = -l_1 v d - l_2 |v| \tilde{\theta}, \tag{9}$$

with $l_1 > 0$ and $l_2 > 0$, as discussed in (de Wit et al. 1996). For a constant v, this controller reverts to a classical linear time-invariant state feedback. The velocity v is included in the controller gains

such that the closed-loop xy-trajectory response is independent of the velocity of the vehicle. As will be demonstrated, the gains l_1 and l_2 are chosen with respect to the *distance* response instead of the corresponding time response in the case of time equations. For a given v, consider the closed-loop equation for output d:

$$\ddot{d} + l_2|v|\dot{d} + l_1 v^2 d = 0, \tag{10}$$

where we identify the undamped natural frequency ω_n and the damping ratio ζ as

$$\omega_n = |v|\sqrt{l_1}, \quad \zeta = \frac{l_2}{2\sqrt{l_1}}. \tag{11}$$

For a second-order linear system, the transient peak time (time from reference change to the maximum value) t_{peak} is a function of the natural frequency ω_n and the damping ratio ζ:

$$t_{\text{peak}} = \frac{1}{\omega_n} \exp\left(\frac{\zeta \cos^{-1}(\zeta)}{\sqrt{1-\zeta^2}}\right), \quad 0 \leq \zeta < 1. \tag{12}$$

Define the peak distance as $d_{\text{peak}} = |v|t_{\text{peak}}$. Thus, (11) and (12) suggest selecting the gains l_1 and l_2 as

$$l_1 = \left(\frac{\exp\left(\frac{\zeta \cos^{-1}(\zeta)}{\sqrt{1-\zeta^2}}\right)}{d_{\text{peak}}}\right)^2, \quad l_2 = 2\zeta\sqrt{l_1}. \tag{13}$$

3.2 Scaling

The controller (9) only determines the angular velocity ω while the forward velocity v is left to the operator to specify. This extra degree of freedom and the fact that for the controller (9) $\omega \to 0$ for $v \to 0$, allow us to handle the velocity constraints by scaling the forward velocity such that $v = \gamma v_{\text{des}}$, where v_{des} is the desired velocity of the vehicle and $\gamma \in [0; 1]$ is a scaling factor. This way the constrained xy-trajectory will remain the same as the unconstrained, only the traverse will be slower.

For a given distance error d and orientation error $\tilde{\theta}$, the scaled control law (9) have the form

$$\omega = -k(d, \tilde{\theta})\gamma v_{\text{des}}, \tag{14}$$

with

$$k(d, \tilde{\theta}) = l_1 d + l_2 \text{sign}(v_{\text{des}})\tilde{\theta}. \tag{15}$$

We need to determine the scaling factor γ such that the following inequality is satisfied,

$$P \begin{bmatrix} 1 \\ -k(d, \tilde{\theta}) \end{bmatrix} \gamma v_{\text{des}} = P'\gamma \leq q. \tag{16}$$

where P and q are defined in (7). Since P' is a vector and $\gamma \in [0; 1]$, the inequality is satisfied by setting

$$\gamma = \min\left\{1, \frac{(q)_i}{(P')_i}, i = \{j \,|(P')_j > 0, j = 1, \ldots, 8\}\right\}.$$

where $(x)_i$ denotes the i'th element of the vector x. The selection of γ can be interpreted as a time-varying velocity scaling and by online determination it is guaranteed that the constraints on the velocities are not violated.

4 Receding Horizon

This section considers the problem of maneuvering a nonholonomic mobile robot through a turning consisting of two straight-line sections by means of a sensor-based closed-loop receding horizon control strategy. A good solution to the path following problem should have the succeeding properties:

- it seems naturally to anticipate the corner and to embark on the turning before reaching the actual turning point. This will smooth the turning
- the forward velocity should be decreased (possibly to zero) as the vehicle rotates around the corner such that the vehicle may follow the path with an arbitrarily accuracy.

The anticipation of the corner suggests a receding horizon approach where the control signals are based on predictions of the robot's posture while the decrease in velocity suggests the use of scaling.

4.1 Model

A discretized version of (8) with sampling period T is found by integration:

$$\begin{aligned} d_{k+1} &= d_k + Tv\left[\theta_k - \psi_k + \frac{T}{2}\omega_k\right] \\ \theta_{k+1} &= \theta_k + T\omega_k, \end{aligned} \tag{17}$$

where k denote the sampling index, i.e. $t = kT$. Since we eventually want to apply the velocity scaling to the receding horizon approach, we introduce a new control signal φ defined as $v\varphi = \omega$ where we for a constant φ have $\omega \to 0$ for $v \to 0$.

Define the state vector $z_k = (d_k, \theta_k)^T$ and the reference vector $r_k = (0, \psi_k)^T$, and rewrite (17) to

$$z_{k+1} = \begin{bmatrix} 1 & Tv \\ 0 & 1 \end{bmatrix} z_k + \begin{bmatrix} \frac{T^2}{2}v^2 \\ Tv \end{bmatrix} \varphi_k + \begin{bmatrix} 0 & -Tv \\ 0 & 0 \end{bmatrix} r_k, \tag{18}$$

or

$$z_{k+1} = Az_k + B_\varphi \varphi_k + B_r r_k. \tag{19}$$

4.2 Criterion

The predictive receding horizon controller is based on a minimization of the criterion

$$J = \sum_{n=0}^{N} (\hat{z}_{k+n} - r_{k+n})^T Q (\hat{z}_{k+n} - r_{k+n}) + \lambda \varphi_{k+n}^2, \tag{20}$$

subject to the inequality constraint

$$P \begin{bmatrix} v_n \\ v_n \varphi_n \end{bmatrix} \leq q, \qquad n = 0, \ldots, N, \tag{21}$$

where $\hat{z}$ is the predicted output, Q is a weight matrix, λ is a scalar weight, and N is the prediction horizon.

4.3 Predictor

An n-step predictor $\hat{z}_{k+n|k}$ is easily found from iterating (18). Stacking the predictions $\hat{z}_{k+n|k}, n = 0, \ldots, N$ in the vector $\hat{Z}$ yields

$$\hat{Z}_k = \begin{bmatrix} \hat{z}_{k|k} \\ \vdots \\ \hat{z}_{k+N|k} \end{bmatrix} = Fz_k + G_\varphi \Phi_k + G_r R_k, \tag{22}$$

with

$$\Phi_k = \begin{bmatrix} \varphi_k, \ldots, \varphi_{k+N} \end{bmatrix}^T, \quad R_k = \begin{bmatrix} r_k, \ldots, r_{k+N} \end{bmatrix}^T,$$

and

$$F = \begin{bmatrix} I & A & \cdots & A^N \end{bmatrix}^T$$

$$G_i = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ B_i & 0 & \cdots & 0 & 0 \\ AB_i & B_i & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ A^{N-1}B_i & \cdots & AB_i & B_i & 0 \end{bmatrix},$$

where index i should be substituted with either φ or r.

4.4 Controller

Using the N-step predictor (22) simplifies the criterion (20) to

$$J_k = \left(\hat{Z}_k - R_k\right)^T I_Q \left(\hat{Z}_k - R_k\right) + \lambda \Phi_k^T \Phi_k, \tag{23}$$

where I_Q is a diagonal matrix of appropriate dimension with instances of Q in the diagonal. The unconstrained controller is found by minimizing (23) with respect to Φ:

$$\Phi_k = -L_z z_k - L_r R_k, \tag{24}$$

with

$$\begin{aligned} L_z &= (\lambda + G_\omega^T I_Q G_\omega)^{-1} G_\omega^T I_Q F \\ L_r &= (\lambda + G_\omega^T I_Q G_\omega)^{-1} G_\omega^T I_Q (G_r - I). \end{aligned} \tag{25}$$

Directly using z_k in (24) has a drawback. For large distance errors, the constant gains in the controller causes unintended large orientation errors. Instead, consider the nonlinear scaled controller

$$\Phi_k = -L_z \begin{bmatrix} \frac{\sin(2\tilde{\theta}_k)}{2\tilde{\theta}_k} & 0 \\ 0 & 1 \end{bmatrix} z_k - L_r R_k. \tag{26}$$

This will reduce the control gain on d when $\tilde{\theta}$ becomes large and hence reduce the orientation error for large distance errors.

4.5 Scaling

The scaling approach from Section 3.2 is applied. The scaling vector $\Gamma = (\gamma_0, \ldots, \gamma_N)^T$ is selected such that

$$P \begin{bmatrix} 1 \\ \varphi_{k+n} \end{bmatrix} v_{\text{des}} \gamma_n = P' \gamma_n \leq q, \quad n = 0, \ldots, N, \tag{27}$$

is satisfied.

4.6 Reference

The predictive controller needs a vector, $\Psi_k = (\psi_k, \ldots, \psi_{k+N})^T$, with $N+1$ future orientation references, one for each prediction, such that the reference vector $R_k = (r_k, \ldots, r_{k+N})^T$ can be completed with elements $r_i = (0, \psi_i)^T$.

Straight Path—No Corner
We simply have

$$\Psi_k = \begin{bmatrix} \psi_1 & \cdots & \psi_1 \end{bmatrix}^T, \tag{28}$$

where ψ_1 is the orientation of the path.

Turning
Consider the turning illustrated in Figure 2. A difficulty exists since the reference vector is related to time but the turning is given by a position and orientation. Thus, we need to determine to what sampling instance the vehicle has reached the corner in order to relate the positional information of the corner to the reference vector. This is done by using information such as the initial distance to the corner and an estimate of the vehicles velocity.

The reference change is defined as the time k_{step} where the distances from the robot to the two path sections are equal, that is

$$d = \tan(\alpha)s \text{ or } d = \tan\left(\alpha + \frac{\pi}{2}\right)s. \tag{29}$$

Geometrically, (29) defines the two bisection lines shown in Figure 2. At the time k_{step}, the distance error d must change from being measured with respect to the incoming path section to the outgoing path section and s should then be directed towards the next corner, if any. At time k, define $\hat{\eta}_k$ as $k_{step} = k + \hat{\eta}_k$. The orientation reference vector is thus given as

$$\Psi_k = \begin{bmatrix} \mathbf{1}_{\hat{\eta}_k}(\psi_1) & \mathbf{1}_{N-\hat{\eta}_k+1}(\psi_2), \end{bmatrix}^T \tag{30}$$

where $\mathbf{1}_m(x) = (x, x, \ldots, x)^T$ is a vector of length m, and ψ_1 and ψ_2 are given by the orientation and direction of the corner. If for example the path is oriented along the x-axis with a left turn along the y-axis, then $\psi_1 = 0$ and $\psi_2 = \pi/2$.

Since the velocity changes due to the velocity scaling, the arrival of the corner and thus the sampling instance where reference should change, must be based on an estimation of the robot's posture. First, we discretize the model (2) using sampling period T

$$\begin{aligned} s_{k+1} &= s_k + Tv_k \cos\left(\theta_k - \psi_k + \frac{T\omega_k}{2}\right) \\ d_{k+1} &= d_k + Tv_k \sin\left(\theta_k - \psi_k + \frac{T\omega_k}{2}\right) \\ \theta_{k+1} &= \theta_k + T\omega_k. \end{aligned} \tag{31}$$

This model's n-step predictor is easily found by iterating the equations

$$\begin{aligned} \hat{\theta}_{k+n|k} &= \theta_k + T\sum_{i=k}^{n-1} \omega_i \\ \hat{d}_{k+n|k} &= d_k + T\sum_{i=k}^{n-1} v_i \sin\left(\theta_k - \psi_k + \frac{T}{2}\omega_i + T\sum_{j=k}^{i-1}\omega_j\right) \\ \hat{s}_{k+n|k} &= s_k + T\sum_{i=k}^{n-1} v_i \cos\left(\theta_k - \psi_k + \frac{T}{2}\omega_i + T\sum_{j=k}^{i-1}\omega_j\right). \end{aligned} \tag{32}$$

From (29) and (32), we can estimate $\hat{\eta}_k$. For $\psi_2 - \psi_1 > 0$ we get

$$\hat{\eta}_k = \min \left\{ n \left| \hat{d}_{k+n|k} \geq \tan\left(\alpha + \frac{\pi}{2}\right) \hat{s}_{k+n|k} \right.\right. \\ \left.\left. \vee \ \hat{d}_{k+n|k} \leq \tan(\alpha) \hat{s}_{k+n|k} \right\}, \quad (33)$$

while for $\psi_2 - \psi_1 < 0$ the inequality signs in (33) are opposite.

This concludes the predictive receding horizon controller defined by the control law (26), the scaling (27), the estimation of $\hat{\eta}$ in (33), and the reference vector (30).

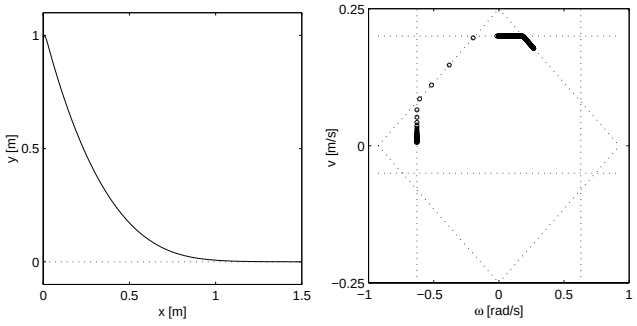
5 Experimental Results

This section verifies the usefulness of the approach by means of a simulation study. The parameters are chosen as $T = 0.04$s, $N = 100$, $\lambda = 0.0001$, and $Q = \left[\begin{smallmatrix} 1 & 0 \\ 0 & \delta \end{smallmatrix}\right]$, with $\delta = 0.02$ if not stated otherwise. The prediction horizon is equivalent to detecting a corner 0.8m before reaching it, desired speed assumed. The velocity constraints on the robot are chosen as

$$\begin{array}{rccccl} \text{wheels:} & -0.25 & \leq & v_{r,l} & \leq & 0.25 \quad [\text{m/s}], \\ \text{forward:} & -0.05 & \leq & v & \leq & 0.20 \quad [\text{m/s}], \\ \text{angular:} & -\frac{2\pi}{10} & \leq & \omega & \leq & \frac{2\pi}{10} \quad [\text{rad/s}]. \end{array}$$

5.1 Straight Path

Consider the task of following a wall with $\psi_1 = 0$ and initial starting point in $(0, 1\text{m}, 0)^T$. The desired velocity is 0.2m/s. Figure 4 shows the trajectory and the scaled controller outputs along



(a) xy-trajectory (b) Velocity scaling

Figure 4. Straight path with velocity constraints and scaling.

with the constraints. The resulting xy-trajectory for the unconstrained and constrained closed-loop system are equal; only the time response is different. For the unconstrained controller, $x = 1$ is reached after 7.6 seconds while for the constrained controller, the time is 9.56 seconds due to the scaling.

5.2 Turning

To start out a 90° turning is considered. Figure 5 shows the xy-trajectories for number of different initial starting positions and indicates a good robustness to initial conditions. It is seen how the trajectories initially seek the first section of the path and later change to follow the second part.

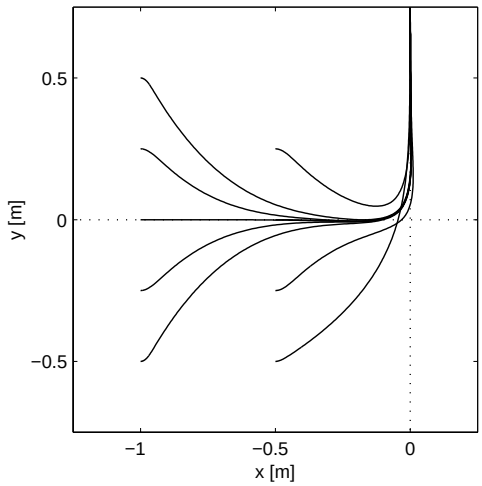


Figure 5. Path following for different initial positions.

Now, consider different turnings with $\psi_1 = 0°$ and $\psi_2 = 30°, 60°, 90°, 120°$, and $150°$, respectively. Figure 6 shows the xy-trajectories for the different turnings where the exact same parameter setting has been used for all the turnings. This demonstrates a good robustness. Even for very sharp turnings ($\psi_2 \geq 120°$) there is hardly any overshooting.

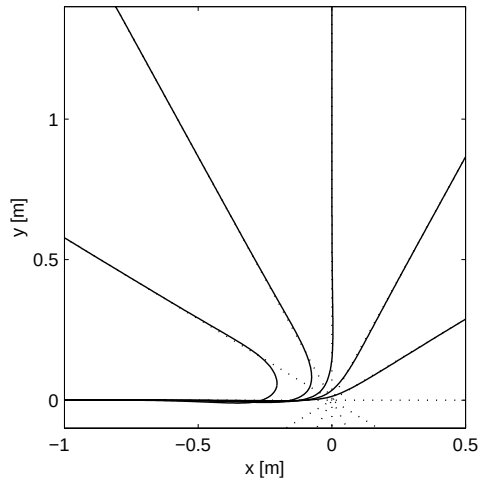
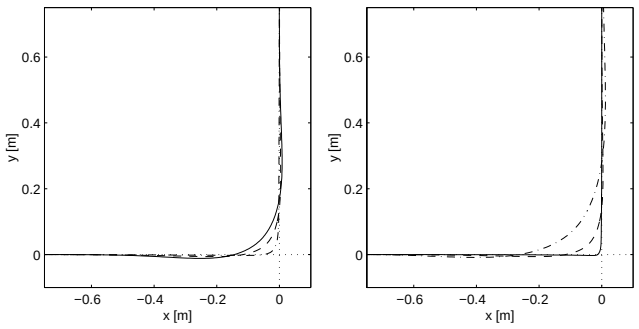


Figure 6. Path following for different turnings.

5.3 Parameter Robustness

Next, we consider parameter robustness and tuning capabilities. Figure 7 shows a 90° turning with different values of δ and λ (small, medium, and large values). The variations of the two parameters δ and λ have similar effects on the closed-loop xy-trajectory. In particular, for large values of λ or small values of δ the response tends to overlapping the path sections at all time. This is possible in spite of the velocity constraints because the forward velocity is allowed to tend to zero at that point. The drawback is the deceleration of the vehicle being very abrupt.



(a) Varying δ, fixed λ (b) Varying λ, fixed δ

Figure 7. Different values of δ and λ with small (solid), medium (dashed), and large (dash-dotted) values.

6 Conclusions

The key results in this paper are the following.

- A receding horizon approach provides a useful way of closed-loop controlling a mobile robot through sharp turnings.
- Online velocity scaling is an easy way of respecting velocity constraints without a significant increase in the computational burden.
- The presented algorithm is simple and fast and is easily implemented into a robot control system.
- The simulation study indicates good robustness to the degree of turning and the initial starting position.
- Parameter tuning allows to specify the smoothness of the xy-trajectory near the intersection point.

A drawback of the presented approach is the lack of optimality in the constrained control problem. However, results reveal good performance and the benefits from an applicable fast real-time algorithm are obvious and significant.

References

Bemporad, A., Marco, M. D. & Tesi, A. (1997), Wall-following controllers for sonar-based mobile robots, *in* 'Proceedings of the 36th IEEE Conference on Decision and Control', San Diego, California, pp. 3063–3068.

de Wit, C. C., Siciliano, B. & Bastin, G. (1996), *Theory of Robot Control*, Springer-Verlag, London.

Dickmanns, E. & Zapp, A. (1987), Autonomous high speed road vehicle guidance by computer vision, *in* 'Proceedings of the 10th IFAC World Congress', Vol. 4, München, Germany, pp. 232–237.

Ferruz, J. & Ollero, A. (1998), Visual generalized predictive path tracking, *in* 'Proceedings of the 5th International Workshop on Advanced Motion Control '98', Coimbra, Portugal, pp. 159–164.

Koh, K. & Cho, H. (1999), 'A smooth path tracking algorithm for wheeled mobile robots with dynamic constraints', *Journal of Intelligent and Robotic Systems: Theory and Applications* **24**(4), 367–385.

Normey-Rico, J., Gómez-Ortega, J. & Camacho, E. (1999), 'Smith-predictor-based generalized predictive controller for mobile robot path-tracking', *Control Engineering Practice* **7**(6), 729–740.

Sampei, M., , Tamura, T., Itoh, T. & Nakamichi, M. (1991), Path tracking control of trailer-like mobile robot, *in* 'Proceedings of IEEE/RSJ International Workshop in Intelligent Robots and Systems', Vol. J, Osaka, Japan, pp. 193–198.

Samson, C. (1992), Path following and time-varying feedback stabilization of wheeled mobile robot, *in* 'Proceedings of the International Conference on Advanced Robotics and Computer Vision', Vol. 13, Singapore, pp. 1.1–1.5.

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**, 64–77.

Samson, C. & Ait-Abderrahim, K. (1991), Feedback control of a nonholonomic wheeled cart in cartesian space, *in* 'Proceedings of IEEE International Conference on Robotics and Automation', Sacramento, California, pp. 1136–1141.