A control method for stable and smooth path following of mobile robots

Kristijan Maček*, Ivan Petrović†, Roland Siegwart*

*Swiss Federal Institute of Technology Lausanne, Switzerland
email: {kristijan.macek, roland.siegwart}@epfl.ch

†University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia
email: ivan.petrovic@fer.hr

***Abstract*— This paper addresses the problem of mobile robot navigation in indoor cluttered environments. A new algorithm for both longitudinal and lateral real-time control of wheel-based mobile robots has been proposed. Its main characteristic is smooth and stable following of the on-line replanned path. Our control method is actually extension of the virtual vehicle method proposed by M. Egerstedt et al., which is based on the introduction of a look-ahead point on the path that serves as reference point for the control algorithm. While the original virtual vehicle method uses only error feedback for reference point movement along the path, our method uses also a feedforward component based on path curve characteristics between the robot and the reference point. In this way stable robot movement is achieved also in the presence of obstacles that are critical with respect to the path following error. Experimental verification is provided for a differential drive robot within an on-line global path planning framework.**

I. Introduction

The problem of controlling wheel-based mobile robots has been well studied in literature. Representatives of open-loop control methods to model-based closed-loop methods for lateral and/or longitudinal control can be found in [1], [2], [3], [4], [5], [6], [7], [8], [9]. However, a control algorithm that ensures stable and smooth robot motion in cluttered environments is still missing. In this paper an attempt to develop such an algorithm is described. The approach presented in this work relies on the virtual vehicle method (VV) in [10] that is a combination of trajectory tracking since the smooth desired path is parametrized in time and path following where the positional and orientation error of the robot is given with respect to a geometric path. An important feature of the virtual vehicle approach is that it is model-independent since the controls are implemented on the kinematic level and are equally applicable to e.g. differential-drive and car-like robots. While the original virtual vehicle method uses only error feedback for reference point movement along the path, the method proposed in this paper uses also the curve characteristics between the robot and the reference point called curvature effort (CE) which ensures stable robot movement in presence of obstacles. In Sec. II the original approach is shortly resumed with motivation to introducting the curvature effort based approach in Sec. III. The on-line path replanning framework is presented in Sec. IV followed by experimental results in Sec. V.

II. Virtual vehicle approach

A. Control algorithm statement

A virtual vehicle approach stated in [10] aims at finding a lateral control $\delta(t)$ and longitudinal control $v(t)$ of a robot to follow a smooth path $\vec{c}(s)$ where the desired reference point on the path is called a virtual vehicle. It is described by its parameter $s(t)$ moving along the path with coordinates:

$$\begin{aligned} x_d &= p(s) \\ y_d &= q(s), (0 \le s \le s_f) \end{aligned} \tag{1}$$

It is assumed that the curve gradient $\|\vec{c}\,'(s)\| = \sqrt{p'^2(s) + q'^2(s)} \neq 0, \forall s \in [0, s_f]$. The control objective is to keep the positional and lateral difference of the robot with the respect to the reference point on the path within specific bounds:

$$\lim_{t\to\infty} \sup \rho_t \le d_\rho \tag{2}$$

$$\lim_{t\to\infty} \sup \|\psi - \psi_d\| \le d_\psi \tag{3}$$

where $\rho = \sqrt{\Delta x^2 + \Delta y^2}$ and $\Delta x = x_d - x$, $\Delta y = y_d - y$ and (x, y) is a reference point on the robot with orientation ψ. The desired yaw angle is described as $\psi_d = atan2(\Delta y, \Delta x)$. The number d_ψ represents the maximum lateral tracking error depending mainly on the path curvature and the maximum angular acceleration of the robot. The number d_ρ is the look-ahead distance to the reference point and is determined in the control design procedure.

The control law proposed in [10] that ensures global stability determines the dynamics of the parametrized reference point $\dot{s}$ as:

$$\dot{s} = \frac{c_o e^{-\alpha\rho} v_n}{\sqrt{p'^2(s) + q'^2(s)}} \tag{4}$$

where v_n is the nominal speed at which the path is to be followed. Expression (4) gives an exponential decay of the positional tracking error. Factor α describes the responsiveness of the reference point to the change in distance to the robot. Choosing a greater α increases its responsiveness but introduces a larger variation in the translational velocity v of the robot. The velocity of the reference point along the curve is expressed as $s_v = \dot{s}||\vec{c}'||$. Parameter c_o can be determined by choosing that the velocity of the reference point be equal the nominal velocity v_n of the robot when in steady-state, i.e:

$$\rho(t) = d_\rho,\ v(t) = v_n \tag{5}$$

Thus c_o becomes $c_o = e^{\alpha d_\rho}$.

With the velocity of the reference point that adapts to the tracking error one can choose the translational and lateral control to be proportional regulators that steer the vehicle towards the reference point at a translational and rotational velocity proportional to the tracking error [10]:

$$\omega_{ref} = k\Delta\psi + \dot{\psi}_d \tag{6}$$

$$v_{ref} = \gamma\rho cos(\Delta\psi) \tag{7}$$

where $\Delta\psi = \psi_d - \psi$ and $\dot{\psi}_d$ is the rate of change of orientation to the reference point.

The factor γ can be determined according to the steady-state conditions in Eq. 5 to be $\gamma = \frac{v_n}{d_\rho}$. As already stated this control strategy is model independent and it provides high level commands v_{ref} and ω_{ref} based only on position and orientation feedback. Thus, low level velocity controllers are assumed to be implemented.

A simulated robot run according to an off-line planned path for the virtual vehicle approach is shown in Figs. 2, 4, 5, 6 and 11. Essentially, after the transient state the robot will try to track the path at the constant nominal velocity $v \approx v_n$ and look-ahead distance d_ρ mainly adjusting its steering angle to the orientation of the reference point. This can be seen from the simulation run for the choice of nominal speed $v_n = 0.5m/sec$ and look-ahead distance as $d_\rho = 1m$. Both robot and reference point velocity $v \approx 0.5m/sec$, $s_v \approx 0.5m/sec$ are kept almost constant at all times. Exceptions are the case of robot initial acceleration stage where the reference point is slowed down in order to "wait" for the robot and the case of extreme curvature at the doorway where the robot approaches the reference point that is turning at a distance $\rho < d_\rho$ resulting in speeding up of the reference point to attain the nominal d_ρ distance.

If for a particular reference path the dynamic and kinematic constraints of the robot are exceeded, the tracking performance will be increasingly degraded within robot limitations. This deviation from the planned path may be prohibitive in the presence of obstacles. This is the case in Fig. 2 where the robot grazed the inner corner to a wall and hit it due to an exceeded velocity when approaching a strong curvature - the upper room position in Fig. 2 is the goal position that was not reached in this case.

Varying the nominal distance to the reference point (i.e. the "line-of-sight" of the robot) does not solve the problem since the robot will still travel at an approximately nominal velocity after the transient state, as shown earlier. A solution may be to decrease the overall nominal velocity that may be achieved, however this is not an efficient solution and may fail in a more complex path configuration.

Therefore, it is mandatory to consider the velocity of the reference point and that of the robot not only according to the tracking feedback error but also according to the configuration of the planned path curve itself in some local vicinity. This is the motivation for introducing a modified virtual vehicle approach.

III. Curvature effort based virtual vehicle approach

A. Local curvature description - curvature effort

In order to preserve the same proportional control law as described in (6) and (7) the dynamic equation of the parameter $\dot{s}(t)$ must be changed according to some measure. A possible solution is to find a local curve description between the robot and the reference point according to the curvature along the planned path since it contains strong information about the curve characteristics. The curve itself is assumed to be smooth and C^2 continuous.

In general, if a curve $\vec{c}(s)$ is parametrized by parameter s according to the arc length along the curve (unit-speed curve) then the curvature $\kappa(s)$ of the curve can be described as [14]:

$$\kappa(s) = \frac{p'q'' - p''q'}{(p'^2 + q'^2)^{\frac{3}{2}}} \tag{8}$$

Its magnitude equals $||\vec{c}''(s)||$ and the sign is defined by the choice of the normal vector $\vec{N}$ at a point $\vec{p}(s)$ described by the parameter s. The curvature at the point $\vec{p}(s)$ is related to the osculating circle having a second order contact to the curve and a radius $R(s)$ as $\kappa(s) = \frac{1}{R(s)}$. It can be therefore stated that the curvature directly represents how much a curve bends locally.

The keypoint to the approach is the fact that a larger change in curvature brings the robot closer to its dynamic and kinematic limitations. From the kinematic equations it is known that if the robot travels at a certain translational velocity v and angular velocity ω than it describes a circular trajectory with the curvature being $\kappa_r = \frac{w}{v}$. By observing only the magnitude of the curvature of a curve κ_c and asserting that $\kappa_r = \kappa_c$ one can only conclude what the kinematic constraints may be in terms of maximum angular velocity at a given maximum translational velocity (the singular case when $v = 0$ is not considered since then the robot turns on spot). Since the curvature κ_c relates to the kinematics of robot motion, one must therefore consider the *change* in curvature to account for robot dynamics.

As an example, the curvature characteristics of a smooth desired path of robot is shown in Fig. 1. The current position of the reference point is at approximately $1m$ in front of the robot (curve parameter s representing the path length). As it can be seen, $\kappa(s)$ equals 0 along the straight line sections whereas in curved segments the magnitude of curvature corresponds to how much the curve bends (i.e. inverse of the radius of the osculating circle along the path). The sign of the curvature determines the direction of robot turn ($\kappa > 0$ is a turn to left and $\kappa < 0$ is a turn right in the robot frame, respectively). In order to circumvent the local obstacle configuration that formed the path the robot has to make turns both in negative and positive orientation. This fact is not taken into account explicitly in the original virtual vehicle approach since the dynamics of the reference point is governed only by the line-of-sight from the robot. Diminishing the nominal distance to the reference point could partially solve this problem. However, the increased relative longitudinal and angular error could bring the proportional control scheme to the stability margin.

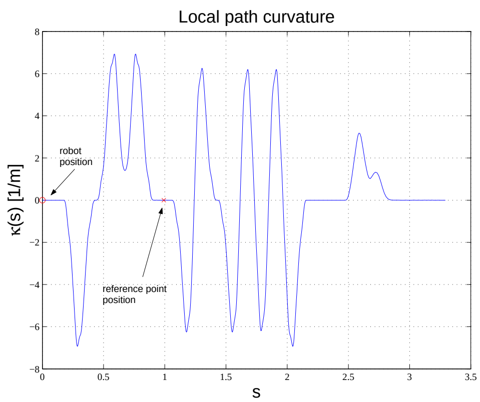

Fig. 1. Curvature of a smooth reference path

Therefore, to quantitatively describe the necessary robot activity on the reference path $\vec{c}(s)$ a term ***curvature effort*** is introduced here [11]:

$$\chi(\rho) = \frac{\sum_{i=1}^{N_\rho} \|\Delta\kappa_i\| \Delta s_i}{\rho}, \ \forall \rho > 0 \tag{9}$$

It is assumed that the reference path is described as a collection of dense waypoints $\left\{(x_d(s), y_d(s)) : s = (s_1, s_2 \ldots, s_{N_\rho})\right\}$ and N_ρ denotes the number of points up to position of the reference point ρ. The $\Delta\kappa_i = \kappa_i - \kappa_{i-1}$ is the incremental change in curvature and $\Delta s_i = s_i - s_{i-1}$ is incremental change in parameter s that may not be equally spaced. The expression (9) takes into account all curvature and therefore all direction changes along the local path up to the current reference point position. To equally account for curvature effort of all possible positions of reference point (depending on overall reference point speed) it is necessary to normalize (9) to the ρ distance. Clearly, if the curvature effort is $\chi(\rho) = 0$ then the robot is moving along a path with no curvature change, i.e. a straight line segment or a circular path and it may proceed at its maximum speed. For a straight line segment the angular velocity must equal $\omega = 0$ and for a circular path with curvature κ_c the angular velocity is $\omega = \kappa_c v$ if it is attainable according to the maximum angular velocity of the robot. With the increasing curvature effort the robot should decrease the translational velocity. The maximum curvature effort depends principally on the structure of the environment and the way the smooth path is constructed. Therefore, if the smooth path to follow can be generated with less curvature changes, the maximum expected curvature effort will be smaller.

B. Modified virtual vehicle dynamics

When there are significant curvature changes encountered (in both positive or negative orientation) then the robot must slow down to follow the path at a safe error margin. Since velocity of the robot is directly influenced by dynamics of the reference point, the following modification is proposed to the original expression in (4):

$$\dot{s} = \frac{c_o e^{-\alpha\rho} v_n}{\sqrt{p'^2(s) + q'^2(s)}} \left[1 - \frac{2}{\pi} atan(\zeta\chi(\rho))\right] \tag{10}$$

The term $P(\rho) = 1 - \frac{2}{\pi} atan(\zeta\chi(\rho))$ can be described as a *penalty factor* that causes the reference point to slow down in presence of curvature changes and it can attain a value $P(\rho) \in (0, 1]$. It introduces a feedforward component in the control scheme that indirectly models the robot movement along the path. The factor ζ essentially determines how fast the reference point will slow down, thus increasing the safety level for larger ζ values but also rendering the robot drive less efficient in terms of speed. The lower limit value of $\zeta = 0$ represents the original virtual vehicle dynamics. The stability analysis provided in [10] also applies here, since the dynamics of the reference point never exceeds that of the original case, therefore the exponential decay of the position tracking error is the same. For large curvature effort, the $atan$ function gets saturated causing the reference point to come practically to a stop. However, this does not cause the robot to come to stop but merely decreases its velocity for a safe following of the path. As the robot approaches the reference point at a decreased velocity, the curvature effort term $\chi(\rho)$ is diminished (i.e. the robot has traversed the critical path segment) which again causes the reference point to speed up and possibly attain the nominal velocity v_n if there is not path curvature change thereafter.

A simulated robot run for the curvature effort based virtual vehicle approach is shown in Figs. 3, 7, 8, 9, 10 and 12. The dashed line in Fig. 3 represents the off-line planned path and the full line represents the actual robot path based on on-line path replanning. The parameters are the same as for the original virtual vehicle approach, i.e. $v_n = 0.5m/sec$, $d_\rho = 1m$, $k = 1$, $v_{max} = 0.5m/sec$, $\omega_{max} = 1.75rad/sec$, and the new factor $\zeta = 1$. As can be seen from curvature effort in Fig. 10, the four regions that are denoted in Fig. 3 correspond well to the four principal turns the robot has to make to achieve the global goal. In reference to the case of original virtual vehicle approach in Fig. 2 the region 3 was the critical one that caused the robot failure due to excessive reference point and robot velocity. In contrast, when comparing the same critical region in Fig. 5 where the reference point was moving at approximately nominal speed, in Fig. 8 one can observe that the reference point slowed down to $s_v \approx 0.2m/sec$ causing the robot to slow down also and approach the reference point at $\rho \approx 0.4m$, thus allowing it to perform the necessary turning maneuvers safely. This is a natural behavior analogy with movement of agents in a queue where in presence of rapid scene change the moving agents are slowed down for safety reasons and are also brought closer together to maintain sensorial information link. The robot was travelling at nominal speed only when there was no curvature change, which is the straight line region in the start of the run.

IV. On-line path replanning

In the curvature effort virtual vehicle approach it is assumed that the robot is on the start of the local curve segment (or in its immediate vicinity) since only then the curvature effort described in (9) is well defined. The criterion could be applied also to a completely static, off-line planned path where the robot may start at some point off the path if the closest point to the robot on the curve would be found, i.e. a point $\vec{p}_s$ on curve whose normal vector $\vec{N}_s$ points in the direction of the robot at the minimum distance. Such a point $\vec{p}_s$ can then be the virtual starting position of the robot on the path. However, the primal interest of the work aims at robot navigation in dynamic/partially unknown environments which inherently implies on-line path planning procedure for which the criterion (9) is well suited.

The implementation in this work involves a D* based on-line graph-search procedure as described in [15] on a grid-based global map of the environment. Thus obtained path based on connected grid cells is smoothed by a cubic B-spline curve where the cells represent the control points [12], [13]. However, regardless of the particular path search and smoothing algorithm, the successive curves may differ in form when the local path replanning is done at each servo tick, i.e. when new objects are encountered in sensor range and the path has to deviate from them. Since the same curve parameter value s_d reflects different positions along different curves it cannot be simply transferred to the next curve.

Therefore, the solution applied here is to perform a one step simulation of the reference point movement along the path for the current reference point position $s_d^{(i)}$ and currently calculated velocity $\dot{s}_d^{(i)}$ according to (10). The robot movement is also simulated according to basic kinematic equations. The new velocities $v^{(i+1)}$, $\omega^{(i+1)}$ that are necessary for the kinematic equation input are calculated based on the simulated control response to $v_{ref}^{(i)}$ and $\omega_{ref}^{(i)}$ for current $v^{(i)}$ and $\omega^{(i)}$ with robot acceleration limits $\dot{v}_{max}$ and $\dot{\omega}_{max}$. Thus, from simulated reference point position $\vec{p}_d\,^{(i+1)}$ and robot position $\vec{p}_r\,^{(i+1)}$, the new tracking distance $\rho^{(i+1)} = \|\vec{p}_d\,^{(i+1)} - \vec{p}_r\,^{(i+1)}\|$ can be calculated. When the new curve $\vec{c}(s)^{(i+1)}$ is replanned, the reference point is placed at the position $\vec{p}_c\,^{(i+1)}$ on the curve that is an Euclidean distance $\rho^{(i+1)}$ away from the starting position of the curve where the robot is placed. Note that $\vec{p}_d\,^{(i+1)}$ does not necessarily correspond to $\vec{p}_c\,^{(i+1)}$ if the curve form is changed. Under reasonable assumption that the curve form does not change significantly between two successive servo ticks, the reference point movement is stable along different curves. The only case where the reference point movement would be unstable is the case of continuous drastic curve change between two steps which would imply that the global planner has entered a limit cycle between two equally desirable paths. This is a theoretical situation that does not occur in practice if the robot is on continuous move. Fig. 13 represents a robot run in the presence of unknown obstacles (1-4) where the path was globally replanned at a $100ms$ servo cycle.

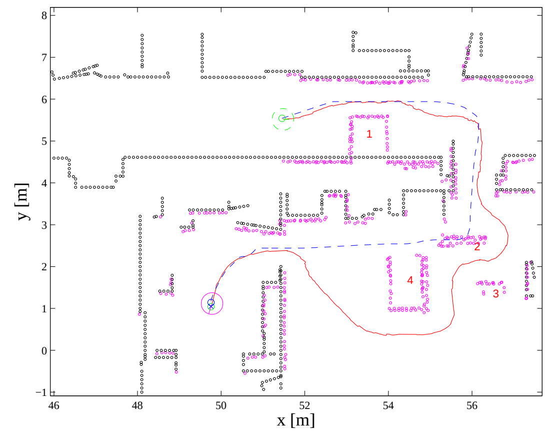

Fig. 13. A robot run in presence of unknown obstacles (1-4)

V. Experimental results

Experiments with the presented path planning/following scheme were carried out at the Faculty of Electrical Engineering and Computing in Zagreb with a Pioneer2DX differential drive mobile robot equipped with a Sick laser sensor. The office like environment was statically mapped for localization purposes. The initial off-line planned path is updated according to the unknown obstacles at run time as can be seen in Fig. 14 (at the time of the experiment the office was cluttered with obstacles due to reconstruction works). Due to the heavy load on the platform (max. support 23kg), the maximum translational speed was set to $v = 0.5m/sec$ (depending on the capacity of the wheel actuators limiting the accelerations) with the average speed of $v = 0.35m/sec$ attained (Fig. 15). Robot responses are also given in Fig. 16, 17, 18, 19 and 20. Results in terms of time to achieve goal position were comparable to the commonly used Gradient navigation method [16].

VI. Conclusion

The paper presents a path following method for mobile robots that provides lateral and longitudinal kinematic level control. The virtual vehicle approach that describes the dynamics of the reference point along a desired path is modified here according to the local curve characteristics called curvature effort. The local path configuration and the dynamic robot limitations are thus indirectly taken into account by introducing a penalizing factor to the speed of the reference point in the presence of obstacles enabling a safe robot operation. By a path replanning scheme at the servo tick level both obstacle avoidance and global path attendance are achieved. Future work would include finding explicitly an optimal choice of the penalty factor parameter according to robot dynamic constraints.

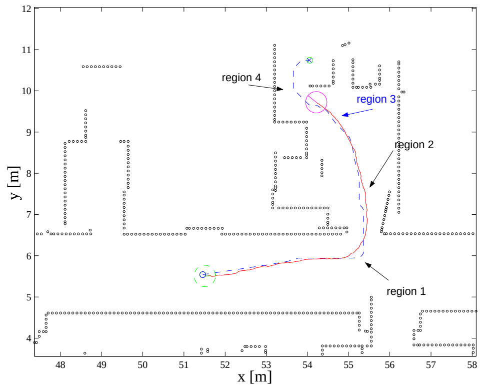

Fig. 2. Virtual vehicle robot run failure (simulation) - VV

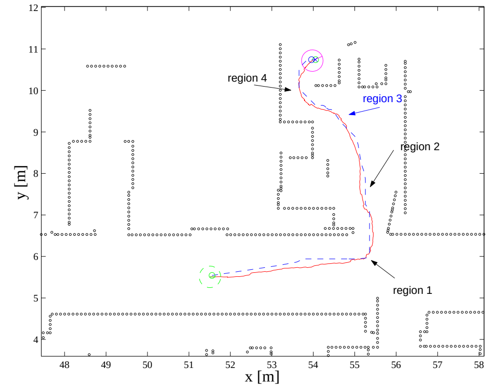

Fig. 3. Curvature effort based virtual vehicle robot run (simulation) - CE

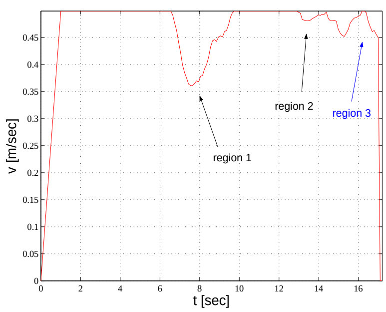

Fig. 4. Translational robot velocity (VV)

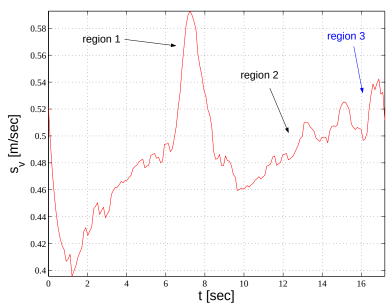

Fig. 5. Reference point velocity (VV)

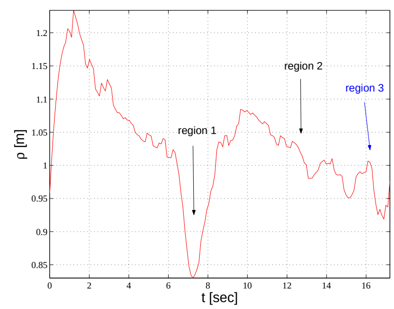

Fig. 6. Robot to reference point distance (VV)

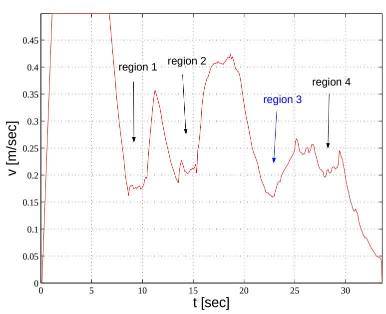

Fig. 7. Translational robot velocity (CE)

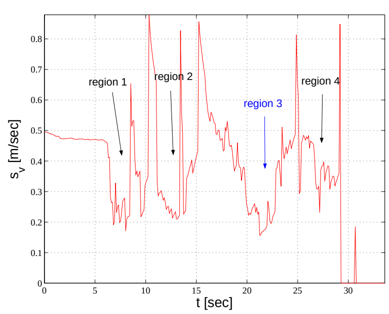

Fig. 8. Reference point velocity (CE)

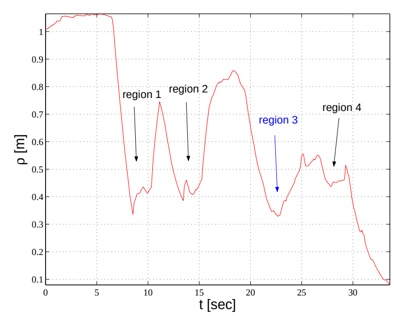

Fig. 9. Robot to reference point distance (CE)

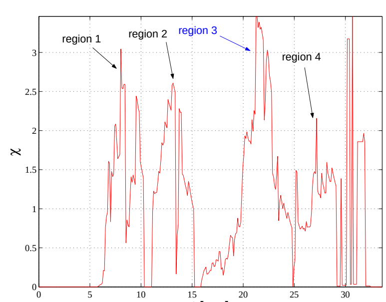

Fig. 10. Curvature effort measure (CE)

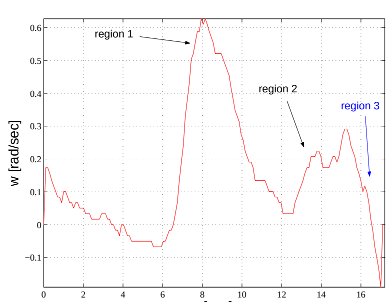

Fig. 11. Rotational robot velocity (VV)

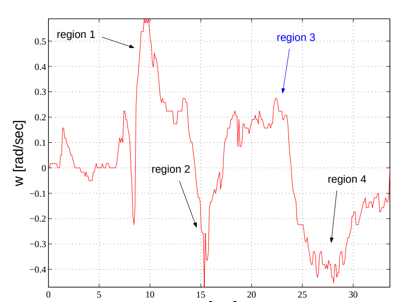

Fig. 12. Rotational robot velocity (CE)

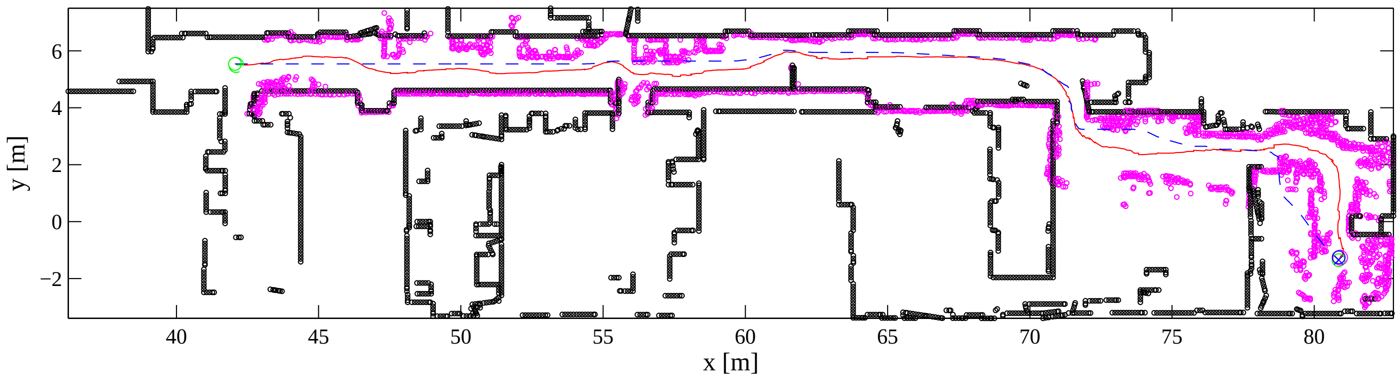

Fig. 14. Curvature effort based virtual vehicle robot run in an office environment (experiment)

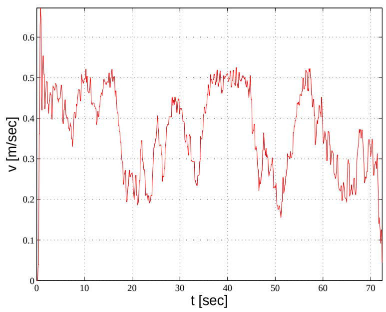

Fig. 15. Translational robot velocity

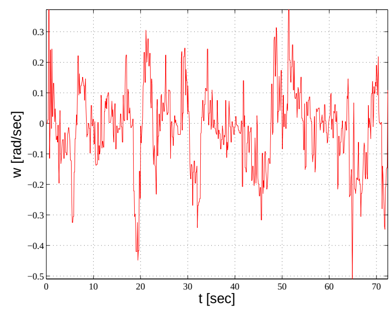

Fig. 16. Rotational robot velocity

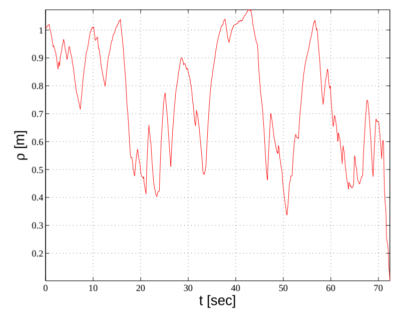

Fig. 17. Robot to reference point distance

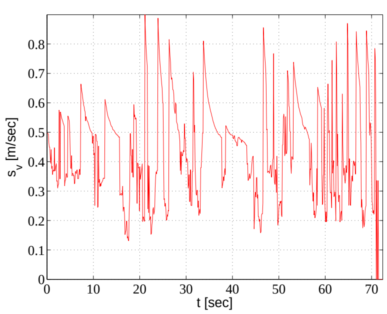

Fig. 18. Reference point velocity

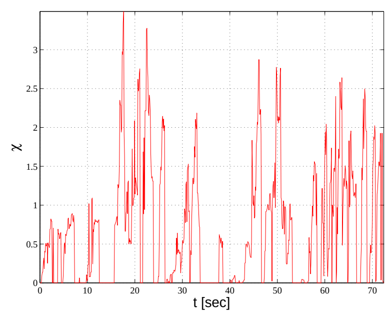

Fig. 19. Curvature effort measure

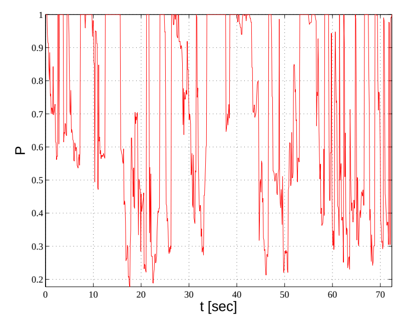

Fig. 20. Penalty factor

REFERENCES

[1] C. Canudas de Wit, B. Siciliano, G. Bastin,"Theory of robot control", New York, Springer Verlag, 1994.

[2] C. Canudas de Wit, A.D. Likodo, A. Michaelli, "Feedback control of a train-like vehicle", *IEEE International Conference on Robotics and Automation*, 1994.

[3] M. Fliess, J. Levine, P. Martin, P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples", *International Journal of Control*, Vol. 61(6), pp. 1327-1361, 1995.

[4] I. Kolmanovsky, N.H. McClamroch, "Developments in nonholonomic control problems", *IEEE Control Systems Magazine*, Vol. 15(6), 1995.

[5] R. Murray, S. Sastry, "Nonholonomic motion planning: Steering using sinusoids", *IEEE Transactions on Automatic Control*, Vol. 38(2), pp. 700-716, 1993.

[6] A. Balluchi, A. Bicchi, A. Balestrino, G. Casalino "Path tracking control for Dubins's car", *IEEE International Conference on Robotics and Automation*, 1996.

[7] H.G. Tanner, K.J. Kyriakopoulos, "Discontinuous backstepping for stabilization of nonholonomic mobile robots", *IEEE International Conference on Robotics and Automation*, 2002.

[8] A.P. Aguiar, J.P. Hespanha, P.V. Kokotović "Path-following for nonminimum phase systems removes performance limitations", *IEEE Transactions on Automatic Control*, Vol. 50(2), pp. 234-239, 2005.

[9] P. Coehlo, U. Nunes, "Path following of mobile robots in presence of uncertainties", *IEEE Transactions on Robotics*, Vol. 21(2), 2005.

[10] M. Egerstedt, X. Hu, A. Stotsky, "Control of mobile platforms using a virtual vehicle approach", *IEEE Transactions on Automatic Control*, Vol. 46(11), pp. 1777-1782, 2001.

[11] K. Maček, "Motion planning of mobile robots in indoor environments", MSc Thesis, University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia, 2004.

[12] R.H. Bartels, J.C. Beatty, B.A. Barsky, "An introduction to splines for use in computer graphics and geometric modeling", Morgan Kaufmann Publishers, Inc., 1986.

[13] S. Quinlan, "Real-time modification of collision-free paths", PhD Thesis, Stanford University, 1995.

[14] D.W. Henderson, "Differential geometry", Prentice Hall, Inc., 1998.

[15] A. Stenz, "The Focussed D* Algorithm for Real-Time Replanning", *International Joint Conference on Artificial Intelligence*, 1995.

[16] K. Konolige, "A gradient method for real-time robot control", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.