15th Triennial World Congress, Barcelona, Spain

A PROVABLY CONVERGENT DYNAMIC WINDOW APPROACH TO OBSTACLE AVOIDANCE

Petter Ögren *,1 **Naomi E. Leonard** **,2

* *Division of Optimization and Systems Theory, Royal Institute of Technology (KTH), SE-100 44, Stockholm, SWEDEN*
** *Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544, USA*

Abstract: The dynamic window approach is a well known navigation scheme developed in Fox *et al.* (1997) and extended in Brock and Khatib (1999). It is safe by construction and has been shown to perform very efficiently in experimental setups. However, one can construct examples where the proposed scheme fails to attain the goal configuration. What has been lacking is a theoretical treatment of the algorithm's convergence properties. Here we present such a treatment. Furthermore, we highlight the similarity between the Dynamic Window Approach and the Control Lyapunov Function and Receding Horizon Control synthesis put forth by Primbs *et al.* (1999). Inspired by these similarities we propose a version of the Dynamic Window Approach that is provably convergent.

Keywords: Mobile Robots, Obstacle Avoidance, Predictive Control, Lyapunov Function

1. INTRODUCTION

The problem of robotic motion planning is a well-studied one, see for instance (Latombe, 1991). Since we present here a new development of the work in (Fox *et al.*, 1997) and (Brock and Khatib, 1999), we refer to these papers for a thorough discussion on related work. We note, however, that the Dynamic Window Approach still remains one of the best approaches to efficient real time obstacle avoidance in an unknown environment. The organization of this paper is as follows. In Section 2 we review the work of (Fox *et al.*, 1997) and (Brock and Khatib, 1999). In Section 3 we briefly outline the synthesis suggested by (Primbs *et al.*, 1999). Our proposed scheme is explained in detail in Section 4. Finally, Section 5 discusses the theoretical properties of our approach and Section 6 contains the conclusions.

2. THE DYNAMIC WINDOW APPROACH AND ITS EXTENSION

The Dynamic Window approach (Fox *et al.*, 1997) is an obstacle avoidance method that takes into account the dynamic and kinematic constraints of a mobile robot (many of the vector field and vector field histogram approaches do not). The basic scheme involves finding the *admissible controls*, those that allow the robot to stop before hitting an obstacle while respecting the above constraints. Then an optimization is performed over those

[1] petter@math.kth.se, Research supported by the Center for Autonomous Systems, KTH, Sweden
[2] Research partially supported by the National Science Foundation under grant CCR-9980058

CLF	RHC
Global information	Local information
Stability oriented	Performance oriented
Off-line analysis	On-line computation
Hamilton Jacobi Bellman type	Euler-Lagrange type

Table 1. Complementary properties of the two approaches

admissible controls to find the one that gives the highest utility in some prescribed sense. There are different suggestions for the utility function in Fox *et al.* (1997) and Brock and Khatib (1999), including components of velocity alignment with preferred direction, large minimum clearances to obstacles, possibility to stop at goal point and the norm of the resulting velocity vector (large being good). Brock and Khatib (1999) extended the work of Fox *et al.* (1997) by looking at holonomic robots (Fox considered synchro drive ones) and more importantly by adding to the problem information about connectivity to the goal. The latter was done by replacing the goal heading term with a *navigation function* defined as the length of the shortest (unobstructed) path to the goal (Latombe, 1991; Barraquand and Latombe, 1991). Thus, they were able to eliminate the local minima problems present in so many obstacle avoidance schemes (hence the term Global in the title of Brock and Khatib (1999)). The experimental results reported in (Fox *et al.*, 1997) and (Brock and Khatib, 1999) are excellent, showing consistent safe performance at speeds up to 1.0 m/s with a Nomadic Technologies XR4000 robot, (Brock and Khatib, 1999). The results demonstrate an algorithm that is safe by construction (in the sense that the robot never hits obstacles) and displays high efficiency in extensive experimental tests. But although Brock and Khatib argue that the use of a navigation function makes the approach 'Global', it is never shown. In fact, examples can be constructed where the robot enters a limit cycle, never reaching the goal or actually consistently moves away from the goal (see Section 4.4).

3. THE CONTROL LYAPUNOV FUNCTION AND RECEDING HORIZON PERSPECTIVE

In a interesting paper by Primbs *et al.* (1999), the connection between Control Lyapunov Functions (CLF) and Receding Horizon Control (RHC) is investigated. They note the complementary properties shown in Table 1. In view of these properties they suggest the following framework to combine the complementary advantages of each approach. The control law is chosen to satisfy a receding horizon optimal control problem under constraints that ensure the existence of a CLF. The problem becomes one of finding a control u and a CLF $V(x)$ that satisfy (1) through (4) as follows:

$$\inf_{u(\cdot)} \int_t^{t+T} (q(x) + u^T u) d\tau \quad (1)$$

$$\text{s.t. } \dot{x} = f(x) + g(x)u \quad (2)$$

$$\frac{\partial V}{\partial x}(f + gu) \leq -\epsilon\sigma(x(t)) \quad (3)$$

$$V(x(t+T)) \leq V(x_\sigma(t+T)), \quad (4)$$

where $q(x)$ is a cost on states, $\epsilon > 0$ is a scalar,$T > 0$ is the horizon length, $\sigma(x)$ is positive definite and x_σ is the trajectory when applying a pointwise minimum norm control scheme (for details see (Primbs *et al.*, 1999)). This formulation inspires our choice of a more formal, continuous time formulation of the Dynamic Window Approach, allowing us to prove convergence.

4. A PROVEN DYNAMIC WINDOW APPROACH

4.1 Robot model, environment and navigation function

In the main parts of this paper we will use the notation $x = (r, \dot{r}) = (r_x, r_y, \dot{r}_x, \dot{r}_y)$ for the state of the system. We adopt the robot model from (Brock and Khatib, 1999), which is basically a double integrator in the plane $\ddot{r} = u$, $r \in \mathbb{R}^2$ with bounds on the control $||u|| \leq u_{max}$ and on the velocity $||\dot{r}|| \leq v_{max}$. Note that it was shown in (Lawton *et al.*, 2001/2002) that an off axis point on the unicycle robot model described by $\dot{r}_x = v\cos\theta$, $\dot{r}_y = v\sin\theta$, $\dot{\theta} = \omega$, $\dot{v} = F/m$, $\dot{\omega} = \tau/J$ can be feedback linearized to $\ddot{r} = u$. For the environment we assume that the robot's sensors can supply an occupancy grid map, i.e. a rectangular mesh with each block being marked as either free or occupied, over the immediate surroundings. Thus a map can be incrementally built as the robot moves around. We assume, as did Brock, that the simultaneous localization and mapping (SLAM) problem is solved for us.

A navigation function $NF(r)$, (Latombe, 1991; Barraquand and Latombe, 1991) basically maps every free space position to the length of the shortest collision free path going from that position to the goal point. It is shown in (Brock and Khatib, 1999) how to deal with the case when the robot at first only knows its immediate surroundings by use of its sensors. The idea is to assume free space at the unknown positions and then recalculate the navigation function when sensor data showing the opposite arrives. In this way the robot guesses good paths and updates them when new information arrives. These updates are made at a time scale much slower than the actual motion control so in our considerations below we assume the map to be static. Brock and Khatib (1999) used the gradient of the navigation function as the desired heading instead of using just the goal direction as Fox did (Fox *et al.*, 1997).

4.2 Control Scheme

We adopt the basic structure from Primbs *et al.* (1999) as seen above, but since we have state constraints, i.e. obstacles, we make some changes. We state our control problem as

$$\inf_{u(\cdot)\in CS_{dss}} V(x(t+T)) \tag{5}$$
$$\text{s.t. } \ddot{r} = u \tag{6}$$
$$\dot{V}(x,u) \le 0 \tag{7}$$

where $V(x)$ is the Control Lyapunov Function defined in equation (8), T is the horizon length and CS_{dss} is the set of Dissipative Safely Stopping Control Sequences of Definition 4.3 below. We have substituted the equations (1) and (4) with (5) since the pointwise minimum norm properties aren't applicable in this setting. Furthermore, the inequality in (7) cannot be bounded away from zero since the robot might have to stop. As can be seen above, the Receding Horizon Control part (5) and the dynamics (6) corresponds to the optimization in the dynamic window of (Brock and Khatib, 1999) and the Control Lyapunov Function part (7) corresponds in spirit to the use of the $NF(x)$ to avoid local minima. Actually the equation (7) condition will be incorporated into the CS_{dss} constraints; it is just stated here for clarity. The Control Lyapunov Function is

$$V(x) = \frac{1}{2}\dot{r}^T\dot{r} + kNF(r), \tag{8}$$

where $NF(r)$ is the navigation function as explained above. The basic idea for the stability proof is to first write the problem as a conservative system with an artificial potential and then introduce a dissipative control term. In the conservative system we choose the artificial potential to be $kNF(r)$, where k is a positive constant. Incorporating the upper bounds on the control magnitude and the ones on velocity, we define the dissipative controls as follows.

Definition 4.1. (Dissipative Controls, $C_d(r,\dot{r})$).

$$C_d(r,\dot{r}) = \{u : \\ u = u_c + u_d, \\ u_c = -k\nabla NF \\ \text{if } \dot{r} = 0 \text{ then } u_d = 0 \text{ else} \\ u_d :\ u_d^T\dot{r} < -\epsilon||\dot{r}|| < 0, \\ ||u|| \le u_{max}, \\ \text{and if } ||\dot{r}|| \ge v_{max} \text{ then } u^T\dot{r} \le 0\}$$

for some given $\epsilon > 0$. We write $u \in C_d(r,\dot{r})$.

A typical shape of the $C_d(r,\dot{r})$ set is shown in Figure 1. Note that $||u_c|| = ||k\nabla NF(r)|| = k$; since $NF(r)$ is the length of the shortest path, $||\nabla NF(r)|| = 1$ (the decrease in the length of the shortest path while travelling along the shortest path is equal to the distance travelled). Thus u_c lies on a circle of radius k. The outer circle of radius u_{max} bounds the control set. Now the

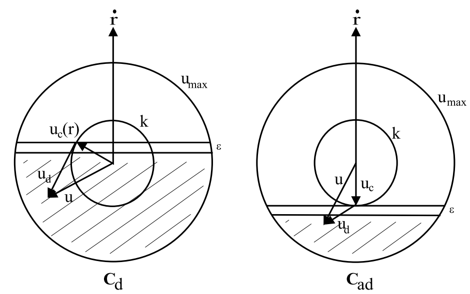

Fig. 1. The Dissipative Controls, $C_d(r,\dot{r})$ and $C_{ad}(\dot{r})$.

problem is to make sure the robot does not run into obstacles. In the standard Dynamic Window Approach this is taken care of by choosing among admissible controls in the optimization and here we shall do the same. To summarize we have the following control system:

$$\ddot{r} = u, \\ u(t) \in C_d(r,\dot{r}), \\ r(t) \in \{\text{Free space}\}$$

If we look closer at the set $C_d(r,\dot{r})$ we see that if $k+\epsilon < u_{max}$ there is a set of controls that are always included, independent of r, centered around $u = -|u_{max}|\frac{\dot{r}}{||\dot{r}||}$, i.e. full brake (it is perhaps not surprising that full brake is always a dissipative strategy). We call this subset the Always Dissipative Controls, $C_{ad}(\dot{r}) \subset C_d(r,\dot{r})\ \forall r$. It is also depicted in Figure 1. This is the set of controls that we can always use to prevent a collision while respecting the constraints guaranteeing stability. Now, on our way to defining the Dissipative Safely Stopping Control Sequences of equation (5) above we need some more definitions.

Definition 4.2. (Safe Set, S). We call the state $x_0 = (r_0, v_0)$ safe if there exists a time $T \ge 0$ and a control sequence $u(\cdot)$ such that

$$r(t) = r_0,\ \dot{r}(t) = v_0,\ \ddot{r} = u \text{ implies}$$
$$\dot{r}(t+T) = 0,\ r(s) \in \{\text{Free Space}\}\ \forall s \in [t, t+T],$$

i.e. there is a way to stop without hitting an obstacle. We write $x_0 \in S$.

Determining whether a state belongs to S can be quite computationally expensive, since the set of control sequences is infinite dimensional. We need a computationally tractable way to make sure the current state is always within S. To achieve this we only consider a finite set of control sequences in C_{ad}. These are the Discretized Always Dissipative Controls $C_{dad}(\dot{r}) \subset C_{ad}(\dot{r})$, i.e. control signals that have constant magnitude and constant direction *relative* to the velocity direction $\dot{r}/||\dot{r}||$. The set is depicted in Figure 3 and will be investigated in more detail in Section 4.3 below. A set of corresponding trajectories can be found in Figure 2. The strategy for avoiding collisions is to make sure that we stay in the Safe Set S. To make this more precise we define the Dissipative Safely Stopping Control Sequences.

Definition 4.3. (CS_{dss}). Given times T_1 and T_2 a control sequence/state pair $(u(\cdot), x_0)$ is called Dissipative Safely Stopping if

$$
\begin{aligned}
&u(s) \in C \;\forall s \in [t, t+T_1], \qquad\qquad (9)\\
&\quad C = C_d(r,\dot{r}) \text{ if } ||\dot{r}|| \geq v_{min} \text{ or } \dot{r}^T u \geq 0 \text{ else},\\
&\quad C = C_{dad}\\
&u(s) \in C_{dad}(\dot{r}) \;\forall s \in [t+T_1, t+T_1+T_2],\\
&r(s) \in \{\text{Free Space}\} \;\forall s \in [t, t+T_1+T_2],\\
&\dot{r}(t+T_1+T_2) = 0
\end{aligned}
$$

i.e. it starts with a dissipative control, ends with a discretized always dissipative control, doesn't hit any obstacles and stops at the end.

The somewhat awkward construction on line 2 of equation (9) is needed in the convergence proof (Theorem 5.1). It guarantees that the speed will not approach zero as time goes to infinity. Instead we make sure that if the speed drops below the design parameter v_{min}, the robot brakes and makes a fresh start, accelerating along the shortest path towards the goal. v_{min} is to be chosen small enough so that the rule only applies in exceptional cases. The time T_1 is to be chosen at least as big as the time step length of the controller and $T_2 = v_{max}/(k+\epsilon)$ i.e. longer than the maximal stopping time using any C_{dad} control. T in (5) is set to $T = T_1 + T_2$. We will see in Theorem 5.2 below how this guarantees staying in S.

Now we are set to apply the control scheme outlined above, i.e.

(1) Find $u(\cdot)$ according to equation (5).
(2) Apply the first C part of it for one time step $T_s \leq T_1$.
(3) Go to (1).

Remark 4.1. Note that we never actually apply the T_2 part of the control sequence. It is there to guarantee that there always exists a dissipative way to avoid collisions. This is similar to always making sure you can stop in the visible part of the road when driving a car.

Remark 4.2. Excess computational resources can be exploited by increasing the time T_1 and thereby the time horizon of the optimization. If there on the other hand is a shortage, the search space can be reduced by only considering constant C_d controls or even a finite subset of constant ones. As long as Assumption 5.1 is true the proofs are still valid. In the same way the discretization set chosen in C_{dad} can be made larger or smaller.

4.3 A closer look at C_{dad}

Some resulting trajectories from applying a C_{dad} control are depicted in Figure 2.

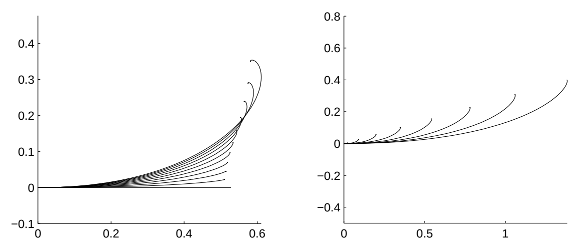

Fig. 2. Left: A set of trajectories with different C_{dad} controls. $\dot{r}_0 = 1$, $||u|| = 1$, $\alpha \in \{0, 5, 10, \ldots, 50\}$, where α is the angle between u and $-\dot{r}$. Right: Varying initial velocity scales curves.

As can be seen, the four upper most trajectories (corresponding to angles $35 \ldots 50$) are very close to one another. Thus if the shortest one is not free of collisions, the other ones probably aren't either. Therefore, there is no need to include controls of angles higher than ≈ 35 in C_{dad}. As can be seen from Figure 1, a smaller C_{ad} set admits a larger k and thus a larger maximal acceleration. An angle of 35 gives the value $k = 0.8u_{max}$.

As will be shown in the lemma below (and seen in Figure 2), the trajectories just scale in both dimensions when changing initial velocity and u_{max}, so this rule of thumb for choosing k is valid for all cases.

Lemma 4.1. (Scaling of Trajectories). The trajectory resulting from applying a constant (relative to the velocity direction) control has the curvature

$$K(s) = \frac{b}{(v_0^2 + 2as)} \qquad (10)$$

and is uniformly rescaled by a factor of

$$h = \frac{h_v^2}{h_u},$$

when changing the initial velocity by a factor h_v and the control magnitude by a factor h_u.

The proof is a straightforward application of first calculations in a moving frame and then the fact that rescaling a curve corresponds to changing the curvature as $K_{new}(s) = \frac{1}{h} K_{old}(hs)$. We omit the details for lack of space.

This means that we don't have to simulate the differential equations to evaluate equation (5). We can just store one precalculated trajectory for each point in the C_{dad} set and scale it depending on initial velocity and u_{max}. Then we determine if the whole control sequence is safe and dissipative, i.e. belonging to CS_{dss}, and evaluate the utility function.

4.4 Example of Convergence Failure of Previous Approach

The Utility function of (Brock and Khatib, 1999) that is to be maximized is

$$\Omega(p, v) = \alpha \text{align}(p, v) + \beta \text{vel}(v) + \gamma \text{goal}(p, v),$$

where p, v is the resulting position and velocity from applying a control signal for one time step. 'goal' is a binary function reflecting whether we will end up in the goal position or not. In the case when the robot is far from the goal the utility function equals

$$\Omega(p, v) = \alpha(1 - \frac{|\theta|}{\pi}) + \beta \frac{||v||}{v_{max}} + 0,$$

where θ is the angle between the goal heading and the direction of motion. Consider a 'T' shaped, very narrow corridor, with the robot being in the top left end and the goal being in the bottom end. This will leave the robot accelerating maximally towards the right. If the corridor is long and narrow enough the speed is going to be too great to allow a right turn at the intersection. Thus the robot will continue away from the goal. And if the corridor is narrow enough to make the velocity term outweigh the alignment term, the robot will keep going to the right until the corridor ends and the admissibility constraint forces it to stop. Thus the control scheme lacks not only global asymptotic stability, but also global stability as the trajectory might diverge infinitely away from the goal.

5. PROOF OF ASYMPTOTICAL STABILITY AND SAFETY

Consider the situation depicted in Figure 3, with the robot halted at a position very close to an obstacle corner. The shortest path is always composed of straight line segments, grazing occupied occupancy grids at their corners. We shall make the following assumption.

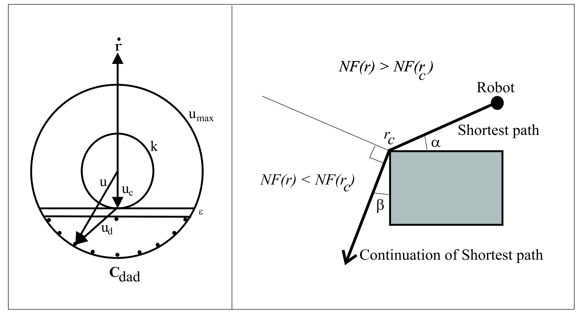

Fig. 3. Left: The C_{dad} set. Right: Note the border between the points closer than r_c to the goal and the points further away. The angles $\alpha, \beta, \alpha + \beta \in [0, \pi/2]$.

Assumption 5.1. Let $R_c(r)$ be the finite set of obstacle corners r_c such that $NF(r_c) \leq NF(r)$. Let furthermore the robot be at rest at $x(t) = (r, 0)$. For an arbitrary r', we assume that

$$\begin{aligned}
&\forall r_c \in R_c(r') \ \exists \delta : \\
&\text{if } ||r_c - r|| \leq \delta \\
&\text{then } \exists cs \in CS_{dss} : \\
&NF(r(t+T)) < NF(r_c), \ \dot{r}(t+T) = 0
\end{aligned}$$

i.e. if the robot is just close enough to a corner and standing still, there is a CS_{dss} control sequence that brings it to a halt beyond that corner.

Note that this is not unreasonable even in the worst case when the angles $\alpha = \beta = 0$ (see Figure 3). Before we formulate the main theorem of this paper we need a few lemmas.

Lemma 5.1. (Control Lyapunov Function). The function

$$V(x) = \frac{1}{2} \dot{r}^T \dot{r} + kNF(r),$$

is a Control Lyapunov Function and satisfies the following inequality

$$\dot{V}(x) \leq -\epsilon ||\dot{r}||.$$

Furthermore, the decrease of $V(x)$ between two stops at times t_i and t_{i+1} along trajectories satisfies

$$\Delta V \leq -\epsilon l$$

where l is the length of the trajectory between times t_i and t_{i+1}.

Proof: The candidate Lyapunov Control Function is $V(x) = \frac{1}{2} \dot{r}^T \dot{r} + kNF(r)$, which is clearly positive definite with a global minimum at $x_{goal} = (r_{goal}, 0)$. Differentiating with respect to time gives

$$\begin{aligned}
\dot{V}(x) &= \dot{r}^T u + k\dot{r}^T \nabla NF(r). \\
&= \dot{r}^T (u_c + u_d + k \nabla NF(r)) \leq -\epsilon ||\dot{r}||,
\end{aligned}$$

by the constraints on u. The last part of the Lemma is seen to hold by considering

$$\Delta V = \int_{t_1}^{t_2} \dot{V}\, dt \leq \int_{t_1}^{t_2} (-\epsilon)||\dot{r}||\, dt = -\epsilon l$$

■

Theorem 5.1. (Asymptotic Stability). If the control scheme in (5) is used and if there is a traversable path from start to goal in the occupancy grid. Then the robot will reach the goal position.

Proof: By Lemma 5.1 we have that $\dot{V}(x) \leq -\epsilon||\dot{r}||$. Thus the system is stable in the sense of Lyapunov. Since the control scheme does not permit the robot's velocity to slowly approach zero (see equation (9)), the remaining difficulty is to show that it does not stop infinitely many times. We will therefore make a worst case analysis of the situation when the robot does stop a lot of times. We use the notation $V(r) = V(r, 0)$ for the Lyapunov function at a position r with zero velocity, $\dot{r} = 0$.

After a stop at some position r_1, equation (5) makes the robot accelerate along the shortest path to the goal. Since the dissipative accelerations are all weaker than the C_{dad} breaking controls, the robot will have travelled at least half the distance to the next corner r_c on the path to the goal before it has to do any evasive maneuvers and possibly stop. This distance from r_1 to r_c is equal to $NF(r_1) - NF(r_c)$. Therefore, between two consecutive stops at r_1 and r_2 we have (by Lemma 5.1) $V(r_1) - V(r_2) \geq \frac{\epsilon}{2}(NF(r_1) - NF(r_c))$. But when $\dot{r} = 0$ we have $V(r) = kNF(r)$, thus

$$V(r_1) - V(r_2) \geq \frac{\epsilon}{2k}(V(r_1) - V(r_c)).$$

In the worst case this means that the robot position will show exponential convergence (in the number of stops) towards some corner r_c' that continues to be the closest on the shortest path to the goal, i.e. after m stops we have

$$V(r_m) - V(r_c') \leq (V(r_1) - V(r_c'))(1 - \frac{\epsilon}{2k})^{(m-1)}.$$

but since r_c' is on the shortest path to the goal $k||r_m - r_c'|| = V(r_m, 0) - V(r_c', 0)$ and therefore there exists M such that

$$||r_m - r_c'|| \leq \frac{1}{k}(V(r_1) - V(r_c'))(1 - \frac{\epsilon}{2k})^M \leq \delta.$$

By Assumption 5.1 there now exists a control to bring the robot to a halt beyond r_c'. The minimization of (5) will thus provide a control that makes $V(x(t+T)) < V(r_c', 0)$ and the corner can never again be on the shortest path to goal. Thus, even in the worst case, in a finite number of stops all corners will be passed. ■

Note that the proof is an extreme worst case analysis. A more probable scenario is that at each stop the robot will face a new corner, and that the number of stops will be very few during the execution, see Remark 4.1.

Theorem 5.2. (Safety). If the control scheme in (5) is used and if the robot starts at rest in an unoccupied position. Then the robot will not run into an obstacle.

Proof: The proof relies on the recursive structure of CS_{dss}. The set CS_{dss} (of safely stopping sequences that we are choosing from) is never empty since we can always choose the remaining (not yet applied) part of the previous CS_{dss} control sequence as our new CS_{dss} control sequence. ■

6. CONCLUSIONS

In this paper we have first presented the well known Dynamic Window Approach to fast and safe obstacle avoidance in an unknown environment. We then recast the approach in a continuous nonlinear control framework suggested by Primbs *et al.* (1999). With a few changes to the basic scheme we were able to prove convergence to the goal position. This is significant since the earlier scheme could be subject to limit cycles and even divergence.

REFERENCES

Barraquand, J. and J. Latombe (1991). Robot motion planning: A distributed representation approach. *Intl. J. of Robotics Research* pp. 10(6)628–649.

Brock, O. and O. Khatib (1999). High-speed navigation using the global dynamic window approach. *IEEE International Conference on Robotics and Automation, Detroit, Michigan.*

Fox, D., W. Burgard and S. Thrun (1997). The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine, March* pp. 23–33.

Latombe, J. (1991). *Robot Motion Planning.* Kluwer Academic Publishers. Boston, MA.

Lawton, J., B. Young and R. Beard (2001/2002). A decentralized approach to elementary formation maneuvers. *To appear in IEEE Transactions on Robotics and Automation.*

Primbs, J., V. Nevistic and John C. Doyle (1999). Nonlinear optimal control: A control lyapunov function and receding horizon perspective. *Asian Journal of Control* pp. March, Vol 1, No. 1, pp. 14–24.