

Incremental Multi-Objective Motion Control of Nonholonomic Mobile Robots

Michail G. Lagoudakis

Intelligent Systems Laboratory

Department of Electronic and Computer Engineering

Technical University of Crete

Kounoupidiana, Chania, Crete 73100, Hellas (Greece)

lagoudakis@intelligence.tuc.gr

Abstract—This paper describes a motion controller which issues incremental velocity commands to a nonholonomic mobile robot guided by a holonomic path planner. The controller finds the best pair of rotational and translational velocities through discrete optimization of a multi-criteria objective function. Valid velocity choices are restricted within the *dynamic velocity window* as determined by the current velocities and the maximum acceleration and deceleration ability of the robot (smoothness criterion). In addition, for each choice the objective function examines the proximity of the resulting stopping configuration to the current goal set by the planner (accuracy criterion). Finally, it takes into account the potential of unforeseen, perhaps unavoidable, collisions due to dynamic obstacles or due to the holonomic nature of the path planner (safety criterion). The controller issues commands which balance these criteria as determined by the user. The proposed controller has been implemented and tested on a Nomad 200 mobile robot achieving smooth, accurate, and safe motion control both in simulated and in real environments.¹

I. INTRODUCTION

Mobile robot motion includes *path planning* at the higher level and *motion control* at the lower level. Path planning is the problem of finding *where* to go next (the desired position), whereas motion control is the problem of finding *how* to get there (the required control input). This convenient functional decomposition allows a path planner to deal with obstacles in a reduced discretized space in order to provide obstacle-free paths to the motion controller which, in turn, controls the trajectory of the robot in the continuous space typically without taking obstacles into account. This separation allows for using different techniques in attacking each problem independently provided a standard interface between them.

Our work adopts this decomposition and addresses mainly the problem of motion control in nonholonomic mobile robots, where motion is typically constrained by steering. This class of robots includes most wheeled mobile robots from small laboratory robots to larger autonomous automobiles. It is assumed that a path planner provides the direction and the distance of the desired position at each time step and the motion controller is responsible for deriving appropriate controls for reaching and stopping at that position. The robots are assumed to be controllable at the velocity level, that is, the control

input consists of a translational and a rotational velocity or speed. Finally, we assume that the robots are equipped with range sensors (laser, sonar, or infrared) which provide distance information to surrounding obstacles.

The motion control problem in nonholonomic robots has been studied extensively [1] [2]. Related research has focused on one of two main subproblems: (a) control inputs which respect the dynamic constraints of the robot (smoothness of motion), and (b) control inputs which minimize potential impact with obstacles (safety of motion). The first problem arises from the fact that robots can only accelerate or decelerate at a finite rate, whereas the second problem is specific to the nonholonomic nature of motion, as the actual trajectory chosen by the motion controller may not coincide with the straight-line trajectory of the path planner. Our work combines both lines of research and addresses both smoothness and safety within motion control at once in addition to accuracy.

Previous work along these lines [3] [4] [5] [6] has focused on integrating some of these objectives, however with somewhat different goals (achieving non-stopping high-speed motion or avoiding collisions irrespectively of dynamic constraints). Another work [7] along similar lines focused on integrating these two subproblems, however under a different objective that dealt with the elimination of local-minima. Similarly, another related work [8] focused on model-based integration of map features into collision avoidance.

In our approach, we have decoupled path planning and motion control functionally, but not temporally. The two processes run in an alternate fashion. At each step of the control cycle, the neural map path planner [9] provides the direction and the distance of the desired position (current subgoal) and the motion controller selects appropriate speeds for the robot in order to reach or at least approach that position. This cycle repeats continuously with the path planner providing updated positions and the motion controller updated velocities. In a sense, the motion controller is “chasing” the position set forth by the path planner with the objective of getting *and* stopping there smoothly, accurately, and safely. Note that we use the term position and not configuration for the goal since we are interested only in reaching some particular goal position on the 2-dimensional plane without any requirement on the orientation of the robot at that position.

¹This research was conducted while the author was with the Center for Advanced Computer Studies at the University of Louisiana, Lafayette, LA 70504, USA between August 1996 and May 1998.

II. MOTION CONTROL OF NONHOLONOMIC ROBOTS

The state of a robot at any given time is described uniquely by its *configuration*, a vector of generalized coordinates, typically one coordinate for each degree of freedom. The coupling between control inputs and robot actuators is described by the *kinematic model*, a set of state equations over the configuration of the robot. Denoting the configuration vector by $s \in R^n$ and the control input vector by $u \in R^{n-p}$, the kinematic equations are $\dot{s} = G(s)u$, where $G(s)$ is an $n \times (n-p)$ matrix. For $p = 0$ and diagonal $G(s)$, the robot is *holonomic* and each degree of freedom can be controlled independently by a single control input. On the other hand, for $p > 0$ there are less control inputs (degrees of action) than degrees of freedom. Such robots are known as *nonholonomic* or *underactuated*, because some of the generalized coordinates are coupled and cannot be controlled independently. A nonholonomic robot is subject to p nonholonomic constraints of the form $H(s)\dot{s} = 0$. Such constraints are nonintegrable and, therefore, the dimension of the configuration space cannot be reduced.

The *motion control* problem is to find an appropriate control input $u(t)$ to drive the robot from some initial configuration $s(0)$ to a desired configuration $s(t_f)$ in finite time t_f or alternatively to force the robot to track a desired trajectory $s_d(t)$ over time. Notice that for a nonholonomic robot the matrix $G(s)$ cannot be inverted to provide the control input. A straightforward projection strategy used in such cases is the pseudoinversion of $G(s)$ [10], which gives the control input $u(t) = (G^\top(s)G(s))^{-1}G^\top(s)\dot{s}_d(t)$, where $\dot{s}_d(t)$ is the time derivative of the desired trajectory.

The kinematic model of the class of mobile robots we consider is the *unicycle*, that is, a single wheel moving on a 2-dimensional plane. There are three generalized coordinates in the configuration $s = (x, y, \theta)^\top$ of the unicycle: the (x, y) position of the unicycle and its orientation θ with respect to some coordinate frame (Figure 1). Given that the control vector consists of a translational speed u and a rotational speed v , the kinematic equations are defined as follows:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} u \\ v \end{pmatrix}$$

Clearly, this system is nonholonomic and the nonintegrable constraint is

$$(\sin\theta \quad -\cos\theta \quad 0) \times \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = 0 \iff \tan\theta = \frac{\dot{y}}{\dot{x}}$$

which simply states that any change in the position of the unicycle occurs in the direction of its orientation. The pseudoinverse control law in this case is [10]:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\theta}_d \end{pmatrix},$$

where $\dot{s}_d = (\dot{x}_d, \dot{y}_d, \dot{\theta}_d)^\top$ is the desired trajectory. Although this control law minimizes the difference between desired and

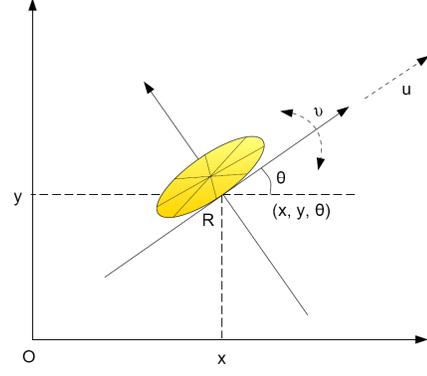


Fig. 1. The unicycle model.

actual trajectory in a least-squares sense, it does not take into account the dynamic constraints of the model, that is, the maximum acceleration/deceleration $(\dot{u}_{\max}, \dot{v}_{\max})$. Moreover, there is no guarantee that the actual trajectory of the unicycle will not overlap with obstacles [10] [3], even if the desired trajectory is collision-free. This phenomenon can occur if the desired trajectory violates the nonholonomic constraint, in which case the control law above will only approximate the desired trajectory moving the robot in areas which might be occupied by obstacles.

Derivation of the unicycle trajectory can be complicated for variable control input, however, it is fairly straightforward for constant controls. In particular, we are interested in deriving the trajectory equations in a local spatial and temporal sense for the time interval between control commands during which speeds are held constant (see also [3]). Consider Figure 2 which shows the local coordinate frame attached to the robot and aligned to its orientation (the robot is located at the origin R and faces in the direction of the axis Rx). Assume that for $0 \leq t \leq \Delta t$ a constant control input $(u_0, v_0)^\top$ is applied, where t offers a local view of time between control intervals. Integrating the three kinematic equations, we obtain the trajectory equations with initial conditions $(x(0), y(0), \theta(0))^\top = (0, 0, 0)^\top$:

$$\theta(t) = \int_0^t \dot{\theta} d\tau = \int_0^t v_0 d\tau = v_0 t$$

$$x(t) = \int_0^t \dot{x} d\tau = \int_0^t \cos\theta(\tau) u_0 d\tau = u_0 \int_0^t \cos(v_0 \tau) d\tau = \frac{u_0}{v_0} \sin(v_0 t)$$

$$y(t) = \int_0^t \dot{y} d\tau = \int_0^t \sin\theta(\tau) u_0 d\tau = u_0 \int_0^t \sin(v_0 \tau) d\tau = \frac{u_0}{v_0} (1 - \cos(v_0 t))$$

It is easy to see that this is a trajectory along the periphery of a circle centered at $(0, \frac{u_0}{v_0})$ with radius $\rho = \left| \frac{u_0}{v_0} \right|$. We can further derive the distance r from the origin as a function of time t and its angle:

$$r(t) = \sqrt{x^2(t) + y^2(t)} = 2\rho \left| \sin\left(\frac{v_0 t}{2}\right) \right|$$

$$\psi(t) = \arctan\left(\frac{y(t)}{x(t)}\right) = \frac{v_0 t}{2}$$

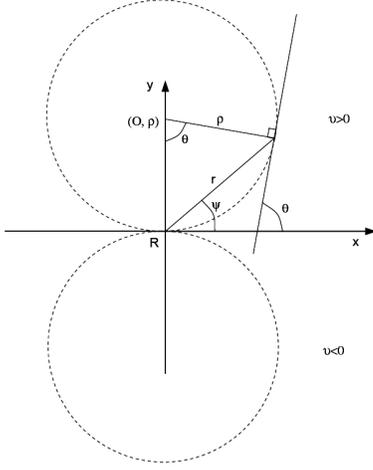


Fig. 2. The unicycle's trajectory under constant control input.

We assumed above that the rotational speed v of the robot is non-zero. For $v = 0$, the robot will follow a straight line along the Rx axis, which can be viewed as a circular trajectory on the periphery of a circle with infinite radius.

III. MULTI-OBJECTIVE MOTION CONTROL

To address the motion control problem effectively we propose a motion controller based on discrete optimization of a multi-criteria objective function which balances smoothness, accuracy, and safety.

A. Smoothness: The Dynamic Window

The dynamics of the robot impose constraints on the control input. Such constraints include upper limits on the velocities and the corresponding accelerations/decelerations. Let u_{\max} be the absolute value of the maximum translational velocity and v_{\max} be the absolute value of the maximum rotational velocity. Also, let \dot{u}_{\max} be the absolute value of the maximum translational acceleration and \dot{v}_{\max} the absolute value of the maximum rotational acceleration. If the robot is moving with speeds $(u_0, v_0)^\top$ at time t_0 , then, for the next time interval Δt , the Dynamic Window (DW) [5] [4] is the subset of allowable velocities in the velocity space as defined by the following inequalities:

$$0 \leq u \leq +u_{\max}$$

$$-v_{\max} \leq v \leq +v_{\max}$$

$$u_0 - \dot{u}_{\max} \times \Delta t \leq u \leq u_0 + \dot{u}_{\max} \times \Delta t$$

$$v_0 - \dot{v}_{\max} \times \Delta t \leq v \leq v_0 + \dot{v}_{\max} \times \Delta t$$

Notice that we only consider forward motion ($u \geq 0$) of the robot². Any pair (u, v) within the current dynamic window $DW(u_0, v_0)$ respects the dynamics of the robot.

²To allow for backward motion of the robot the first inequality simply becomes $-u_{\max} \leq u \leq +u_{\max}$.

B. Accuracy: Trajectory Estimation

Each pair of speeds must be assessed for its appropriateness as the next control command with respect to the desired goal position. Figure 3 illustrates the situation. The coordinate frame is the egocentric reference frame of the robot. Initially, the robot is located at the origin R facing in the direction of the axis Rx . The desired position, specified by the path planner, is $s_d = (\Delta r_d, \Delta \phi_d)$.

Assume that at some time t_0 the robot is moving with speeds $(u_0, v_0)^\top$. Let the *Minimum Braking Distance* (MBD) be the least distance the robot will travel between time t_0 and the time it comes to a complete stop, assuming that it is decelerating with maximum translational deceleration. Similarly, let the *Minimum Braking Angle* (MBA) be the least rotation angle of the robot between time t_0 and the time it comes to a complete stop, assuming that it is decelerating with maximum rotational deceleration. Formally,

$$MBD(u_0) = \frac{1}{2} \times u_0 \times \frac{u_0}{\dot{u}_{\max}} \quad MBA(v_0) = \frac{1}{2} \times v_0 \times \frac{v_0}{\dot{v}_{\max}}$$

Let $(u', v')^\top$ be a pair of speeds selected from the current dynamic window. Assume that the pair $(u', v')^\top$ is applied as the next control command for time Δt and immediately after the robot starts decelerating with maximum deceleration. During the (short) control interval Δt the robot moves with constant speeds $(u', v')^\top$, reaching a configuration $s(\Delta t) = (x(\Delta t), y(\Delta t), \theta(\Delta t))^\top$ which can be determined using the trajectory equations of the unicycle for the current speeds $(u', v')^\top$. Assuming maximum deceleration, computation of the final configuration $F = (F_x, F_y, F_\theta)^\top$ of the robot (when it comes to a complete stop) becomes a non-trivial problem. However, F can be approximated by a straight-line deceleration of length $MBD(u')$ rotated by $MBA(v')/2$ with respect to the orientation right before deceleration:

$$F_x = x(\Delta t) + MBD(u') \times \cos\left(\theta(\Delta t) + \frac{MBA(v')}{2}\right)$$

$$F_y = y(\Delta t) + MBD(u') \times \sin\left(\theta(\Delta t) + \frac{MBA(v')}{2}\right)$$

$$F_\theta = \theta(\Delta t) + MBA(v')$$

Note that the loss due to the approximation is not crucial as the robot enters the next control loop at position $s(\Delta t)$ before even reaching position F . Given the estimated configuration of the robot at position F , our goal is to minimize the distance between the points s_d and F , as well as the angle marked by $\hat{\theta}$ in Figure 3.

C. Safety: Impact Minimization

The presence of obstacles along the expected trajectory of the robot for a given choice of speeds may impose additional constraints within the dynamic window. We abstain, however, from a constraint satisfaction formulation, which may provide no solution at all if all pairs of speeds violate the imposed constraints (for example, due to an unavoidable collision).

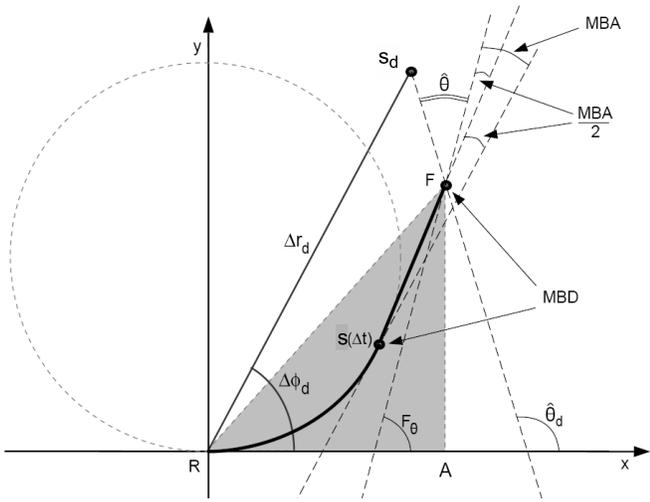


Fig. 3. Estimation of trajectory and final position.

Even in the case of an unavoidable collision due to the inability to stop instantly, a pair of velocities must still be selected, so as to minimize the impact of the collision.

The need for taking obstacles into account stems from the decomposition into holonomic path planning and nonholonomic motion control. Consider Figure 3. The path planner returns the desired configuration $s_d = (\Delta r_d, \Delta \phi_d)$, guaranteeing by definition that the straight line $R \rightarrow s_d$ is free of obstacles. However, due to the nonholonomic constraint the robot in practice follows a curved trajectory ($R \rightarrow s(\Delta t) \rightarrow F$) which may overlap with obstacles. A naive solution in some cases would be to align the orientation of the robot with the $R \rightarrow s_d$ line before translation, in essence disassociating rotational and translational control, however such a scheme would result in unnatural and slow robot motion.

To assess the possibility of a collision for the pair of speeds (u, v) we use a measure $WDO(u, v)$ of weighted density of obstacles in the (shaded) triangle RFA , where obstacles close to the origin R (current position) have higher weight than obstacles away from the origin. If there are no obstacles in RFA , $WDO(u, v)$ is zero. The lower the value of $WDO(u, v)$, the lower the possibility for a collision and the lower the impact of an unavoidable collision. Information about obstacles is provided directly by the range sensors of the robot (centered at R).

D. The Objective Function

The motion control problem is now reduced to finding the best pair of speeds within the dynamic window given the desired goal position and the current translational and rotational speeds. The best choice is defined with respect to an additive multi-objective function:

$$\mathcal{F}(u, v) = w_1 f_1(u, v) + w_2 f_2(u, v) + w_3 f_3(u, v)$$

Each f_i optimizes (minimizes) some particular measure: the Euclidean distance from the desired position, the orientation

toward the desired position, and the impact of a potential collision. We consider the squared error in each case:

$$f_1(u, v) = (F_x - \Delta r_d \times \cos(\Delta \phi_d))^2 + (F_y - \Delta r_d \times \sin(\Delta \phi_d))^2$$

$$f_2(u, v) = (F_\theta - \hat{\theta}_d)^2$$

$$f_3(u, v) = (WDO(u, v))^2$$

where $\hat{\theta}_d$ is a desired angle at F , so that the robot is facing towards the desired position³:

$$\hat{\theta}_d = \arctan2(\Delta r_d \times \sin(\Delta \phi_d) - F_y, \Delta r_d \times \cos(\Delta \phi_d) - F_x)$$

The weights/gains w_i (determined empirically) provide the means for scaling the different ranges and weighing the importance of each sub-objective.

The resulting objective function can be computed at any point (u, v) , however it cannot be minimized analytically in general. A simple solution is to discretize the dynamic window and probe the value of the function at those discrete points with the purpose of finding the minimum. Given that the function is defined over a low-, two-dimensional space, the combinatorial effects are not prohibitive and even a fine grid can be probed efficiently. There are some pruning techniques one can apply to speed up the process. For example, if a set of speeds (u, v) with small v leads to overshooting, any pair with a higher value of u can only lead to further overshooting and needs no probing. However, rarely do these pruning methods yield significant savings, while their overhead is quite significant and increases the control cycle time. In our case, we found that simply probing all points on the grid results in the shortest cycle time.

In summary, each control loop consists of the following operations:

- 1) determine the current dynamic window
- 2) discretize the dynamic window as a 2-dimensional grid
- 3) evaluate the function at every pair of speeds in the grid
- 4) select the pair that minimizes the objective function
- 5) use the selected pair of speeds as the next control input

IV. EXPERIMENTAL RESULTS

The proposed motion controller has been tested on Boudreaux (shown in Figure 4), a Nomad 200™ mobile robot, manufactured by Nomadic Technologies Inc. The mobile base of the robot has a three-wheel synchronous drive mechanism which allows for nonholonomic motion. Two independent motors control the motion; one steers and the other drives the wheels. A zero gyro-radius enables the robot to turn in place. The maximum achievable speeds are 24 inches/sec for translation and 60 degrees/sec for rotation. The robot is equipped with a Sensus 200™ Sonar System. This is a ring of 16 ultrasonic range finders (sonars) distributed uniformly on the periphery of the robot, providing 360° coverage with 22.5° resolution. Each sensor can measure depth from 6 inches up to 255 inches.

³ $\arctan2(y, x)$ calculates $\arctan(y/x)$ and returns an angle in the correct quadrant.

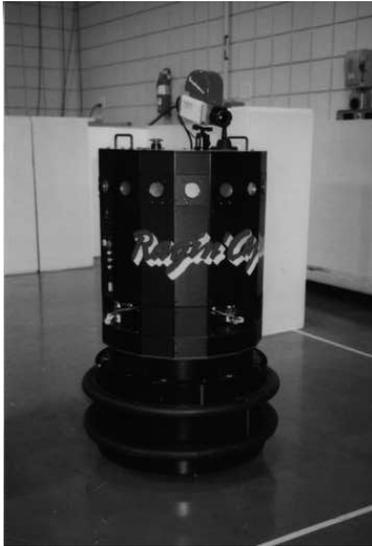


Fig. 4. Boudreaux, our Nomad 200 mobile robot.

Path planning is provided by a local path planner [9] which uses a polar neural map and sonar range data to provide desired positions to the motion controller in egocentric coordinates. The time interval for each control cycle is measured dynamically as it depends on several run-time factors (computer architecture, processor load, communication delays). The average value of the Δt in our experiments was 0.25 seconds which implies that control took place at a frequency of 4 Hz. Our navigation system is also enhanced by appropriate transformation of recent sensory information within a short time window and by forward prediction of the current robot configuration to the configuration at the beginning of the next control cycle [11]. The complete architecture of our system is shown in Figure 5.

In our experiments we used a 50×50 discrete grid for optimizing the objective function. The exact resolution of the grid is rather a choice of the designer and need not be uniform along the two dimensions. Higher resolution will result in better choices at the cost of longer control cycles, whereas lower resolution might not yield the best values, but will allow for more frequent choices. Distances on the 2-dimensional plane were measured in tenths of inches and angles in radians. The values of the weights/gains w_i strongly depend on the particular implementation. In our experiments we used $w_1 = 1$, $w_2 = 50$, and $w_3 = 20000$ which were found empirically by trial-and-error. These values stress the importance of the safety sub-objective, whereas the other two sub-objectives are scaled to comparable importance.

A head-to-head experimental comparison of our motion controller with similar controllers [3] [4] [5] [6] is not directly possible. The controllers closer to ours [4] [5] [6] focus on non-stop high-speed motion and do not explicitly consider the case of moving to *and* stopping at a certain goal position. In our approach, the high-speed criterion is automatically taken into account through the accuracy criterion and needs no

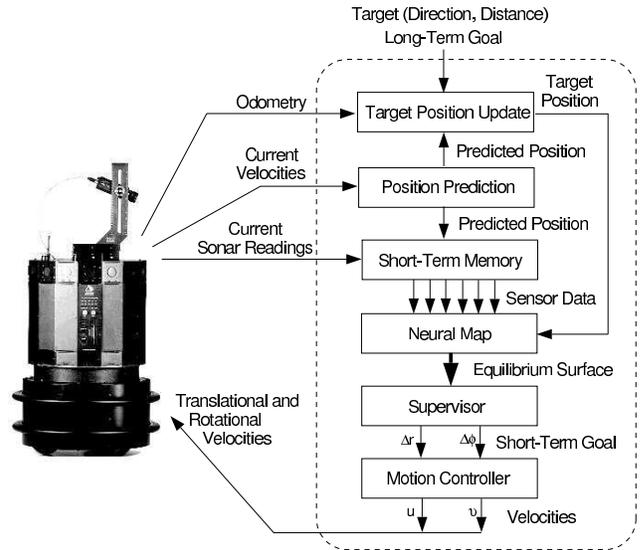


Fig. 5. The complete architecture of our navigation system.

explicit optimization term. The remaining controller [3] takes potential collisions into account, however it does not respect the dynamics of the robot and therefore the controller output could not be possibly executed on the real robot. In all cases, a comparison would be unfair and perhaps meaningless.

We did, however, compare our motion controller against a heuristic controller of a type used in many navigation systems. Such a controller basically issues velocity commands proportional to the magnitude of Δr and $\Delta \phi$ along with some basic safety guards for avoiding unforeseen collisions and some constraints for respecting the dynamics of the robot. Figure 6 shows a sample run of the robot between the same points in a cluttered environment using both controllers. Figure 6 also shows the velocity history over time (both translational and rotational) for both controllers. The translational velocity is shown in inches/second and the rotational in degrees/second. The time axis shows control steps. Although the robot trajec-

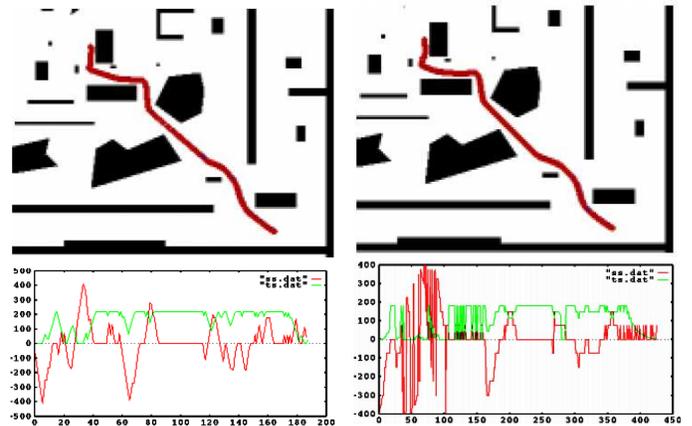


Fig. 6. Comparison of the proposed motion controller (left) to a typical heuristic controller (right).

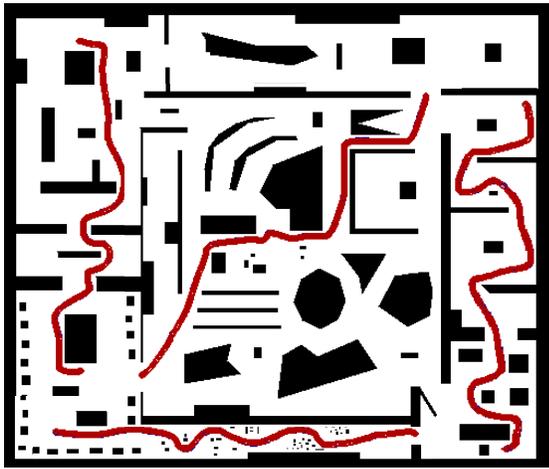


Fig. 7. Robot paths for various navigation problems.

tory looks smooth enough over the 2-dimensional plane in both cases, the performance difference is pronounced over the time axis. Under the heuristic controller, the robot came to sudden stops and jerky starts several times taking a total of more than 400 steps to complete the path. Notice the oscillations present in the velocity profile. On the other hand, under the proposed motion controller the robot jerk was decreased significantly. There were no significant oscillations and the few ones present were due to noisy sonar data. It is remarkable that the path was completed in less than 200 steps in this case, as the robot was moving at maximum speed except during turns.

Figure 7 shows a set of four different runs in a large environment (approximately 45×40 meters) and Figure 8 shows the corresponding velocity histories (sampled every few time steps). In all test cases, the motion of the robot is safe and smooth and maintains high translational speed which is lowered significantly only when taking sharp turns corresponding to high rotational speed. At the end of each run the robot slowly decelerates and stops at the desired position. Notice the variety of obstacle shapes and sizes, as well as potential route choices. Where possible, narrow and dangerous passages which could slow down the robot are avoided.

V. CONCLUSION

This research combines the best elements of previous research on motion control of nonholonomic mobile robots. The proposed scheme allows for a complete functional decomposition of path planning and motion control by enhancing the lower level of motion control with the ability to balance several objectives including smoothness, accuracy, and safety. It differs from similar approaches in its purpose (move to and stop at a goal position) and its methods (criteria which automatically encourage high-speed navigation). Our motion controller can be used on any nonholonomic mobile robot and especially on autonomous automobiles where motion is highly nonholonomic and takes place at relatively high speeds.

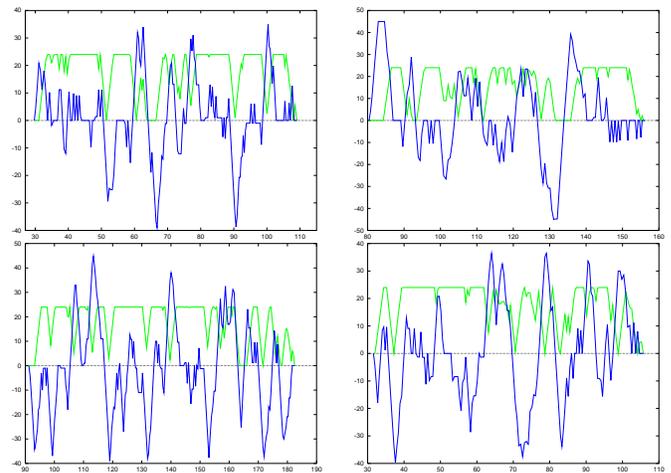


Fig. 8. Translational and rotational velocities for each path.

ACKNOWLEDGMENT

The author would like to thank Anthony S. Maida and Kimon P. Valavanis for their help and support, and the Lilian Boudouri Foundation in Greece for partial financial support.

REFERENCES

- [1] C. C. de Wit, H. Khenouf, C. Samson, and O. Sordalen, *Recent Trends in Mobile Robots*. River Edge, New Jersey: World Scientific Publishers, 1993, ch. Nonlinear Control Design for Mobile Robots, pp. 121–156.
- [2] G. Lafferriere and H. Sussmann, “Motion planning for controllable systems without drift,” in *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, T. Tarn, Ed. Sacramento, California: IEEE Press, 1991, pp. 1148–1153.
- [3] S.-G. Hong, S.-W. Kim, K.-B. Park, and J.-J. Lee, “Local motion planner for nonholonomic mobile robots in the presence of the unknown obstacles,” in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, G. Lee, Ed. Minneapolis, Minnesota: IEEE Press, 1996, pp. 1212–1217.
- [4] R. Simmons, “The curvature-velocity method for local obstacle avoidance,” in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, G. Lee, Ed. Minneapolis, Minnesota: IEEE Press, 1996, pp. 3375–3382.
- [5] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics and Automation Magazine*, vol. RA-4, pp. 23–33, March 1997.
- [6] D. Castro, U. Nunes, and A. Ruano, “Reactive local navigation,” in *Proceedings of the IEEE 2002 28th Annual Conference of the Industrial Electronics Society*. IEEE Press, 2002, pp. 2427–2432.
- [7] O. Brock and O. Khatib, “High-speed navigation using the global dynamic window approach,” in *Proceeding of the 1999 IEEE International Conference on Robotics and Automation*, 1999, pp. 341–346.
- [8] D. Fox, W. Burgard, S. Thrun, and A. Cremers, “A hybrid collision avoidance method for mobile robots,” in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*. IEEE Press, 1998, pp. 1238–1243.
- [9] M. G. Lagoudakis and A. S. Maida, “Neural maps for mobile robot navigation,” in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN’99)*, vol. III. Washington, DC: IEEE Press, July 1999, pp. 2011–2016.
- [10] A. D. Luca and G. Oriolo, “Local incremental planning for non-holonomic mobile robots,” in *Proceedings of the 1994 International Conference on Robotics and Automation*, E. Straub and R. S. Sipple, Eds. Los Alamitos, California: IEEE Computer Society Press, May 1994, pp. 104–110.
- [11] M. G. Lagoudakis, “On improving mobile robot motion control,” in *Proceedings of the 4th Hellenic Conference on Artificial Intelligence (SETN’06)*, ser. Lecture Notes in Artificial Intelligence (LNAI). Herakleion, Crete, Hellas: Springer, May 2006, pp. 551–554.