The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Online Smooth Trajectory Planning for Mobile Robots by Means of Nonlinear Filters

Marcello Bonfè and Cristian Secchi

Abstract— **The paper presents a nonlinear filtering technique that can be adopted to generate smooth trajectories for mobile robotic applications. The proposed trajectory planner can be fully executed online by the robot control system, thanks to its inherently discrete-time behavior and to its limited computational requirement. The outputs of the nonlinear filter used as a trajectory planner include the derivatives of the desired position in the cartesian plane up to the third order. This allows the implementation of feedback linearization control schemes that can transform the dynamics of a mobile robot in a double chain of three integrators, exploiting the highest derivative of the filter's output as a feedforward action. Finally, the paper reports experimental results obtained by the full implementation of the proposed trajectory planning and control scheme on a real unicycle-like robot.**

I. INTRODUCTION

The problem of smooth trajectory planning is a fundamental issue in robotics. When mobile robots subject to nonholonomic constraints are considered, the planning problem is complicated by the requirement on geometrical admissibility of the paths, while smoothness is necessary to keep the system within the conditions assumed for simplified modeling (e.g. absence of wheels slipping). In general, motion planning is usually separated into the geometric problem (*path planning*), whose solution may be a parametric path depending on an unspecified timing law to become executable, and the actual *trajectory planning*, in which the timing law for a given path is designed.

If robot motion can be planned in advance, efficient, but computationally expensive, interpolation methods (i.e. spline curves [1]) can be easily adopted for path planning and constrained optimization techniques can be used for timing law specification [2] (Chap. 4 and 7). However, when the robotic task is not fully known a priori, online adaptation or even complete re-planning of the desired path/trajectory is required [3]. Of course, online motion planning is much more computationally demanding, since it must be executed in real-time.

In mobile robotics, the geometric planning problem and, in particular, the search for shortest admissible paths in the configuration space have been intensively studied for many years. In fact, this subject is covered by a number of books and reviews [4] (Chap. 1), [5]. The avoidance of obstacles in the operational space is another typical issue of mobile robotics, whose solution may rely, for example, on Voronoi diagrams [6], Probabilistic Roadmaps [7] or artificial potential fields [8]. Finally, the search for time-optimal motion plans, either on specified paths [9], [10] or directly in the configuration space [11], is the last task required to allow a mobile robot to reach its goals with the highest efficiency, provided that perfect trajectory tracking is guaranteed by closed-loop control.

Many navigation strategies produce the desired path in terms of a set of via-points, that have to be crossed by the mobile robot in order to obtain a collision free motion to a predefined goal in the workspace [5]. The path that the robot has to track to cross the via-points has to be computed by a low level navigation system that needs to take into account the velocity and the acceleration constraints of the robot. It has been shown in [12] that the minimum length path connecting two points in a planar workspace is a sequence of straight lines and circular arcs. Unfortunately, such a kind of paths cannot be perfectly tracked because they induce a discontinuity in the acceleration, which cannot be achieved in practice. To avoid this problem, several path smoothing strategies have been introduced (see e.g. [13], [14], [15]). These techniques produce a path characterized by a countinuous curvature that can be tracked with bounded acceleration. However, path smoothing is an operation whose computational burden can be high [16] and, therefore, not suitable for low cost applications, where cheap computational platforms are required. Velocity and acceleration bounds have also been considered by [17], where the nonholonomic deformation method is extended in order to take into account kinematic constraints. With all of these approaches, only a trackable path is generated, so that further computational efforts must be spent to determine a feasible motion profile (in terms of velocity and acceleration) through which the robot can follow the geometric path. In summary, at the best of the authors' knowledge, most of the algorithms proposed in literature for path planning and timing-law optimization require a powerful computational platform, in order to be effectively executed online.

In this paper, we propose a trajectory planning solution for planar mobile robots that is designed for online execution, thanks to its limited computational demand and to its discrete-time behavior. Moreover, the output of the proposed trajectory planner is fully specified with respect to time. The solution is based on a nonlinear filter that generates a smooth trajectory, with continuous curvature, in cartesian coordinates of the operational space. The path of the robot does not need to be specified a priori, but it can be defined by setting a number of fixed via-points or a reference point

M. Bonfè is with the Engineering Department (ENDIF), University of Ferrara, 44100 Ferrara, Italy. E-mail: marcello.bonfe@unife.it

C. Secchi is with the Deparment of Science and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, 42100 Reggio Emilia, Italy. E-mail: cristian.secchi@unimore.it

moving along non-smooth paths. Both kind of references may come from higher-level planning algorithms, running at a slower sampling rate, implementing obstacle avoidance or goal reaching tasks. The nonlinear filter will transform these non-smooth references into trajectories that can be perfectly tracked by a unicycle-like robot with bounded linear/angular velocity and acceleration. Moreover, limitation on the third order derivatives of generated trajectories makes the approach suitable for autonomous wheelchairs or similar applications, in which the comfort of humans is a prominent issue.

The rest of the paper is organized as follows: Section II introduces the basic theoretical framework of nonlinear filtering for smooth trajectory generation, Section III describes the proposed trajectory planning scheme for mobile robotics applications, Section IV gives some details about unicycle-like robot with feedback linearization, that theoretically allows perfect trajectory tracking. The proposed algorithm has been fully implemented on a low-cost DSP-based motion control card and tested on a real differential-drive platform. Section V reports the experimental results.

II. NONLINEAR SMOOTHING FILTERS

Several approaches to online trajectory generation have been proposed in the literature on robotics. An interesting solution is the one based on nonlinear filtering. Considering, for example, basic positioning tasks, it is clear that a trajectory that reach a fixed set-point can be viewed as the output of a low-pass filter, whose input is a step signal with amplitude equal to the desired final position. Nonlinear design of the filter is required to guarantee bounded output derivatives. This principle has been exploited first in [18],with a strong emphasis on algorithmic or decision-tree based solutions. The approach proposed in [19], instead, is soundly based on control theory, which allows formal proof of its time-optimality and no-overshoot behavior. The smooth trajectory generator designed in [19] is a Variable Structure (VS) dynamic system that acts as a nonlinear filter for rough (steps, discontinuous ramps, etc.) reference position signals. Thanks to the design of the filter, perfect tracking of the reference signal can be achieved in minimum time, compatibly with constraints on the first and second derivative of the filter output. The VS system is composed by a chain of two integrators and a nonlinear controller that guarantees the requirement on bounded output derivatives (i.e. velocity and acceleration) and minimum time response. The filtering method can be realized with either a continuous-time or a discrete-time system. Since in robotics a digital control system is generally expected, only the discrete-time case, whose block diagram is shown in Figure 1, will be recalled here. The two discrete-time integrators have a different structure: this choice is necessary to guarantee the same dynamic behavior of a continuous time system when the control input is constant [19].

The VS controller receives at each sampling instant nT the following inputs: the rough reference signal r_n and its derivative $\dot{r}_n$, the bounds U on the acceleration/deceleration

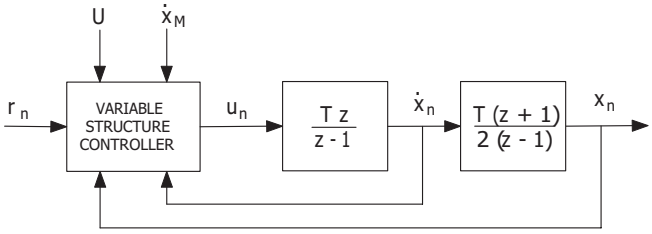

Fig. 1. Block diagram of a nonlinear filter for trajectory generation

and $\dot{x}_M$ on the velocity absolute value, the current outputs of the integrators: $\dot{x}_n$ and x_n. The control law proposed in [19], designed according to the principles of Sliding Mode [20], is the following:

$$u_n = -U\mathrm{sat}(\sigma_n)\frac{1+\mathrm{sign}[\dot{x}_n\,\mathrm{sign}(\sigma_n)+\dot{x}_M - T\,U]}{2} \quad (1)$$

in which sat(.) and sign(.) are the standard saturation and signum functions and, denoting with $\mathrm{Int}\,[.]$ the integer part of a number:

$$\sigma_n = \dot{z}_n + \frac{z_n}{m} + \frac{m-1}{2}\,\mathrm{sign}(z_n) \quad (2)$$

$$m = \mathrm{Int}\left[\frac{1+\sqrt{1+8|z_n|}}{2}\right] \quad (3)$$

$$z_n = \frac{1}{T\,U}\left(\frac{y_n}{T}+\frac{\dot{y}_n}{2}\right), \quad \dot{z}_n = \frac{\dot{y}_n}{T\,U} \quad (4)$$

$$y_n = x_n - r_n, \quad \dot{y}_n = \dot{x}_n - \dot{r}_n \quad (5)$$

The function σ depends on the tracking error y_n and the velocity error $\dot{y}_n$, normalized into z_n and $\dot{z}_n$ by means of the state-space transformation of Eq.(4). The condition $\sigma = 0$ defines a sliding surface (i.e. an invariant and attractive state subspace) in the error phase plane. The control law guarantees that the trajectories in the error space approach and reach in minimal time the sliding surface, keeping bounded velocity and acceleration. Then, the sliding mode brings the system towards a perfect tracking condition, which is also achieved in minimal time. The role of Eq.(3) and of the saturation in Eq.(1) is to ensure that the sliding surface is reached without overshoot, while the last factor of the control law is required to constrain the behavior of the filter when the initial conditions lie in regions of the error phase plane in which the velocity limit is not satisfied.

Remark 1: The output of the filter when r_n is constant is actually a standard trapezoidal velocity profile [2] (Chap. 4). The great advantage of the approach is that r_n can be changed abruptly at any time and the filter will simply generate, sampling instant by sampling instant, a time-optimal and feasible trajectory to reach the newest set-point. Moreover, r_n need not to be piecewise-constant.

Remark 2: The execution of the filter requires only 10 multiplications/divisions, 10 additions/subtractions and the

execution of a square root, for the control law, plus two numerical integrations and few IF..THEN statements to implement sign(.) and sat(.). This means that its real-time implementation on even a standard low-cost DSP for motion control would be executed within a few microseconds.

III. SMOOTHING FILTERS FOR MOBILE ROBOTICS

The extension of the nonlinear filter proposed in previous section to a multidimensional system is not straightforward. In manipulation robotics, the approach can be directly applied in joint-space by simply decomposing the multidimensional filtering problem into N (one for each joint) independent one-dimensional problems. In mobile robotics, instead, the trajectories must be planned in the operational space, because of nonholonomy and obstacles. On the other hand, direct decomposition of the filtering problem into two separate one-dimensional problems for each cartesian coordinate is not possible, since their derivatives are related by nonholonomic constraints. In this section, we propose a practical solution based on the generation with nonlinear filters of the two components of desired velocity in cartesian coordinates, compatibly with kinematic and dynamic constraints, and subsequent numerical integration of these velocity components in order to obtain the desired position vector.

We focus on the class of robots that can be represented by the so-called unicycle-like kinematic model, subject to *rolling without slipping* constraint:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \tag{6}$$

in which θ is the orientation of the robot w.r.t to the fixed cartesian frame. The vector $[x, y, \theta]^T$ generally defines the robot configuration, subject to the following equations:

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{7}$$

in which v and ω, respectively *driving velocity* and *steering velocity*, are in most applications assumed as the control inputs. Considering only the first two rows of Eq.(7), it is clear that the generation of two sufficiently smooth signals $v(t)$ and $\theta(t)$ and the integration of $\dot{x}$ and $\dot{y}$ would allow to solve the trajectory planning problem in the cartesian plane, compatibly with the kinematic model of the unicycle.

In order to design the nonlinear filter to generate $v(t)$ and $\theta(t)$, it is necessary to recall the following preliminaries. The acceleration of a planar trajectory for the unicycle is given by the sum of *tangential* and *radial* acceleration orthogonal vectors, whose lengths are respectively $a_t = \dot{v}$ and $a_r = v\,\dot{\theta} = v\,\omega$. Both components must be bounded to preserve from the presence of lateral slipping, which invalidates the kinematic model, and to keep proportionality between wheels velocity and driving velocity. Of course, if the robot is not turning, v can be set to the maximum value allowed by the actuators. Limiting radial acceleration by reducing (without zeroing) the driving velocity when $\omega \neq 0$, involves a limitation also on the scalar curvature of the path ($\kappa = \omega/v = \omega^2/a_r$).

Since the main objective of trajectory planning is to reduce as much as possible both time and space necessary to reach a given target position, it seems reasonable to increase speed up to the limit, when the robot is oriented towards the target, and instead set the driving speed as the ratio between maximum allowed radial acceleration and maximum steering velocity, when the robot must change its orientation to put the target within its "line of sight". Next, it is necessary to understand how to execute this change of orientation, by means of nonlinear filtering. For this purpose, we can recall the so-called *planar pursuit-evasion* equations, that are typically used for missile-guidance algorithms design [21]. Referring to Figure 2, in which $[x_t, y_t]^T$ is the position of the target and v_t its velocity, $[x_d, y_d]^T$ and v_d are the outputs at a given instant of the trajectory generator (i.e. the position/velocity of the "missile"), pursuit-evasion equations can be written as follows:

$$\begin{aligned} R &= \sqrt{(x_t - x_d)^2 + (y_t - y_d)^2} \\ \theta_e &= \mathrm{ATAN2}\,(y_t - y_d,\, x_t - x_d) \\ \dot{\theta}_e &= \tfrac{1}{R}\left[v_d \sin(\theta_e - \theta_d) - v_t \sin(\theta_e - \theta_t)\right] \end{aligned} \tag{8}$$

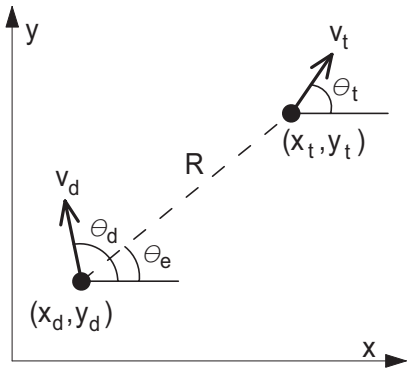

Fig. 2. Target interception geometry

From now on we will assume $v_t = 0$ (i.e. fixed target positions). However, the following remarks can be easily extended to moving targets. If the orientation of the trajectory generated by the filter is different from θ_e, then this value should be set as the (time-varying) reference for the nonlinear filter. During the turning phase necessary to align with the target, the bound on steering velocity should be the highest possibile, while the reference for driving velocity must be set to maximum value that allows limitation of radial acceleration. Once that target alignment is achieved and the trajectory is pointing with maximum driving velocity towards the final position, the latter can be reached without overshoot by simply triggering a deceleration phase with zero final velocity, as soon as the distance from the target becomes

equal to the space required to stop with given bounds on $\dot{v}$ and $\ddot{v}$.

Starting from these remarks, we can design a nonlinear smoothing filter, to generate trajectories for a unicycle-like robot, by means of the system whose block diagram is described in Fig. 3. The filter is a discrete-time system, but in the rest of the section explicit reference to the sampling instant nT will be dropped, to simplify the notation, and we will improperly refer to differentiation and integration in the same way it is usually done for continuous-time operations.

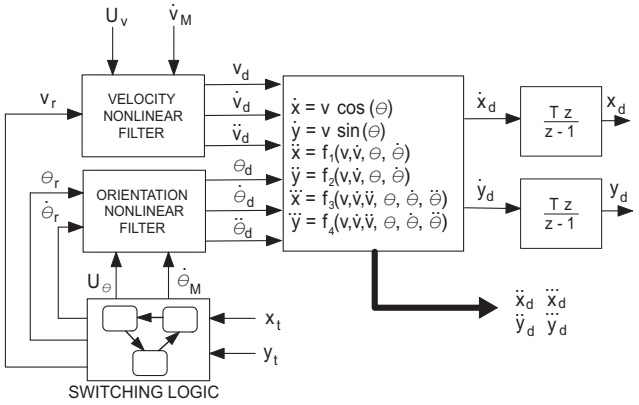

Fig. 3. Block diagram of the nonlinear smoothing filter for mobile robotics

The blocks in Fig. 3 can be described as follows:

- *Velocity nonlinear filter*: a filter with the structure of Fig. 1, but its output is a driving velocity v_d (*instead* of a desired position), that perfectly tracks the reference velocity v_r with bounds on first and second derivatives ($|\dot{v}_d| \leq \dot{v}_M$ and $|\ddot{v}_d| \leq U_v$).
- *Orientation nonlinear filter*: a filter that differs from the previous one only in the calculation of the tracking error, since the difference between two angles must be limited in the interval $[-\pi; \pi]$ and its sign must take into account the shortest distance. The output of the filter is the desired orientation θ_d tracking at best θ_e with bounded derivatives ($|\dot{\theta}_d| \leq \dot{\theta}_M$ and $|\ddot{\theta}_d| \leq U_\theta$).
- *Switching logic*: the reference signals v_r, θ_r and $\dot{\theta}_r$ ($\dot{v}_r = 0$ at any time) and the bounds of the orientation filter are specified according to a target approaching sequence that can be described by the state machine shown in Fig. 4.

  The state diagram refers to the following parameters:
  - v_M: maximum allowed driving velocity;
  - A_{RM}: maximum allowed radial acceleration;
  - $\dot{\theta}_B$: maximum allowed steering velocity;
  - R_{stop}: distance required to decelerate from $v_r = v_M$ to $v_r = 0$. This value can be easily calculated since the output of the velocity filter, assuming $\dot{v}_r = 0$,is a standard profile with trapezoidal first derivative (i.e. acceleration, in this case). Integrating further this velocity profile, we obtain R_{stop}.
- *Final calculation of derivatives and integration*: the trajectory generated by the filter needs to be expressed in cartesian coordinates. Therefore, the outputs of the two nonlinear filters previously described must be combined as follows:

$$\begin{aligned} \dot{x}_d &= v_d \cos\theta_d \\ \dot{y}_d &= v_d \sin\theta_d \end{aligned} \tag{9}$$

  Integrating Eq.(9) we finally obtain $[x_d, y_d]^T$. Higher order derivatives of the trajectory can be directly obtained from the outputs of velocity and orientation nonlinear filters, using functions resulting from differentiation of the first two rows of Eq.(7).

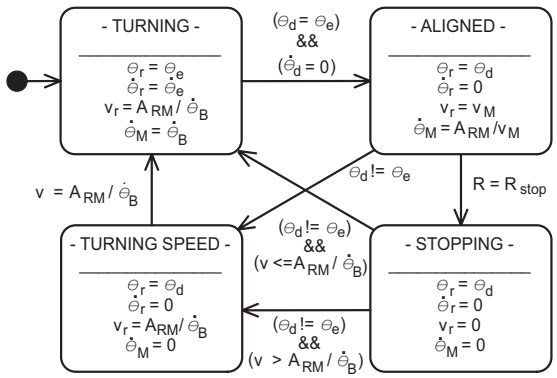

Fig. 4. Switching logic for setting references and bounds of the proposed nonlinear filter

Remark 3: Target position $[x_t, y_t]^T$ can be abruptly changed at any time, causing the condition $\theta_d \neq \theta_e$ and, therefore, forcing a reduction of driving velocity, set to $A_{RM}/\dot{\theta}_B$ before starting the turning phase. In this way, radial acceleration is guaranteed to be bounded by A_{RM}. Of course, this abrupt change of $[x_t, y_t]^T$ may be executed by a higher-level planning algorithm, that switches between different via-points when $[x_d, y_d]^T$ is in their vicinity.

Remark 4: The second order derivatives of velocity and orientation filters are bounded, so that also third order derivatives of $[x_d, y_d]^T$ are limited. Moreover, since second order derivatives are continuous, the curvature of the resulting path is continuous. Therefore, the trajectories generated by the filter are suitable for applications in which the comfort of humans transported by a mobile robot (e.g. autonomous wheelchairs) is of interest.

Now that the trajectory generator is fully specified, it is useful to address the issue of perfect trajectory tracking under closed-loop control. As is well-known in robotics, this condition can be achieved by means of dynamic inversion or feedback linearization techniques, provided that the model of the controlled system is perfectly known. In particular, we can observe that the vector $[x_d, \dot{x}_d, \ddot{x}_d, y_d, \dot{y}_d, \ddot{y}_d]^T$ corresponds to the desired value of the state (in *normal coordinates*) of a unicycle-like robot controlled by a dynamic feedback linearization loop, as described in [22]. In the following section, we extend the approach of [22] in order to obtain third-order feedback linearization of the unicycle model, that allows for perfect tracking of the trajectory generated by the proposed nonlinear filter.

IV. DYNAMIC FEEDBACK LINEARIZATION

In [22] the kinematic model of Eq.(7) is I/O feedback linearized by choosing $\eta = [x, y]^T$ as the output and adding an integrator before the original input v, so that the I/O decoupling matrix (see [23], Chap. 5) becomes nonsingular (if the state of the integrator $\xi \neq 0$). The integrator is part of the dynamic compensator and its input a is part of the new input vector $u = [a, \omega]^T$, which allows exact linearization by means of a control law based on the inverse of the I/O decoupling matrix. The final result of the procedure is that $\ddot{\eta} = \nu = [\nu_1, \nu_2]^T$ (i.e. the system is transformed in a double chain of two integrators).

This procedure can be extended in order to take care of the dynamics of a unicycle-like robot, whose model can be written as follows (see [2], Par. 11.4):

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{v} &= F\,/\,m \\ \dot{\theta} &= \omega \\ \dot{\omega} &= \tau\,/\,J \end{aligned} \tag{10}$$

in which F is the *driving force*, τ is the *steering torque*, m is the total mass and J is its moment of inertia around the vertical axis. Applying the linear input transformation $a = F/m$ and $\alpha = \tau/J$ and adding an integrator before a, such that:

$$a = \xi \qquad \dot{\xi} = j \tag{11}$$

the I/O feedback linearization can be obtained by subsequent differentiation of the output $\eta = [x, y]^T$ until a nonsingular relationship with the input is obtained. This happens when the third derivative of η is computed:

$$\begin{aligned} \dddot{\eta} &= F(q) + B(q)u \\ q &= [x, y, v, \xi, \theta, \omega]^T, \quad u = [j, \alpha]^T \\ F(q) &= \xi\omega \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} + v\omega \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} - v\omega^2 \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \\ B(q) &= \begin{bmatrix} \cos\theta & -v\sin\theta \\ \sin\theta & v\cos\theta \end{bmatrix} \end{aligned} \tag{12}$$

Starting from Eq.(12), we can design the linearizing control law as:

$$u = B^{-1}(q)(\nu - F(q)) \tag{13}$$

so that $\dddot{\eta} = \nu$. Notice that $B(q)$ is invertible as long as $v \neq 0$, so that the remarks of [22] on this singularity must be considered.

Remark 5: The state in normal coordinates of the linearized system is $z = [x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}]^T$. Given a reference trajectory whose derivatives are known up to the third order, the control input ν can be calculated as follows:

$$\nu = \begin{bmatrix} \dddot{x}_d \\ \dddot{y}_d \end{bmatrix} + K\,(z_d - z) \tag{14}$$

in which K is any 2×6 constant matrix that stabilizes the control-loop on the linearized system. This control law achieves perfect trajectory tracking after a transient depending on the closed-loop eigenvalues, which can be arbitrarily placed by means of a proper design of K.

V. EXPERIMENTAL RESULTS

The proposed nonlinear filter has been tested on a mobile robotic platform completely in-house developed. The robot has a differential-drive mechanical structure, with one castor wheel and two wheels actuated by DC motors generating a peak torque of 2 Nm at 2000 RPM. The structure is realized in aluminium and its total weight $m = 15$ kg. Since the robot has a rectangular shape (350x450 mm), the moment of inertia has been estimated as $J = 0.4$ kg m^2.

The trajectory planning algorithm and the closed-loop controller described in previous sections have been implemented on a motion control card specifically developed for the robot, based on a dsPIC30F produced by Microchip Technology, which is a 16-bit Fixed-Point Digital Signal Controller (DSC) running at 30 MIPS. The nonlinear filter is fully computed with 32-bit resolution by the DSC in less than 800 μs. As a rough comparison, the path planner described in [16] requires 15 ms on a Pentium IV at 2.2 GHz. On the other hand, the sampling time for both the nonlinear filter and the control law has been set to $T = 4$ ms during the experiments reported next, in order to apply a conservative choice.

The proposed trajectory planner has been tested on a sequence of fixed via-points that were set around an hypothetical square obstacle. Each time that the filter output $[x_d, y_d]$ was at a distance slightly larger than R_{stop} from the current reference point, the latter was switched to a newer entry in the list of via-points. Fig. 5 shows the output of the nonlinear filter (green) in the cartesian space, obtained setting $v_M = 0.25$ m/s, $\dot{v}_M = 0.4$ m/s^2, $U_v = 2$, $A_{RM} = 0.1$ m/s^2, $\dot{\theta}_M = 0.6$ rad/s, $U_\theta = 2$ rad/s^2. The behavior of the robot during the test sequence can also be observed in an accompanying video. As can be seen, the speed of the robot is actually limited with respect to its capacity. On the other hand, these limits are quite similar to those mentioned in [22]. Again, this choice is conservative and is motivated by the requirement on wheel slippage avoidance.

The trajectory tracking is referred to the robot pose estimation (red line plus blue triangles showing orientation) obtained only with standard encoder-based odometry. The tracking error in cartesian coordinates is shown in Fig. 6. As can be noticed also in the accompanying video, trajectory tracking is affected by slight oscillations. This fact is not surprising and is related to the presence of unmodeled dynamics (DC motors) and mismatch in the dynamic parameters, which means that feedback linearization is not exact.

VI. CONCLUSION AND FUTURE WORK

The paper has described an approach to trajectory planning for mobile robots based on the theory of nonlinear smoothing filters. The nonlinear filter designed in this project is able to generate reference trajectories in the cartesian plane, with

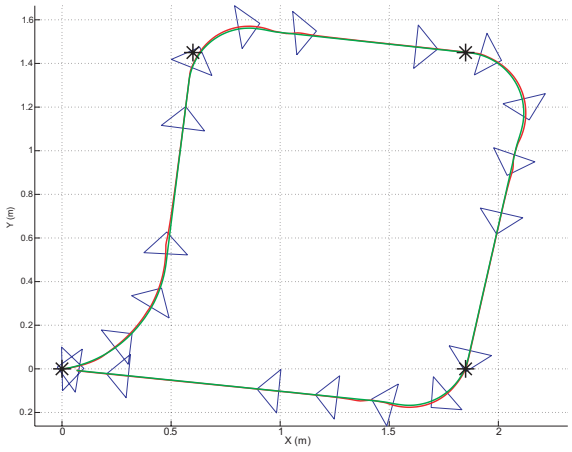

Fig. 5. Experiment on a real differential-drive robot: trajectory planned (green), trajectory tracked (red) and via-points (stars)

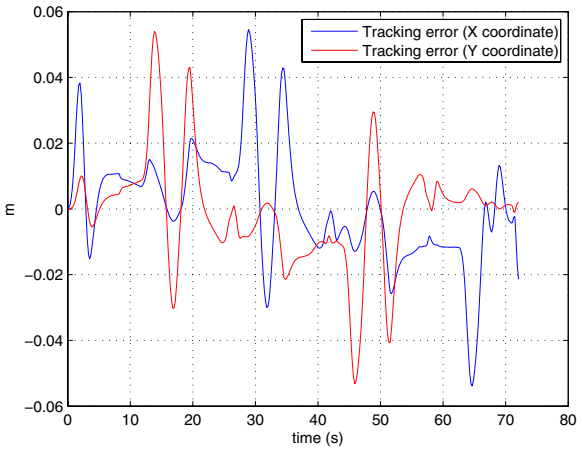

Fig. 6. Tracking error in the cartesian space

known derivatives up to the third order and continuous curvature, that are inherently compatible with kinematic and dynamic constraints of a classical unicycle-like robot. The trajectories obtained with the proposed approach can be ideally tracked without steady-state error if the control system is implemented by means of I/O linearization with dynamic state feedback.

It is important to remark that full digital implementation of the trajectory planner can be executed even on a standard low-cost microcontroller or DSP. Therefore, online applications do not require any oversized computational platform. Only a higher-level planning system, running at a slower sampling rate, that outputs rough references by means of fixed via-points or reference points moving along non-smooth paths, is required to support the robotic platform with obstacle avoidance or goal reaching capabilities.

In future works we aim to formalize the properties of the filter, in terms of length of the generated paths and total traveling time, and compare them with the results described in the literature. Moreover, the implementation of dynamic feedback linearization controller will be calibrated better, in order to improve tracking performance.

REFERENCES

[1] C. DeBoor, *A Practical Guide to Splines*. Springer-Verlag, 1978.

[2] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modeling, planning and control*, ser. Advanced Textbooks in Control and Signal Processing. Springer-Verlag, 2009.

[3] S. Macfarlane and E. Croft, "Jerk-bounded manipulator trajectory planning: design for real-time applications," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 1, pp. 42–52, February 2003.

[4] J.-P. Laumond, Ed., *Robot Motion Planning and Control*. Berlin: Springer-Verlag, 1998.

[5] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[6] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2000.

[7] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, August 1996.

[8] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[9] J.-Y. Fourquet and M. Renaud, "Time-optimal motions for a torque controlled wheeled mobile robot along specified paths," in *Proc. of 35th Conference on Decision and Control*, Kobe, Japan, December 1996.

[10] C. Guarino Lo Bianco and M. Romano, "Optimal velocity planning for autonomous vehicles considering curvature constraints," in *Proc. of IEEE Conference on Robotics and Automation*, Roma, Italy, April 2007.

[11] M. Renaud and J.-Y. Fourquet, "Minimum time motion of a mobile robot with two independent acceleration-driven wheels," in *Proc. of IEEE Conference on Robotics and Automation*, Albuquerque, New Mexico, April 1997.

[12] J. Laumond, J. Jacobs, M. Taix, and M. Murray, "A motion planner for nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 577–593, 1994.

[13] T. Fraichard and A. Scheuer, "From Reed and Shepp's to continuous-curvature paths," *IEEE Trans. on Robotics*, vol. 20, no. 6, pp. 1025–1035, December 2004.

[14] G. Parlangeli, L. Ostuni, L. Mancarella, and G. Indiveri, "A motion planning algorithm for smooth paths of bounded curvature and curvature derivative," in *Proc. of 17th Mediterranean Conference on Control*, June 2009, pp. 73–78.

[15] M. Kanehara, S. Kagami, J. Kuffner, S. Thompson, and H. Mizoguhi, "Path shortening and smoothing of grid-based path planning with consideration of obstacles," in *Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics*, October 2007, p. 991//996.

[16] N. Montes, A. Herraez, L. Armesto, and J. Tornero, "Real-time clothoid approximation by rational bezier curves," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, May 2008, pp. 2246–2251.

[17] M. Hillion and F. Lamiraux, "Taking into account velocity and acceleration bounds in nonholonomic trajectory deformation," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, April 2007, pp. 3080–3085.

[18] J. Lloyd, "Trajectory generation implemented as a non-linear filter," University of British Columbia, Computer Science Department, Tech. Rep. TR–98–11, August 1998.

[19] R. Zanasi, C. Guarino Lo Bianco, and A. Tonielli, "Nonlinear filters for the generation of smooth trajectories," *Automatica*, vol. 36, p. 439/448, 2000.

[20] V. Utkin, "Variable structure systems with sliding modes," *IEEE Transactions on Automatic Control*, vol. 41, no. 4, 1977.

[21] C. Lin, *Modern Navigation, Guidance and Control Processing*. Prentice-Hall, 1992.

[22] G. Oriolo, A. De Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: Design, implementation and experimental validation," *IEEE Trans. on Control Systems Technology*, vol. 10, no. 6, pp. 835–852, November 2002.

[23] A. Isidori, *Nonlinear Control Systems*, 3rd ed. Springer-Verlag, 1995.