High-Speed Navigation Using the Global Dynamic Window Approach

Oliver Brock Oussama Khatib
Robotics Laboratory, Department of Computer Science
Stanford University, Stanford, California 94305
email: {oli, khatib}@CS.Stanford.EDU

Abstract

Many applications in mobile robotics require the safe execution of a collision-free motion to a goal position. Planning approaches are well suited for achieving a goal position in known static environments, while real-time obstacle avoidance methods allow reactive motion behavior in dynamic and unknown environments. This paper proposes the global dynamic window approach as a generatlization of the dynamic window approach. It combines methods from motion planning and real-time obstacle avoidance to result in a framework that allows robust execution of high-velocity, goal-directed, reactive motion for a mobile robot in unknown and dynamic environments. The global dynamic window approach is applicable to non-holonomic and holonomic mobile robots.

1 Introduction

Algorithms that generate motion for mobile robots can be divided into planning algorithms and real-time obstacle avoidance algorithms. Planning algorithms consider a model or map of the environment to compute a path from the robot's current position to the goal, whereas obstacle avoidance algorithms usually use sensory information to determine a motion that will avoid collision with obstacles in the environment.

For most applications in mobile robotics the environment is partially or completely unknown and changes with time. Under such circumstances the trajectories generated by planning algorithms become inaccurate and replanning is required to reach the goal. Since planning can be too time-consuming to avoid collisions in real-time, motion commands for mobile robots are usually generated by computationally efficient real-time obstacle avoidance approaches. Purely reactive obstacle avoidance, however, may not result in the behavior required to accomplish the robot's task.

In this paper the global dynamic window approach is introduced. This framework combines planning and real-time obstacle avoidance algorithms to generate motions for mobile robots that achieve the robot's task, while securely navigating in an unknown and dynamic environment. The framework is applied to high-speed navigation in unknown environments using a holonomic mobile base.

2 Related Work

2.1 Real-Time Obstacle Avoidance

Most of the earlier real-time obstacle avoidance approaches were based on artificial potential fields [10]. The robot is kept at a safe distance from obstacles by a repulsive force, while being drawn towards the goal by an attractive force. To refine the trajectories generated by this approach, various extensions have been suggested [9]. While artificial potential field approaches are computationally efficient, the robot can get stuck in local minima before reaching the goal position. This is due to the fact that no information about the connectivity of the free space is used to determine the motion.

In the vector field histogram approach [2] a direction of motion is chosen based on sensory information such that obstacles are avoided while the robot continues to move towards the goal. As with the potential field approach the robot can get trapped in local minima. Extending this approach, parameterized path families [5], or more specifically steer angle fields, take the nonholonomic kinematic constraints of the robot into account when choosing a motion. This reduces the search space and makes the approach more efficient.

The curvature-velocity method [14] and the dynamic window approach [6] are based on the steer angle field approach. In addition to kinematic constrains these frameworks take into account dynamic constraints to reduce the search space even further. Although these approaches yield very good results for obstacle avoidance at high velocities, the problem of local minima persists.

The dynamic window approach has been integrated with a gross motion planner [13] and was extended to use a map in conjunction with sensory information to generate collision free motion [7]. A Bayesian approach to obstacle avoidance was linked with global path planning [8]. However, these approaches require a priori knowledge about the environment for the execution of a motion command.

2.2 Motion Planning

There is a large number of robot motion planning algorithms presented in the literature [12]. In low-dimensional configuration spaces, like those for mobile robots, the use of a navigation function [11] seems to be an appealing approach to motion planning. A navigation function represents a virtually local minima-free[1] artificial potential function that can be used locally to guide the robot to the global goal. Constructing a grid-based navigation function results in very simple and computationally efficient motion planning algorithms [1].

Other navigation functions include the harmonic potential function [4] and circulatory fields [15]. Harmonic potential functions use fluid dynamics to compute a local minima-free potential function. In the circulatory field approach obstacles are surrounded by a magnetic field caused by a fictitious current flowing through their surface. The robot navigates around an obstacle by aligning itself with this field. These approaches require complete knowledge of the shape of obstacles, their location, and motion to construct a navigation function. This is an unreasonable assumption for many applications in mobile robotics.

In another approach the concepts of potential field-based obstacle avoidance and approximate cell decomposition motion planning were used in conjunction to yield a framework for planning and execution of robot motion [3]. This framework requires partial knowledge of the environment but will tolerate small, unforeseen obstacles and minor changes in the environment.

[1] Saddle points are ignored here.

3 Holonomic Dynamic Window Approach

Holonomic robots have several advantages over car-like and synchro-drive robots. Since they allow instantaneous acceleration in all directions they are much easier to control and have an increased maneuverability. The orientation of the robot can be controlled independently of its motion in the plane. In addition, the equations of motion have a simple closed-form solution. This provides the motivation for the generalization of the dynamic window approach to holonomic robots presented in this section. In Section 4 the holonomic dynamic window approach will be integrated with a global planning method to result in the global dynamic window approach.

3.1 Dynamic Window Approach

The dynamic window approach [6] is an obstacle avoidance method that takes into account the kinematic and dynamic constraints of a synchro-drive robot. Kinematic constraints are taken into account by directly searching the velocity space of a synchro-drive robot. The search space is the set of tuples (v, ω) of translational velocities v and rotational velocities ω that are achievable by the robot.

Among all velocity tuples those are selected that allow the robot to come to a stop before hitting an obstacle, given the current position, the current velocity, and the acceleration capabilities of the robot. These velocities are called the *admissible velocities*.

Restricting the search to a dynamic window further reduces the search space in accordance with dynamic limitations of the robot. The dynamic window contains those velocities that can be achieved by the robot, given its current velocity and its acceleration capabilities, within a given time interval. This time interval corresponds to a servo tick of the control loop. Figure 1 illustrates the subdivision of the search space in the dynamic window approach. The dynamic window is a rectangle, since acceleration capabilities for translation and steering are independent.

To determine the next motion command all admissible velocities within the dynamic window are considered. Among those a velocity is chosen that maximizes the alignment of the robot with the target and the length of the trajectory until an obstacle is reached.

Using this approach, robust obstacle avoidance behavior has been demonstrated at high velocities [6]. However, since the dynamic window approach only considers goal heading and no connectivity informa-

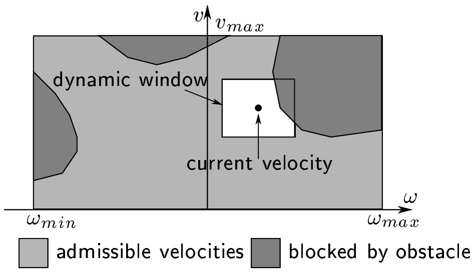


Figure 1: Search space for dynamic window approach

tion about the free space, it is still susceptible to local minima.

3.2 Search Space

The most important difference between the dynamic window approach and the holonomic dynamic window approach is the overall search space. A holonomic robot has no limitations on the direction of instantaneous acceleration. However, it is impractical to search the entire space of possible velocity changes. Therefore a subset has to be selected that exploits the kinematic advantages of holonomicity while retaining computational feasibility.

For the holonomic dynamic window approach the search space consists of all possible velocities in a global reference frame. It is discretized in polar coordinates, choosing a fixed set of directions and scalar velocities. This results in a circular search space and a circular dynamic window, as depicted in Figure 2.

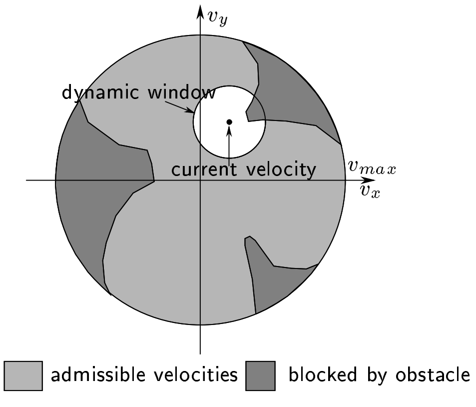


Figure 2: Search space for holonomic dynamic window approach

The use of a global reference frame allows the decoupling of the two translational axes, yielding the following equation of motion for the x-axis for constant acceleration a_x and velocity v_x:

$$x(t_i) = x(0) + v_x t_i + \int_0^{t_i} a_x t \, dt = x(0) + v_x t_i + \frac{1}{2} a_x t_i^2$$

and similarly for the y-axis. These equations show that when accelerating from a constant velocity to achieve a given velocity command the robot describes a quadratic curve until the desired velocity is attained. The curvature of those curves depends on the magnitude of the acceleration. In order to achieve curves with low curvature the two-dimensional search space shown in Figure 2 is searched for different accelerations. Low accelerations result in low curvature and allow to imitate car-like behavior. If the accelerations are chosen such that $a_{xy} = a_x = a_y$ the resulting overall search space is three-dimensional and a motion command is defined by $\vec{v} = (v_x, v_y)$ and $\vec{a} = (a_{xy}, a_{xy})$.

To determine if a motion command $(\vec{v}, \vec{a})$ is admissible the length of the resulting trajectory has to be determined. Simulation of the base motion according to the equations of motion will determine the duration t_i of the trajectory until hitting an obstacle. The length $l(\vec{v}, \vec{a}, t_i)$ of the trajectory can then be computed analytically:

$$\begin{aligned} l(\vec{v}, \vec{a}, t_i) &= \int_0^{t_i} v_{t_i} \, dt \\ &= \int_0^{t_i} \sqrt{(v_x + a_x t)^2 + (v_y + a_y t)^2} \, dt. \end{aligned}$$

If the length of the trajectory permits the robot to come to a halt after moving for the duration of one servo tick, the motion command is considered admissible.

3.3 Objective Function

A desired velocity $\vec{v} = (v_x, v_y)$ and an acceleration $\vec{a} = (a_x, a_y)$ are selected from the search space according to the objective function

$$\Omega(\vec{p}, \vec{v}, \vec{a}) = \alpha \cdot \mathit{align}(\vec{p}, \vec{v}) + \beta \cdot \mathit{vel}(\vec{v}) + \gamma \cdot \mathit{goal}(\vec{p}, \vec{v}, \vec{a}),$$

where $\vec{p} = (x, y)$ is the position vector of the mobile base. This objective function is a linear combination of three functions. The ranges of those functions are normalized to the interval $[0, 1]$.

To favor trajectories that are directed towards the goal, the function $\mathit{align}(\vec{p}, \vec{v}) = 1 - |\theta|/\pi$, where θ is the angle between the direction of motion and goal heading, results in large values for good alignment with the goal heading. The goal heading is modified if the robot's lateral distance to an obstacle becomes too small.

The function $vel(\vec{v})$ is defined as follows:

$$vel(\vec{v}) = \begin{cases} \frac{\|\vec{v}\|}{v_{max}} & \text{if robot is far from goal} \\ 1 - \frac{\|\vec{v}\|}{v_{max}} & \text{if robot is close to goal} \end{cases},$$

where v_{max} the maximum velocity the robot can achieve. It will favor high velocities if the robot is far from the goal and low velocities when it is close. If the trajectory that results from the motion command $(\vec{v}, \vec{a})$ passes through the goal region, the value of the binary function $goal(\vec{p}, \vec{v}, \vec{a})$ is 1 , otherwise it is 0. The parameters α, β, and γ can be adjusted to modify the behavior of the robot. The algorithm's performance is robust over a wide range of values.

4 Global Dynamic Window Approach

The dynamic window approach and the holonomic dynamic window approach are both susceptible to local minima. The robot's motion with respect to the goal is only influenced by the goal heading. This limitation can be removed by incorporating information about the connectivity of the free space into the selection of a motion command.

The global dynamic window approach presented in this section extends the dynamic window approach [6] and the holonomic dynamic window approach presented in Section 3 by incorporating a simple and efficient motion planning algorithm. The global planning is efficient enough to be executed for each servo tick of the motion controller. This allows robot navigation in real-time in a globally goal-directed fashion.

No prior knowledge about the environment is assumed in the global dynamic window approach. Such knowledge can be provided in form of a model of the environment or be acquired during motion through sensing.

4.1 Free Space Connectivity

To exploit information about the connectivity of the free space, a model of the environment is required. The model-based dynamic window approach [7] incorporates sensory data and a given map of the environment to determine collision-free motion. A similar technique could also be adapted for the global dynamic window approach. The work presented in this paper is restricted to the case where no a priori knowledge about the environment is available and hence global planning algorithms cannot be applied.

To collect information about the connectivity of the free space sensory information is merged into a map. In order to achieve real-time performance for the overall algorithm no preprocessing of the sensory data is performed. At each servo tick the sensory data is translated into configuration space obstacles that are represented in an occupancy grid.

This simple approach is motivated by the fact that only connectivity information about the free space is needed. Furthermore, the mobile base used in the experiments presented in Section 5 has little slippage resulting in maps that are very accurate. For collision avoidance the motion integration error is irrelevant, as the map is frequently updated with very accurate sensory information.

4.2 Navigation Function

Since the environment is represented as an occupancy grid, a grid-based navigation function is a natural and efficient choice for a global planning algorithm. The global dynamic window approach combines the dynamic window approach for reactive obstacle avoidance with the global, local minima-free navigation function NF1 [1, 12]. This function is computed using a wave-propagation technique starting at the goal. It labels cells in the occupancy grid with the L^1 distance to the goal, taking into account obstructions by obstacles. The result is a local minima-free potential function with a unique minimum at the goal.

Employed with classical motion planning algorithms the navigation function NF1 has the disadvantage of producing trajectories that graze obstacles. Selecting motion commands using the dynamic window approach eliminates this problem, since a minimum clearance from obstacles is maintained.

The classical motion planning algorithm [1, 12] computes NF1 for the entire occupancy grid. This is motivated by the fact that the same NF1 can be reused for every location of the robot as long as the environment does not change. The global dynamic window approach recomputes the NF1 each time a motion command is selected, allowing it to operate in unknown and dynamic environments. Hence it is not necessary and not desirable to compute NF1 for the entire grid. Instead, NF1 is computed in a rectangular region aligned with the goal heading. The width of this rectangular region is increased until the robot's current position is reached by the wave front.

Figure 3 shows a narrow NF1 for an unobstructed path and a wider NF1 for an obstructed path. The NF1 is shown as gradient colors, the robot is the black dot at the bottom of the NF1 and the goal position

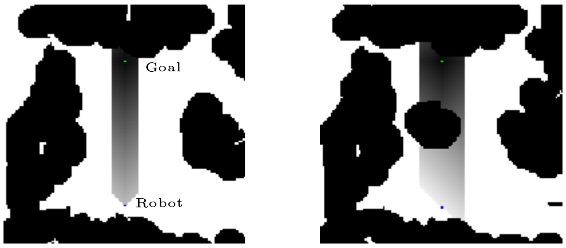


Figure 3: Navigation function computation

is the gray dot in the darker region of the NF1, obstacles are shown in black. Although widening the NF1 may cause partial recomputation of previously computed regions, this modification to the NF1 algorithms greatly reduces the cost of NF1 computation.

4.3 Objective Function

The objective function of the dynamic window and the holonomic dynamic window approach described in Section 3.3 can easily be modified to incorporate the navigation function described in Section 4.2. The function $align(\vec{p}, \vec{v})$ is replaced by the function $nf1(\vec{p}, \vec{v})$. This function's value increases if $\vec{v}$ is aligned with the gradient of the navigation function at the robot's location $\vec{p}$. This makes the global dynamic window approach immune to local minima, since NF1 is a local minima-free potential.

In addition, the function $\Delta nf1$ is added to the objective function. Its value indicates by how much a motion command is expected to reduce the value of the NF1 during the next servo tick. This favors motion commands that quickly reduce the distance to the goal. The objective functions Ω_g for the global dynamic window is then defined as:

$$\begin{aligned}\Omega_g(\vec{p}, \vec{v}, \vec{a}) &= \alpha \cdot nf1(\vec{p}, \vec{v}) + \beta \cdot vel(\vec{v}) + \\ &\quad \gamma \cdot goal(\vec{p}, \vec{v}, \vec{a}) + \delta \cdot \Delta nf1(\vec{p}, \vec{v}, \vec{a}).\end{aligned}$$

The value of $nf1(\vec{v}, \vec{p})$ can be determined by examining the neighbors of the grid cell that corresponds to the robot's location. However, since NF1 is grid-based, its gradient can only be a multiple of 45°, resulting in unnatural behavior along passages that are not grid-aligned. By examining neighbors at a constant distance from the cell that corresponds to the robot's position this behavior can be improved. Similar to the holonomic dynamic window approach, the desired direction of motion is modified to maintain a safe lateral distance to obstacles.

5 Experimental Results

The global holonomic dynamic window approach has been implemented and tested on the Nomad XR4000 mobile base by Nomadic Technologies, Inc. shown in Figure 4. This base moves at omnidirectional translational velocities of up to $1.2\frac{m}{s}$ and accelerations of up to $1.5\frac{m}{s^2}$. It is equipped with a SICK laser range finder with a field of view of 180° and an accuracy of up to 1cm. Using the on-board 450 MHz PC, servo rates of above 15 Hz are achieved for map sizes of $30m \times 30m$ at a resolution of $5cm$. The robot navigates reliably with very high velocities in tight environments. In long but not necessarily wide open areas the base moves at its maximum velocity.

Figure 4: The Nomad XR4000

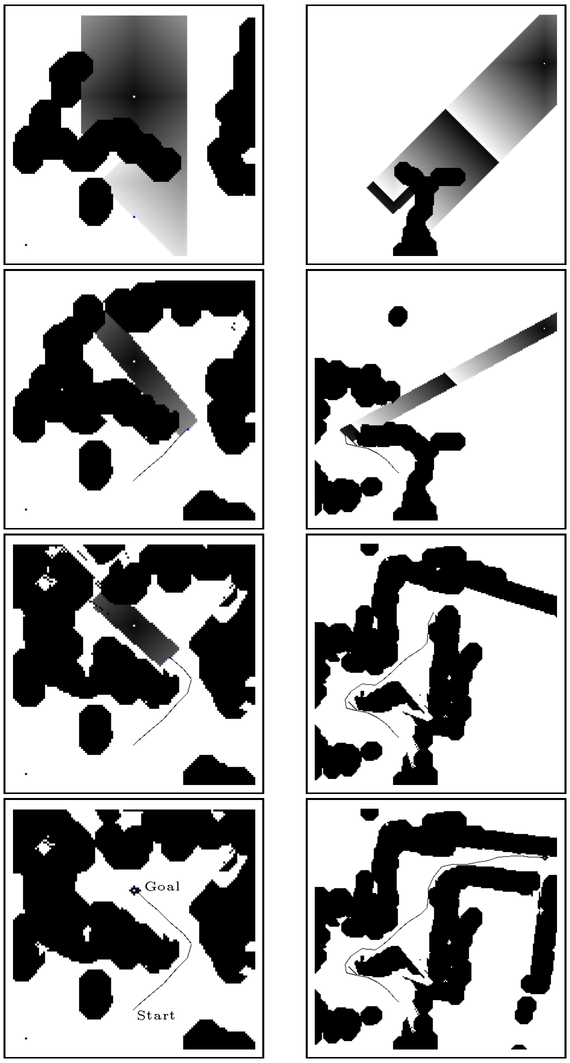
Figure 5 shows two example executions of the global holonomic dynamic window approach. The velocities achieved in both examples are above $1.0\frac{m}{s}$. Obstacles are shown in black. The trajectory of the robot is shown as a line and the current NF1 is shown as a gradient. The four images on the left represent a $6m \times 6m$ map, the images on the right a $10m \times 10m$ map, both with grid cell sizes of $5cm \times 5cm$. In both examples the robot started without any knowledge about the environment. The first image in both cases corresponds to the robot at rest; the obstacles are only those visible from its position. As the robot starts moving, obstacles are added to the map and the NF1 is recomputed correspondingly, until the robot reaches the goal.

The third image on the right side shows a situation where sensory information indicates that the goal is obstructed. Hence, no NF1 can be constructed and the global dynamic window approach reduces to the dynamic window approach. When updated sensory information shows that the goal is not obstructed, as can be seen in the fourth image, the NF1 is reconstructed.

6 Conclusion

The dynamic window approach to obstacle avoidance was extended to holonomic robots. Taking advantage of the increased maneuverability of such robots, obstacle avoidance can be performed in dynamic environments at high velocities.

Integrating the holonomic dynamic window approach with an efficient motion planning method results in the global dynamic window approach. It is



$6m \times 6m$ map $\quad$ $10m \times 10m$ map

Figure 5: Two example executions

an effective framework for global, reactive robot navigation. Global goal behavior is integrated with local obstacle avoidance. The global dynamic window approach is particularly well suited for unknown and changing environments. It allows the robot to navigate safely and at high speeds to reach a goal position without prior knowledge of the environment.

Acknowledgments

The authors would like to thank Philippe Pignon, Richard LeGrand, and Jake Sprouse from Nomadic Technologies and Kyong-Sok Chang, Bob Holmberg, and Diego Ruspini from Stanford for their helpful insights and discussion in preparing this paper. The financial support of Nomadic Technologies, Inc. is gratefully acknowledged.

References

[1] Jérôme Barraquand and Jean-Claude Latombe. Robot motion planning: A distributed representation approach. *Intl. J. of Robotics Research*, 10(6):628–649, 1991.

[2] Johann Borenstein and Yoram Koren. The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–88, 1991.

[3] Wonyun Choi and Jean-Claude Latombe. A reactive architecture for planning and executing robot motions with incomplete knowledge. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, volume 1, pages 24–29, 1991.

[4] Hans Jacob S. Feder and Jean-Jacques E. Slotine. Real-time path planning using harmonic potentials in dynamic environments. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, volume 1, pages 874–811, 1997.

[5] W. Feiten, R. Bauer, and G. Lawitzky. Robust obstacle avoidance in unknown and cramped environments. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, volume 3, pages 2412–7, 1994.

[6] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1):23–33, March 1997.

[7] Dieter Fox, Wolfram Burgard, Sebastian Thrun, and Armin B. Cremers. A hybrid collision avoidance method for mobile robots. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, volume 2, pages 1238–43, 1998.

[8] Huosheng Hu and Michael Brady. A bayesian approach to real-time obstacle avoidance for a mobile robot. *Autonomous Robots*, 1:69–92, 1994.

[9] Maher Khatib and Raja Chatila. An extended potential field approach for mobile robot sensor-based motions. In *Proc. of Intl. Conf. on Intelligent Autonomous Systems*, pages 490–496, 1995.

[10] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Intl. J. of Robotics Research*, 5(1):90–8, 1986.

[11] D. E. Koditschek. Exact robot navigation by means of potential functions: Some topological considerations. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, pages 1–6, 1987.

[12] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[13] Sebastian Thrun et al. Map learning and high-speed navigation in RHINO. In *Artificial Intelligence and Mobile Robots*, chapter 1, pages 21–52. MIT/AAAI Press, 1998.

[14] Reid Simmons. The curvature-velocity method for local obstacle avoidance. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, volume 4, pages 2275–82, 1996.

[15] Leena Singh, Harry Stephanou, and John Wen. Real-time robot motion control with circulatory fields. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, volume 3, pages 2737–42, 1996.