Latin American Applied Research 38:289-298 (2008)

IMPROVED DYNAMIC WINDOW APPROACH BY USING LYAPUNOV STABILITY CRITERIA

H. BERTI†, A.D. SAPPA‡ and O.E. AGAMENNONI*

†*Facultad de Ingeniería (UNLPam), L6360GLH General Pico, ARGENTINA*
hberti@ing.unlpam.edu.ar
‡*Computer Vision Center, 08193 Bellaterra, Barcelona, SPAIN*
angel.sappa@cvc.uab.es
* *Departamento de Ingeniería Eléctrica y de Computadoras (UNS), B8000CPB Bahía Blanca, ARGENTINA*
Instituto de Investigaciones en Ingeniería Eléctrica (UNS-CONICET)
Comisión de Investigaciones Científicas (CIC)
oagamen@uns.edu.ar

***Abstract*— This paper presents improvements over the Dynamic Window Approach (*I-DWA*), used for computing in real time autonomous robot navigation. A novel objective function that includes Lyapunov stability criteria is proposed. It allows to guarantee a global and asymptotic convergence to the goal avoiding collisions and resulting in a more simple and self-contained approach. Experimental results with simulated and real environments are presented to validate the capability of the proposed approach. Additionally, comparisons with the original DWA are given.**

***Keywords*— Autonomous Mobile Robot, Lyapunov Stability.**

I. INTRODUCTION

Autonomous robot navigation involves the real time achievement of user defined goal/s. The autonomy degree of a given robotic system fixes or defines both the capability of adaptation in front of environment changes and the abstraction level in which a given goal can be represented. For example, the achievement of a given goal in a static and known environment can be tackled with a *global planning strategy*. On the contrary, unknown or partially known environments, as well as dynamic environments, should be tackled by means of *reactive navigation strategies*. These reactive strategies allow solving unexpected events in real time by means of the use of sensors in order to capture the surrounding environment.

Several autonomous robot navigation approaches were proposed during the last decades. Early techniques were based on the use of an artificial potential field (e.g., Khatib, 1986; Khatib and Chatila, 1995; Borenstein and Koren, 1989; among others): an attractive force produced by the goal drives the robot to the objective, while at the same time, repulsive forces produced by the obstacles keep the robot away from them. Since then, several improvements were introduced giving rise to more evolved techniques such as: *Virtual Field Histogram* (VFH; Borenstein and Koren, 1991), *Curvature-Velocity Method* (CVM; Simmons, 1996) and *Dynamic Window Approach* (DWA; Fox *et al.*, 1997). The CVM (Simmons, 1996) and DWA (Fox *et al.*, 1997) are widely used approaches since a high speed navigation can be reached. They search for *control commands* (v, w) directly in the velocity space, rather than in the position space or in the configuration space (Latombe, 1991). Similarly, in Shimoda *et al.* (2005) a trajectory space is used for searching the control commands (*steering angle* and *velocity*). In these cases, control commands are selected by maximizing an objective function, which includes criteria such as: speed, goal-directedness and safety. At the same time that function incorporates constrains in velocity space from the obstacles and from the robot. In spite of these advantages, a hard constraint of these techniques is that they ignore the way in which the robot approaches the goal, so convergence criteria are not considered.

Extensions to the original DWA have been proposed in Brock and Khatib (1999), Arras *et al.* (2002), Philippsen and Siegwart (2003) and Ögren and Leonard (2005), to mention a few. Brock and Khatib (1999) present a Global-DWA to avoid the local minima problems by using connectivity information about the free space. However this global feature is never shown (Ögren and Leonard, 2005). A Reduced-DWA, to speed up the translational velocity selection, is proposed by Arras *et al.* (2002). As a result a *dynamic line* is obtained, which requires much less processing power. This velocity selection is not appropriate when the robot is not oriented towards the goal—for instance when the angle between the robot orientation with respect to the goal is higher than 90 degrees. A more elaborated method, which integrates three different approaches (DWA, elastic band and NF1), is

introduced in Philippsen and Siegwart (2003). This integration mitigates the drawbacks of DWA. Finally, Ögren and Leonard (2005) also combines different elements from the original DWA to guarantee convergence. Although Philippsen and Siegwart (2003) and Ögren and Leonard (2005) guarantee convergence to the goal, none of them improve the original DWA, they add other approaches for compensating DWA drawbacks, which gives as a result a more expensive and complex strategy.

Lyapunov stability theory can be used to tackle the aforementioned global convergence problem. For instance, Aicardi *et al.* (1995) proposes an approach based on Lyapunov quadratic functions of a polar representation; it reaches both smooth trajectories and a global convergence to the goal. The main drawback of this approach is that it needs to know the whole scene in advance in order to avoid collisions. This approach has been extended in Secchi *et al.* (1999), by incorporating the use of virtual repulsive forces (Borenstein and Koren, 1991) and impedance control Hogan (1985) to achieve a reaction degree required to avoid collisions with unexpected objects. Unfortunately, this extension is unsuitable for cluttered scenes, since it prevents navigation on narrow passages. The use of Lyapunov functions has been also considered in Tanner *et al.* (2001). In that work a motion planner and nonholonomic controller for a mobile robot is presented. Global collision avoidance and convergence properties are stated.

The control of nonholonomic systems has received a lot of attention during last decades. For instance, Kolmanovsky and McClamroch (1995) present a comprehensive survey of nonholonomic control problems. In brief, stabilization of nonholonomic systems is a difficult problem, since there is no smooth (or even continuous) time invariant static state feedback law that can stabilize such systems (Brockett, 1983). In spite of that, it has been shown that with a suitable choice of the state variables, the determination of smooth time-invariant stabilizing feedback laws for a system is feasible (e.g., Astolfi, 1994; Aicardi *et al.*, 1995; Aguiar *et al.*, 2000). The nonholonomic robot navigation problem has been also tackled in Belkhouche (2007), which proposes an approach based on linear navigation functions. Main drawbacks of this approach lie on the methodology used for collision avoidance, which mainly takes into account the geometry of the scene by defining a new set of intermediary goals; no clues about the way of defining these intermediary goals are given.

Having in mind the aforementioned factors, in the current work we propose a new compact autonomous navigation strategy as an improvement of the DWA. This new technique, based on the velocity space (Simmons, 1996; Fox *et al.*, 1997) evaluates non-holonomic constraints, proposes an oriented to the goal, safe and efficient navigation. The incorporation of Lyapunov stability criteria, inside the kernel of DWA, is the novelty of our proposed approach. Hence, a simple and self-contained approach is obtained. Additional contributions are also proposed as follows: *i*) common problems of DWA related to the discrepancy between the robots orientation and the goal are solved (see (7)); *ii*) the proposed improvement permits to take into account angular closeness due to translational velocity (see (8)).

The paper is organized as follow. Next section briefly describes the DWA, as well as its mathematical formalism. Section III presents the proposed navigation technique: I-DWA. Then, experimental results with both simulated and real environments and comparisons with DWA are presented; at the same time, additional constraints required for extending the I-DWA to deal with differential drive robots are also introduced in Section IV. Finally, conclusions and future works are given in Section V.

II. DYNAMIC WINDOW APPROACH

As mentioned above, the proposed technique is based on the *Dynamic Window Approach* (DWA). Therefore, in this section a summary of DWA is given; more details about it can be found in Fox *et al.* (1997).

The main distinctive feature of DWA is based on the fact that control commands (v, w) are directly selected in the velocity space. This space is bounded by constrains directly affecting the robot's behavior; some of those constrains are imposed by the configuration of the obstacles in the environment, while others constrains are from physic limitations of the robot (maximum velocity and acceleration).

It is assumed that the robot moves with a constant velocity (v, w) during each control loop (e.g., Orqueda *et al.*, 2000; Arras *et al.*, 2002; Brock and Khatib, 1999). In other words, assuming a null acceleration the robot only moves with *circular trajectories*—with a constant curvature $c = w/v$ (see illustration in Fig. 1).

Obstacles near to the robot impose constraints over translational and rotational velocities—they will be referred to as *admissible velocities*. The maximum admissible velocity, over a given curvature, depends on the distance to the next obstacle over that curvature. The set of admissible velocities (V_a) is computed by means of a function $Dist\,(v, w)$ that evaluates the distance to the nearest obstacle for a given curvature. This can be expressed as:

$$Dist\,(v, w) = \min_{obs \in OBS} dist\,(v, w, obs)\,, \qquad (1)$$

where *obs* is an element from the set of obstacles OBS. The set V_a can be expressed as:

$$V_a = \left\{ (v, w) | \begin{array}{l} v \leq \sqrt{2 \cdot Dist\,(v, w) \cdot \dot{v}_{max}}, \\ w \leq \sqrt{2 \cdot Dist\,(v, w) \cdot \frac{\dot{w}_{max}}{c}} \end{array} \right\}, \quad (2)$$

where, $\dot{v}_{max}$ and $\dot{w}_{max}$ are the maximum translational and rotational accelerations respectively.

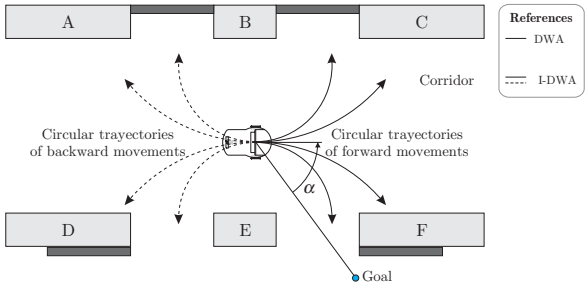

Figure 1: Illustration of a robot navigation environment.

On the other hand, a set of velocities, referred to as *reachable velocities*, indicates those velocities that the robot can achieve during a control loop—velocities defining a *dynamic window* V_d. The set of V_d is expressed as:

$$V_d = \left\{ (v,w) | \begin{array}{c} \frac{v-v_c}{\Delta t} \in [-\dot{v}_{max},\, \dot{v}_{max}]\,, \\ \frac{w-w_c}{\Delta t} \in [-\dot{w}_{max},\, \dot{w}_{max}] \end{array} \right\}, \quad (3)$$

where, v_c and w_c are the current translational and rotational velocities, and Δt is the duration of the control loop. Summarizing, the *search space* of the control commands is reduced to three kinds of constraints: (i) circular trajectories, (ii) admissible velocities, and (iii) reachable velocities. From the constraints imposed over the robot's velocities, a *resulting search space* (V_r) can be defined as:

$$V_r = V_p \cap V_a \cap V_d, \quad (4)$$

where V_p represents the whole space of *possible velocities* for the robot. It is defined by:

$$V_p = \left\{ (v,w) | \begin{array}{c} v \in [0, v_{max}]\,, \\ w \in [-w_{max}, w_{max}] \end{array} \right\}, \quad (5)$$

note that v, in the originally proposed DWA, is only defined for positive values, which means that the robot cannot move backward.

Figure 1 illustrates an example where the robot first goes through a corridor, then it has to cross a door in order to finally reach a user-defined goal. Figure 2 presents the velocities involved in the computation of control actions for the situation presented in Fig. 1.

Finally, from the resulting search space (V_r), DWA selects the couple of velocities that maximize an objective function; different functions have been proposed in the literature (Orqueda *et al.*, 2000; Arras *et al.*, 2002; Brock and Khatib, 1999; among others). The objective function includes terms that trade-off driving the robot at a high speed, oriented to the goal and far away from obstacles: *speed, goal-directedness* and *safety*. Therefore, the objective function is defined as:

$$G(v,w) = \mu_1 \cdot Speed(v) + \mu_2 \cdot Goal(w) + \mu_3 \cdot Dist(v,w)\,, \quad (6)$$

where, $\mu_i > 0$, $i = 1, 2, 3$, and $\sum_i \mu_i = 1$ are weighting factors for each one of those terms. The *Speed* function is used to enforce a navigation with a high speed.

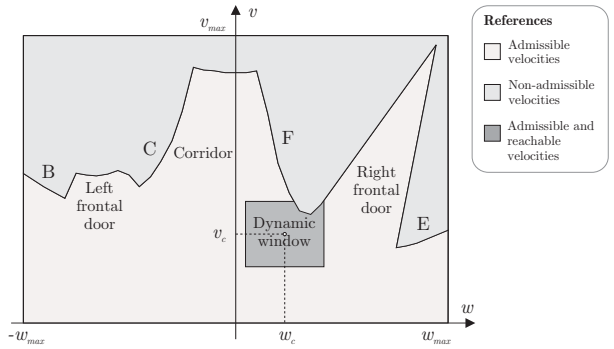

Figure 2: Velocity map corresponding to the illustration presented in Fig. 1.

It is defined as:

$$Speed(v) = v / v_{max}. \quad (7)$$

Nevertheless, the use of this function could suppose taking a wrong action under some particular conditions. For example, it happens when the robot's orientation has a high discrepancy with the goal ($\alpha > 90°$); in this case the robot will move at a high speed away from the goal. On the other hand, the *Speed* function does not take into account the closeness to the goal, therefore when the robot is near the goal this function will promote a wrong action: fast navigation.

The *Goal* function measures the alignment of the robot orientation with respect to the goal, defined with the parameter (α) (Fig. 3). It computes an orientation error, assuming that the robot moves with a constant velocity w during the interval of time of the control loop Δt. This function could be defined as (e.g., Simmons, 1996; Schlegel, 1998; Brock and Khatib, 1999; Arras *et al.*, 2002)

$$Goal(w) = 1 - |\alpha - w \cdot \Delta t| / \pi. \quad (8)$$

Note that it does not include the angular closeness due to translational velocity; this drawback is emphasized in those cases where the robot is near the goal but with a wrong orientation (high α).

The *Dist* function, presented in (1), represents the distance to the nearest obstacle over a circular trajectory with a curvature given by the velocities (v, w).

References Simmons (1996) and Ko and Simmons (1998) represent each term of the objective function as a piece-wise linear function, where the maximum value of (6) is computed by traditional linear programming methods. On the contrary, in Fox *et al.* (1997), Arras *et al.* (2002) and Brock and Khatib (1999); non-linear functions are adopted and the maximum value is computed by looking for it in a discrete space. Alternatively, in Schlegel (1998) and Philippsen and Siegwart (2003), the selective use of a precalculated lookup table is proposed; this allows to control any-shaped robot contours, as well as a fast search for the maximum.

It has been proved that methods based on control command space are appropriate to implement navigation strategies (e.g., Shimoda *et al.*, 2005; Jensen *et*

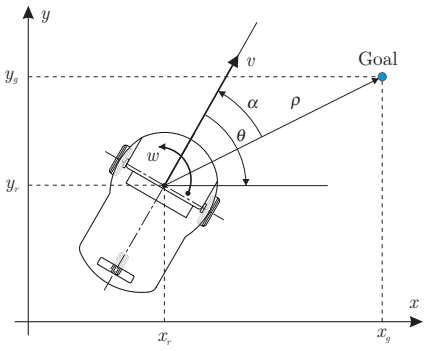

Figure 3: Robot's representation in Cartesian and polar coordinate systems.

al., 2005; Ögren and Leonard, 2005; Bruce and Veloso, 2006). They include the environment constrains (2) and the robot dynamics constrains, (3) and (5). Additionally, they state an objective function that impose suitable behaviors: speed, goal-directedness and safety. However, other convergence criteria need also to be considered in order to evaluate the arrival to the goal.

III. PROPOSED METHOD (I-DWA)

In this section the proposed control strategy, which guarantees and characterizes the arrival to the goal, is presented. In addition, further improvements to the original DWA are introduced to avoid the inconveniences found in the terms of the objective function that impose a fast (7) and oriented to the goal (8) navigation. Next, kinematics equations used to model the robot's motion are introduced. Then, the proposed control law based on Lyapunov stability criteria is presented. Finally, a new objective function is given.

A. Kinematics Equation

Assuming the robot is represented by a point, its kinematics equations, in a Cartesian space, can be expressed as:

$$\begin{aligned} \dot{x} &= v \cdot \cos(\theta), \\ \dot{y} &= v \cdot \sin(\theta), \\ \dot{\theta} &= w, \end{aligned} \tag{9}$$

where θ defines the robot's orientation according to a global reference frame (Fig. 3). This set of equations can also be expressed in a polar reference frame associated with the goal (Casalino *et al.*, 1994; Aicardi *et al.*, 1995; Belkhouche, 2007):

$$\begin{aligned} \dot{\rho} &= -v \cdot \cos(\alpha), \\ \dot{\alpha} &= -w + v \cdot \frac{\sin(\alpha)}{\rho}, \\ \dot{\theta} &= -\dot{\alpha}. \end{aligned} \tag{10}$$

Although the robot has been represented as a point, this model can be extended to different kinds of robots, for instance synchronous drive robots or differential drive (see Section IV). In these cases, a direct implementation could be to represent the robot rotation center as the reference together with the minimum circle bounding the robot.

B. Convergence analysis

This section presents an *ideal control law* (v_i, w_i) that allows driving the robot to the goal guaranteeing a global convergence. In order to do that we propose a candidate Lyapunov law involving two state variables (ρ, α) in a polar reference frame:

$$V(\rho, \alpha) = V_1 + V_2 = \frac{1}{2}\rho^2 + \frac{1}{2}\alpha^2. \tag{11}$$

The time derivation of (11), over the trajectories defined by the set of kinematics Eqs. (10), is expressed as:

$$\begin{aligned} \dot{V}(\rho, \alpha) &= \dot{V}_1 + \dot{V}_2 = \dot{\rho} \cdot \rho + \dot{\alpha} \cdot \alpha = \\ &= -v_i \cdot \cos(\alpha) \cdot \rho + \left(-w_i + v_i \cdot \frac{\sin(\alpha)}{\rho} \right) \cdot \alpha. \end{aligned} \tag{12}$$

The sought convergence is reached by using a control law where the terms of $\dot{V}(\rho, \alpha)$ are always negative defined. Additionally, both velocity values should not be bigger than the maximum values. Hence, we propose a modification in the term related to a fast navigation (7) of the objective function:

$$v_i := k_v \cdot v_{max} \cdot \cos(\alpha) \cdot \tanh(\rho / k_\rho), \tag{13}$$

where, the function $\tanh(\rho/k_\rho) \to 1$ if $\rho \to \infty$, therefore a limit for the translation velocity is defined by v_{max}; k_ρ is a weighting factor that works when the robot approaches to the goal, smoothing its speed reduction. Consequently, the selected velocity will increase according to the value of ρ, but it will be asymptotically bounded. An interesting point of the proposed function is that $\cos(\alpha)$ permits to consider the robot orientation according to the goal. Thus, the selected speed will be high when the robot orients to goal. At the same time, $\cos(\alpha)$ allows even backward movements ($\alpha > 90$). A more elaborated objective function could be envisaged to avoid switching problems on the velocity, by adding some form of hysteresis in the control law—noisy data could induce a positive/negative oscillation from the $\cos(\alpha)$ term. In none of our experiments we found such a problem (see Section IV).

Otherwise, the rotation velocity selection must evaluate the relative closeness originated by the translation velocity. In that sense, the following law is defined for selecting the correct rotation velocity:

$$\begin{aligned} w_i &:= k_\alpha \cdot \alpha + v_i \cdot \frac{\sin(\alpha)}{\rho} = \\ &= k_\alpha \cdot \alpha + k_v \cdot v_{max} \cdot \frac{\tanh(\rho/k_\rho)}{\rho} \cdot \frac{\sin(2 \cdot \alpha)}{2}, \end{aligned} \tag{14}$$

where, $\frac{\tanh(\rho/k_\rho)}{\rho} \rightarrow 1/k_\rho$ if $\rho \rightarrow 0$, therefore the rotation velocity is bounded; k_α and k_v are positive weighting factors intended for obtaining the required robot behavior: k_α works over the angular error whereas k_v works over the distance error. From Eq. (14), the following relationship between these factors and the maximum robot's velocities is obtained: $k_\alpha \leq (|w_{max}| - |k_v \cdot v_{max}/(2 \cdot k_\rho)|)/\pi$.

The next expression is obtained after including the control laws (13) and (14) in (12):

$$\dot{V}(\rho,\alpha) = -\rho \cdot v_{max} \cdot \tanh(k_1 \cdot \rho) \cdot \cos^2(\alpha) - k_\alpha \cdot \alpha^2,$$
$$\dot{V}(\rho,\alpha) \leq 0. \quad (15)$$

Hence, the proposed Lyapunov function (11) is always non-incremental in time, as we were looking for. Additionally, this kind of function guarantees a global and asymptotical convergence to the goal; numerically it can be expressed as:

$$\dot{V}(\rho,\alpha) = \dot{V}_1 + \dot{V}_2 \Rightarrow \left\{ \begin{array}{l} \rho(t) \\ \alpha(t) \end{array} \right. \rightarrow 0, \text{ if } t \rightarrow \infty . \quad (16)$$

This function is only valid when there are not obstacles in the environment and when the robot does not present acceleration limitations. The latter condition appears in the initial point when the robot needs to start the motion and reach the velocities stated in (13) and (14). These velocities take into account only the robot's position and orientation discrepancies with the given goal (ρ, α). Therefore, additional constrains, to avoid choosing unreachable velocities, should be imposed. Unreachable velocities are due to physical limitations in the robot's accelerations; in addition velocities that involve some collision risk should be also avoided. Next, both drawbacks are considered.

C. Navigation function

The proposed navigation function used to drive the robot to the goal, by avoiding collisions and taking into account robot dynamics constraints (maximum accelerations and velocities), is presented in this section.

Dynamics constraints imposed by the robot and obstacles are considered by using the search space defined in (4). As in previous works, a dynamic window, which contains the current set of reachable and admissible velocities, is computed. Then, an improved objective function $G_m(v,w)$ is proposed to replace (6). The domain of that objective function is defined by a dynamic window and contains the next three terms:

$$G_m(v,w) = \begin{array}{l} \lambda_1 \cdot (1 - |v - v_i| / (2 \cdot v_{max})) + \\ \lambda_2 \cdot (1 - |w - w_i| / (2 \cdot w_{max})) + \\ \lambda_3 \cdot Dist(v,w), \end{array} \quad (17)$$

where, $\lambda_i > 0$, $i = 1,2,3$, and $\sum_i \lambda_i = 1$, represent weighting factors. The variables v_i and w_i are defined by the proposed (13) and (14) and they are the responsible for driving efficiently the robot to the goal.

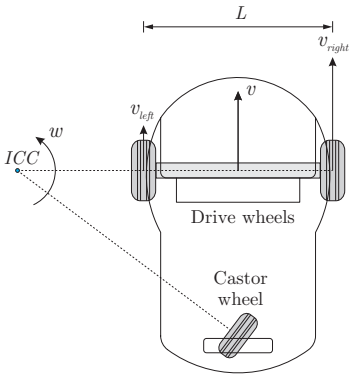

Figure 4: Kinematics characteristics of Pioneer 1 (differential drive robot).

The chosen set of velocities are those that maximize the objective function (17).

The first two terms of (17) favor choosing velocities that drive the robot to the goal, the third term implements the collision avoidance strategy. Therefore, the values adopted by the weighting factors will be reflected in the robot behavior. High λ_1 and λ_2 values will result in a goal oriented behavior, while a high λ_3 value will result in a highly reactive behavior that favor the collision avoidance part. Actually, the tuning of these parameters is defined by characteristics such as: the environment structure (number and distribution of obstacles), the required robot behavior (level of reaction), and the degree of knowledge of the environment.

IV. EXPERIMENTAL RESULTS

This section presents experimental results obtained with both, simulated environments and real environments. In both cases a PIONEER 1 robot was used (see Fig. 4). It presents a maximum translation velocity of 600 mm/s, a maximum rotation velocity of about $2.5\,rad/s$, and a distance between drive wheels of 325 mm. Before going into details about the obtained experimental results the required extension of I-DWA to tackle differential drive robots is presented.

A. Differential drive robots

This section presents an extension of the I-DWA algorithm in order to be able to tackle differential drive locomotion problems. Kinematics equations, together with the corresponding transformations in the velocity space, to handle this kind of robots are also introduced.

A differential drive robot uses a simple locomotion system composed of two drive wheels and a passive rear wheel. Drive wheels are independently controlled while the passive rear wheel is only used as an additional leaning point to keep the robot's balance. The rear wheel is automatically oriented according to the robot motion. The robot's displacement is achieved by means of a separated control of each drive wheel. A pure translation is reached when both drive wheels have the same velocity—magnitude and direction—,

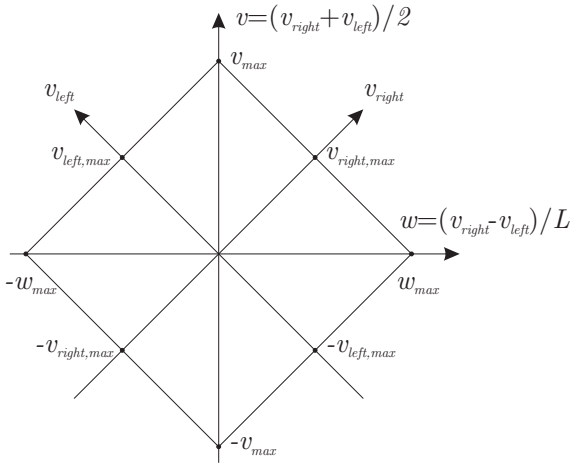

Figure 5: Velocity map of a differential drive locomotion robot.

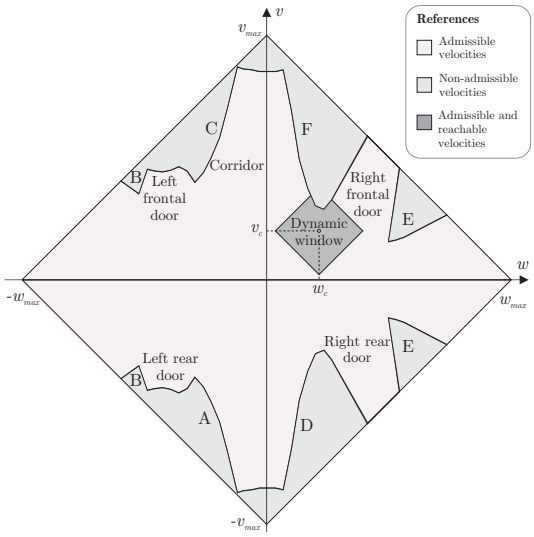

Figure 6: Velocity map corresponding to the illustration presented in Fig. 1.

while a pure rotation appears when the velocities of drive wheels are the same in magnitude but with an opposed direction. An instantaneous center of curvature (ICC), defined by the intersection of the drive wheels axis with the passive wheel axis, is automatically defined according to the robot's displacement (see Fig. 4).

Kinematics equations define the interaction between control commands and the corresponding space state. Thus, in a differential drive locomotion robot, these equations will reflect the robot's position (x, y, θ) when the velocity of each drive wheels is controlled (v_{right}, v_{left}). Therefore, from (9), translational and rotational velocities can be expressed by means of the drive wheel velocities:

$$\begin{aligned} v &= \frac{v_{right} + v_{left}}{2}, \\ w &= \frac{v_{right} - v_{left}}{L}, \end{aligned} \tag{18}$$

where L is the length of the drive wheels axis (Fig. 4). A clockwise displacement ($w > 0$) is performed when $v_{right} > v_{left}$, otherwise a counter-clockwise displacement will be executed. From (18) the kinematics equations of a differential drive locomotion robot can be expressed as:

$$\begin{aligned} \dot{x} &= \frac{v_{right} + v_{left}}{2} \cdot \cos(\theta), \\ \dot{y} &= \frac{v_{right} + v_{left}}{2} \cdot \sin(\theta), \\ \dot{\theta} &= \frac{v_{right} - v_{left}}{L}. \end{aligned} \tag{19}$$

Equation (18) can be also represented by means of a matrix as:

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 \\ 1/L & -1/L \end{bmatrix} \cdot \begin{bmatrix} v_{right} \\ v_{left} \end{bmatrix} \tag{20}$$

hence, its inverse representation can be easily expressed as:

$$\begin{bmatrix} v_{right} \\ v_{left} \end{bmatrix} = \begin{bmatrix} 1 & L/2 \\ 1 & -L/2 \end{bmatrix} \cdot \begin{bmatrix} v \\ w \end{bmatrix} \tag{21}$$

The Eqs. (20) and (21) allow the representation of the robot's velocities (assuming the robot is represented by a point) together with the velocities of drive wheels (see Fig. 5). After adding admissible and reachable velocities a representation such as the one presented in Fig. 6 is obtained. This representation is used to define the corresponding dynamic window and select the appropriate control action for the next control loop (17). Experimental results with a differential drive robot are presented in the next section.

B. Simulated environments

Simulated environments were used to test the proposed technique. Additionally, simulated environments are useful to study the robustness of the technique and the performance of the robot when different values are used for tuning the parameters. Figure 7 shows one of the proposed scenarios to evaluate the robot's behavior in front of different situations. Goals are indicated with flags while the robot's trajectories through that scenario are illustrated by means of small icons. Note that the robot is plotted by regular intervals of one second, which allows to infer the robot velocities. Each obstacle has associated two concentric circles. The inner circle corresponds to an obstacle enlargement according to the robot's radius (minimum bounding circle of the robot). The outer circle represents the influence limit; in other words, when the robot reaches the area defined by that outer circle it should start with the collision avoidance strategy. The set of trajectories performed by the robot to reach each one of the proposed goals shows the efficiency of the proposed approach. Note that I-DWA correctly drives

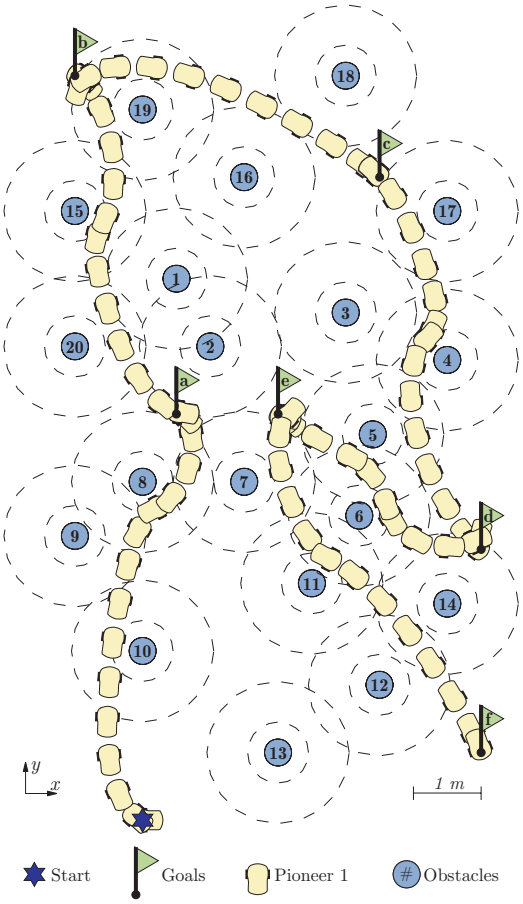

Figure 7: Trajectories in a simulated environment performed during the test of the proposed approach.

the robot along narrow passages, such as the <e> trajectory that goes through the 5 and 6 obstacles. High speeds were reached even in presence of obstacles.

Table 1 shows the average speed reached by the robot in every trajectory. Note that in all the cases the average translation velocity value is higher than half the maximum speed. Moreover, that maximum speed is also reached in every trajectory (see Fig. 8).

Figure 8 shows a plot of the velocities reached by the robot in its way to the goal. This figure demonstrates that the movements are smooth with few oscillations on the robot's orientation. It is also observable that the robot first tries to orientate towards the goal, in some cases by moving backward, and then, it advances trying to reach the highest speed in absence of obstacles. This behavior is regulated with the factors k_ρ, k_α and k_v from ideal control law, (13) and (14); as well as the weighting factors of the objective function $(\lambda_1, \lambda_2, \lambda_3)$, (17). In the current implementation they were defined as: $k_\rho = r_r^{-1} \simeq 3\,[m^{-1}]$, $k_\alpha = 0.59\,[s^{-1}]$, $k_v = 1$, $\lambda_1 = 3/13$, $\lambda_2 = 3/13$ and $\lambda_3 = 7/13$.

These values were experimentally selected by using test-bed environments. As mentioned above, since a simulated environment is used, several experiments can be performed keeping away from risky situations. In addition, simulated environments make it easier and

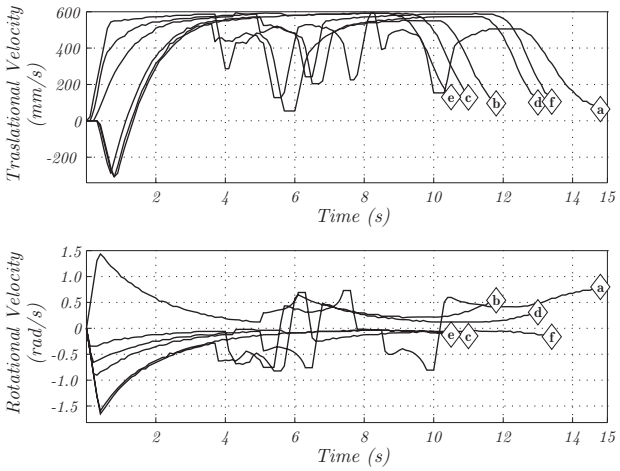

Figure 8: Velocities corresponding to the trajectories presented in Fig. 7.

faster the parameters tuning. The used strategy is as follow. Firstly, a free-obstacles environment is considered until the sought robot behavior is obtained; i.e., efficient and successful movements to the goal. Finally, a fine tuning is performed by using a highly populated environment, where a safe collision avoidance is reached.

C. Comparisons

The proposed I-DWA has been compared with the original DWA (Section II.). Comparisons were performed by using a differential drive robot where it can be easily appreciated the convergence drawbacks of DWA for arriving to the goal. A simple test, with a free-obstacle environment, shows the inability of DWA for regulating the robot behavior to reach the goal.

Independently of the values used for tuning (μ_1 and μ_2), only two different trajectories are obtained. These trajectories are the results of selecting $(\mu_1/\mu_2) > k_\mu$ or $(\mu_1/\mu_2) < k_\mu$ (in our case $k_\mu = 0.5$), μ_3 is not considered since a free-obstacle scene is used. The first trajectory is obtained when more weight is given to the *Speed* term in (6); trajectory <g> in Fig. 9, note that the goal is not reached. On the contrary, the second trajectory is obtained when more weight is given to the *Goal* term, trajectory <h> in Fig. 9; here, the robot reaches the goal, but firstly it oscillates until the right

Table 1: Speed and time for each trajectory.

Trajectory	Average Speed	Time
$< a >$	442.95 $[mm/s]$	14.8 $[s]$
$< b >$	466.30 $[mm/s]$	11.8 $[s]$
$< c >$	450.85 $[mm/s]$	11.0 $[s]$
$< d >$	463.00 $[mm/s]$	13.0 $[s]$
$< e >$	370.60 $[mm/s]$	10.5 $[s]$
$< f >$	453.75 $[mm/s]$	13.4 $[s]$

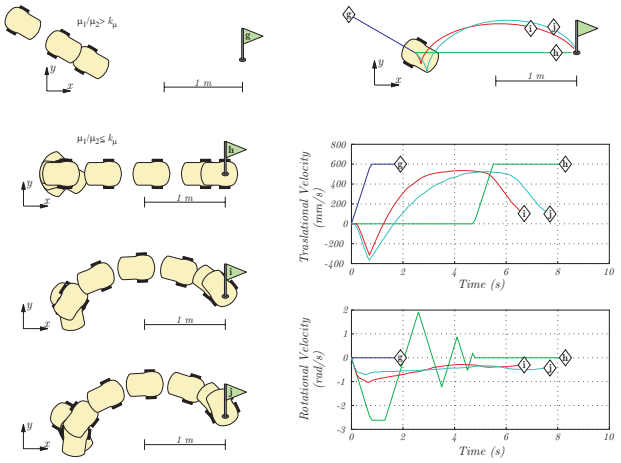

Figure 9: Comparisons between DWA (trajectories $< g >$ and $< h >$) and the proposed I-DWA (trajectories $<i>$ and $<j>$).

orientation is obtained and then it moves to the goal.

On the contrary, with the proposed I-DWA different trajectories can be performed by tuning (13), (14) and (17) as required. Figure 9 shows some of these trajectories: $<i>$ and $<j>$. Note that in these cases, smooth translational and rotational velocities are simultaneously developed, giving rise to more efficient trajectories; no careful tuning of the user-defined parameters has been done, the presented results are only illustrative. Moreover, it can be appreciated that I-DWA allows a faster arrive to the goal.

D. Real environments

Finally, the proposed technique was validated in real environments (about 4x5 m) with a PIONEER 1 robot. The I-DWA parameters were tuned with the same values than the selected in the simulated environments. I-DWA was implemented off-board, in the framework Aria/Saphira (Konolige, 1998; Laffary, 2002)), using a master-slave architecture.

Figure 10 presents a scene with six unknown obstacles and a sequence of five user-defined static goals. Obstacles and robot are represented as in the simulated environment (Fig. 7). Note that this scene is considerably smaller than the one presented in Fig. 7 since robot localization is only based on odometry. As mentioned above, the main objectives of using real environments are: on the one hand, to shown the similarity between these results and those obtained in simulated environments; on the other hand, to validate the proposed I-DWA by showing both the arrival to the goal and collision avoidance performances. As it was expected, in all the cases the robot reaches the goals with smooth and safe trajectories. Figure 11 depicts the velocities corresponding to every trajectory. Equally than in the simulated scenario, efficient movements, with few oscillations on the robot's orientation, are performed. Note that the robot stops when it reaches a goal; in some cases the robot needs

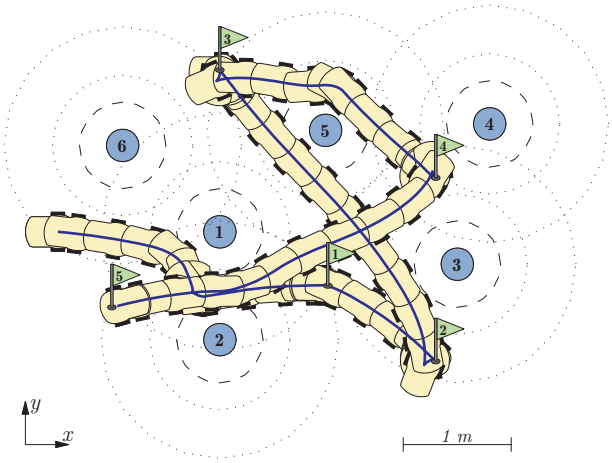

Figure 10: Real navigation environment used for testing I-DWA with the PIONEER 1.

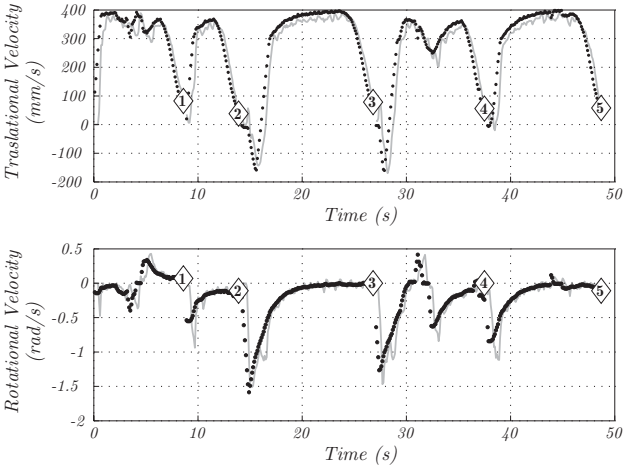

Figure 11: Velocities corresponding to the trajectories presented in Fig. 10.

to go backward in order to quickly orientate to the next goal. The robot maximum speed has been reduced up to 400 mm/s to avoid localization problems related to the odometry.

V. CONCLUSIONS

This paper presents a novel and compact approach (I-DWA) for autonomous robot navigation. It improves the original technique by incorporating Lyapunov stability criteria inside the kernel of DWA. Therefore, an arrival to the goal with a global and asymptotic convergence is guaranteed. As a result, a more simple and self-contained approach is obtained. Different robot behavior can be reached by tuning a set of control law parameters.

Simulated and real experimental results validate the proposed technique when different scenarios are used. Additionally, comparisons with the original DWA, by using a differential drive robot, are presented. It shows that with the proposed approach the conver-

gence drawbacks of DWA are solved. At the same time, the capability of I-DWA to regulate different behaviors can be easily appreciated.

ACKNOWLEDGEMENTS

This work has been partially supported by: the Argentinian *"Agencia Nacional de Promoción Científica y Tecnológica"* (project PICT 2003 Code 13468), *"Universidad Nacional del Sur"* (project 24/K083), the Spanish *"Ministry of Education and Science"* (project TRA2007-62526/AUT and research programme Consolider Ingenio 2010: MIPRCV-CSD2007-00018). The second author was supported by The Ramón y Cajal Program.

REFERENCES

Aguiar, A.P., A.N. Atassi and A.M. Pascoal, "Regulation of a Nonholonomic Dynamic Wheeled Mobile Robot with Parametric Modeling Uncertainty using Lyapunov Functions," *IEEE Conference on Decision and Control,* 2995-3000 (2000).

Aicardi, M., G. Casalino, A. Bicchi and A. Balestrino, "Closed Loop Steering of Unicicle-like Vehicles via Lyapunov Techniques," *IEEE Robotics and Automation Magazine,* **2,** 27-35 (1995).

Arras, K.O., J. Persson, N. Tomatis and R. Siegwart, "Real-Time Obstacle Avoidance For Polygonal Robots With A Reduced Dynamic Window," *IEEE International Conference on Robotics and Automation,* 3050-3055 (2002).

Astolfi, A., "On the stabilization of non-holonomic systems," *IEEE Conference on Decision and Control,* 3481-3486 (1994).

Belkhouche, F., "Nonholonomic Robot Navigation Using Linear Navigation Functions," *American Control Conference,* 5328-5332 (2007).

Borenstein, J. and Y. Koren, "Real-time Obstacle Avoidance for Fast Mobile Robots," *IEEE Transactions on Systems, Man, and Cybernetics,* **19,** 1179-1187 (1989).

Borenstein, J. and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation,* **7,** 278-288 (1991).

Brock, O. and O. Khatib, "High-Speed Navigation Using the Global Dynamic Window Approach," *IEEE International Conference on Robotics and Automation,* 341-346 (1999).

Brockett, R.W., "Asymptotic stability and feedback stabilization," *Differential Geometric Control Theory*, Edts. R.W. Brockett, R.S. Milman and H.J. Sussmann, Birkhauser, Boston, 181–191 (1983).

Bruce, J.R. and M.M. Veloso, "Safe multirobot navigation within dynamics constraints," *Proceedings of the IEEE*, **94**, 1398-1411 (2006).

Casalino, G., M. Aicardi, A. Bicchi and A. Balestrino, "Closed-loop steering for unicycle-like vehicles: A simple lyapunov like approach," *IFAC Symposium on Robot Control*, 291-298 (1994).

Fox, D., W. Burgard and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, **4**, 23-33 (1997).

Hogan, N., "Impedance control: an approach to manipulation," *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, **107**, 1-24 (1985).

Jensen, B., N. Tomatis, L. Mayor, A. Drygajlo and R. Siegwart, "Robots meet Humans-interaction in public spaces," *IEEE Transactions on Industrial Electronics,* **52**, 1530-1546 (2005).

Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research,* **5**, 90-98 (1986).

Khatib, M. and R. Chatila, "An extended potential field approach for mobile robot sensor-based motions," *Intelligent Autonomous Systems (IAS'04)*, 490-496 (1985).

Ko, N.Y. and R.G. Simmons, "The Lane-Curvature Method for Local Obstacle Avoidance," *IEEE/RSJ International Conference on Intelligent Robots and Systems,* 1615-1621 (1998).

Kolmanovsky, I. and N.H. McClamroch, "Developments in nonholonomic control problems," *IEEE Control Systems Magazine*, **15**, 20-36 (1995).

Konolige, K., *Saphira Software Manual*, SRI International(1998).

Laffary, M., *ARIA Reference Manual*, ActivMedia Robotics LLC (2002).

Latombe, J.-C., *Robot Motion Planning*, Kluwer Academic Publishers (1991).

Ögren, P. and N.E. Leonard, "A Convergent Dynamic Window Approach to Obstacle Avoidance," *IEEE Transactions on Robotics and Automation,* **21**, 188-195 (2005).

Orqueda, O.A.A., H. Berti, O.E. Agamennoni, "A Strategy for Safe Goal-Directed Autonomous Navigation," *4th IFAC International Symposium on Intelligent Components and Instruments for Control Applications,* 135-140 (2000).

Philippsen, R. and R. Siegwart, "Smooth and Efficient Obstacle Avoidance for a Tour Guide Robot," *IEEE International Conference on Robotics and Automation,* 446-451 (2003).

Schlegel, C., "Fast Local Obstacle Avoidance under Kinematic and Dynamic Constraints for a Mobile Robot," *IEEE/RSJ International Conference on Intelligent Robots and Systems,* 594-599 (1998).

Secchi, H., R. Carelli and V. Mut, "Design of Stable Algorithms for Mobile Robot Control with Obstacle Avoidance," *14th IFAC World Congress on Automatic Control,* 185-190 (1999).

Shimoda, S., Y. Kuroda and K. Iagnemma, "Potential Field Navigation of High Speed Unmanned Ground Vehicles on Uneven Terrain," *IEEE International Conference on Robotics and Automation,* 2839-2844 (2005).

Simmons, R., "The Curvature-Velocity Method for Local Obstacle Avoidance," *IEEE International Conference on Robotics and Automation,* 3375-3382 (1996).

Tanner, H.G., S. Loizou and K.J. Kyriakopoulos, "Nonholonomic Stabilization with Collision Avoidance for Mobile Robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1220-1225 (2001).

Received: December 2, 2006.
Accepted: November 28, 2007.
Recommended by Subject Editor: Julio Braslavsky