NEARNESS DIAGRAM NAVIGATION (ND):

A NEW REAL TIME COLLISION AVOIDANCE APPROACH

J. Minguez L. Montano
Computer Science and Systems Engineering
Universidad de Zaragoza (SPAIN)
e-mail : {jminguez@robot3.cps,montano@posta}.unizar.es

Abstract

This paper presents a new real-time collision avoidance approach for mobile robots. The Nearness Diagram method (ND) performs a high level information extraction and interpretation of the environment. Subsequently, this information is used to generate the motion commands. The proposed approach is well-suited to deal with unknown, unstructured and dynamic environments, where problems of other approaches are avoided. Some experimental results are shown using an holonomic mobile base to demonstrate the usefulness of the method.

1 INTRODUCTION

The robot movement points towards the most important task in a typical indoor-outdoor mission (without it other tasks can not be done), which determine drastically the global success of a robotic mission.

The algorithms that generate motion for mobile robots can be classified into planning algorithms and real-time collision avoidance algorithms. Planning algorithms consider a model of the environment (either previously known or dynamically built), to compute a collision free path between the current robot location and the goal. On the other hand, the collision avoidance algorithms generate directly, from sensory information, the motion commands that drive the robot towards the final location. The use of these real-time collision avoidance approaches is widely justified when, under certain circumstances (unknown and dynamic environments), the trajectories generated by the planning algorithms become inaccurate and replanning is continuously required to reach the goal. Since the planning algorithms can be too time consuming to avoid collisions in real time, motion commands are generated in an efficiently way by real-time collision avoidance approaches.

The main problem of these collision avoidance approaches is that they use a local fraction of the information available, so it is impossible to generate optimal solutions and thus they can fall in trap situations.

In this paper a new collision avoidance approach is presented, which differs from other methods, in the way to deal with the environmental information available and to find the motion commands. This framework has been applied to unknown, unstructured and dynamic environments, where the results show that some problems of others approaches are avoidable, including when navigating in very complex and dense environments.

The paper is organized as follows. After discussing the related work, Section 3 explains the information extraction and interpretation and Section 4 introduces the navigation strategy. Section 5 shows some results obtained with an holonomic mobile base. Section 6 and 7 discuss the contributions of the approach with respect to other methods, and the conclusions.

2 RELATED WORK

The movement of a robot between locations, avoiding collisions in real time, has been a subject widely studied by other researchers. We propose two ways to classify the collision avoidance methods. They can be sorted into different groups depending on the quantity of analysis and interpretation carried out on the environmental information available. On the other hand, they can be grouped depending on the number of heuristics used to find the motion commands.

Groups of information analysis and interpretation

The environmental analysis and interpretation, is a step of the local collision avoidance algorithms where some information is obtained. This information can be presented in different ways (sets of feasible trajectories, set of motion commands, sets of free space, ...). Subsequently, there is a navigation strategy which uses this information to generate the motion commands.

In the first information analysis and interpretation group, there are the methods that directly calculate the motion commands from the environmental information available. They usually make a physical analogy. They directly apply to the information acquired, some mathematical equations, whose solutions can then be easily transformed into motion commands: [1], [2], [3], [4], [5], [6], [7], [8].

In the second information analysis and interpretation group, there are methods that obtain, from the information acquired, some sets of motion commands. After this step, there is always a navigation strategy, which selects one element of these sets of motion commands. We separate the methods in two branches, the methods that give a set of steering angles ([9], [10], [11], [12]), and the methods that give a set of velocity commands ([13], [14], [15]).

In the third information analysis and interpretation group, we would like to find some methods that obtain, from the information acquired, some form of high level description about the environment. This will allow the next

This work was partially supported by spanish CICYT project TAP97-0992-C02-01, Caja de Ahorros de la Inmaculada (Proyecto Europa ref: CB15/99), and Departamento de Educación y Cultura de la Diputación General de Aragón (Ref P29/98).

step to calculate the motion commands, but not to select them from a precalculated set. We have found in this group the Elastic Band approaches ([16],[6]). They use the concept of bubble to describe parts of the free space. Subsequently, they are used to use, and modify, a collision free path to generate the motion commands. The ND proposed in this paper is placed in this group. It extracts from the information available, a description of regions which are free of obstacles. It chooses one of these regions, and it evaluates the robot security. Next, it selects one of the five situations which will be discussed later on. This information is subsequently used to generate the motion commands.

Groups of navigation heuristics

On the other hand, from a navigational point of view, the collision avoidance approaches can be divided by the number of motion heuristics they use. The motion heuristics are the laws used to calculate the motion commands, from a set of possible motion commands or from other types of information.

In the first group, the methods that only use one heuristic to generate the motion commands are found. There are some methods that calculate the motion commands directly from one mathematical equation [1], [2], [3], [4], [5], [6], [7], [8], [16]. Other methods choose one solution from a precalculated set, with a criteria that usually is a cost function ([9], [12]), or a constraint optimization ([13] [14], [15]).

In the other group, the methods that use more than one heuristic to generate the motion commands are found. Here we have found [10] and [11]. They change among three different motion selection heuristics, due to the set of motion commands precalculated and the goal direction. The ND is placed in this group because it uses five laws to calculate the motion commands, which are adapted to the five general situations of the robot, and so on to the information extracted in the previous step.

The contributions of the ND regarding other approaches will be further discussed in Section 6.

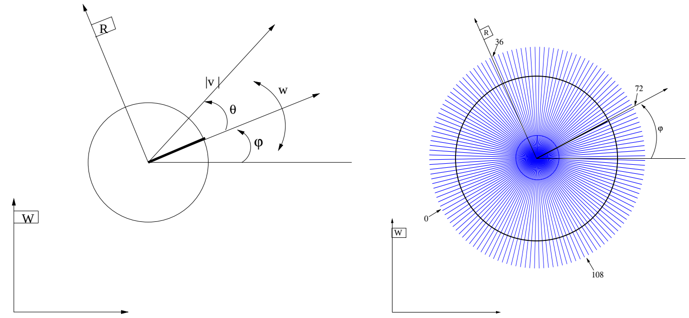
3 ENVIRONMENTAL INFORMATION EXTRACTION AND INTERPRETATION

The proposed navigation method carries out the environmental information extraction in three steps. Firstly, from the information available, two nearness diagrams are constructed (the PND and the RND). Secondly, the PND is analysed to identify regions and to select one of them. Thirdly, the RND is analysed to evaluate the robot safety situation. Subsequently, this information is used to identify one of the five general situations.

3.1 DIAGRAMS DEFINITION

The ND uses a sectored representation of the environment (see Fig. 1). From it, two diagrams are constructed, the PND (Nearness Diagram from the central Point) and the RND (Nearness diagram from the Robot), to represent information about the nearness of the obstacles. Some PND and RND diagrams can be seen in Fig. 2.

To define the diagrams, we adopt the notation introduced in [17].



(a) (b)

Fig. 1. a) Robot reference and motion commands ($\mathbf{v}$,$\mathbf{w}$). b) Sectors distribution in the robot reference.

Let $\mathbf{b} \in A$ the point of the robot which is used as the center of sectors, which is placed in the center of the robot, and called *central point*. Let n be the number of sectors (n=144 in this work, 2.5° being the angle of each sector), and φ the robot orientation in the global reference. Let δ be a function, that when given a robot configuration $\mathbf{q}$ and a point $\mathbf{b}$, calculates a vector such that $\delta_i(\mathbf{q},\mathbf{b})$ is the smallest distance to an obstacle in the sector i, φ being the angle corresponding to the bisector of the $\frac{n}{2}$ sector (see Fig. 1b).

Definition 1: **Nearness Diagram from the central Point (PND)**

$$\begin{array}{llll} PND: & C_{free} \times A & \rightarrow & (\mathbb{R}^+ \cup \{0\})^n \\ & (\mathbf{q},\mathbf{b}) & \rightarrow & (D_1,\cdots,D_n) \end{array}$$

$$\begin{array}{ll} \text{if } \delta_i(\mathbf{q},\mathbf{b}) > 0, & D_i = PND_i(\mathbf{q},\mathbf{b}) = d_{max} + l - \delta_i(\mathbf{q},\mathbf{b}) \\ \text{otherwise} & D_i = PND_i(\mathbf{q},\mathbf{b}) = 0 \end{array}$$

where:

- d_{max}: maximum value of δ, representing the maximum range of the sensor used.
- l: maximum distance between two points of the robot (being the diameter for a circular robot).

Definition 2: **Nearness Diagram from the Robot (RND)**

$$\begin{array}{llll} RND: & C_{free} \times A & \rightarrow & (\mathbb{R}^+ \cup \{0\})^n \\ & (\mathbf{q},\mathbf{b}) & \rightarrow & (D_1,\cdots,D_n) \end{array}$$

$$\begin{array}{ll} \text{if } \delta_i(\mathbf{q},\mathbf{b}) > 0, & D_i = RND_i(\mathbf{q},\mathbf{b}) = d_{max} + E_i(\mathbf{b}) - \delta_i(\mathbf{q},\mathbf{b}) \\ \text{otherwise} & D_i = RND_i(\mathbf{q},\mathbf{b}) = 0 \end{array}$$

where

- E: is an enlarging function that depends on the robot geometry. The value of this function in each sector $E_i(\mathbf{b})$ is the robot radius for a circular robot.

The PND represents the nearness of the obstacles to the central point, and the RND represents the nearness of the obstacles to the robot boundary (see Figs. 2).

Let $PND_i \equiv PND_i(\mathbf{q},\mathbf{b})$ and $RND_i \equiv RND_i(\mathbf{q},\mathbf{b})$.

3.2 PND ANALYSIS

The PND represents the nearness of the obstacles to the central point. The topology of the environment does not vary in this diagram, so it is used to extract information of environmental characteristics. The PND analysis is performed in three stages. Firstly, gaps in the environment are searched for. From these gaps, regions of the free space are obtained, and finally one of them is chosen by one criteria (see Fig. 2a).

Discontinuity

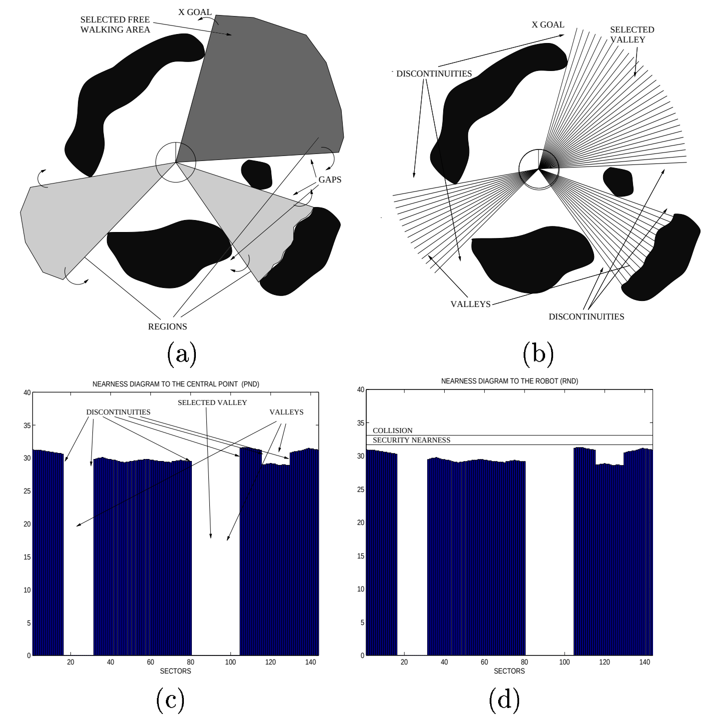


Fig. 2. a) Gaps, regions and the selected free walking area. b) Discontinuities, valleys and the selected one mapped on the environment. c) PND generated. d) RND generated.

Between two adjacent sectors there is a discontinuity if their height difference in the PND is greater than l (introduced in Definition 1). See Fig. 2c.

Discontinuities represent gaps among obstacles or gaps due to the end of an obstacle.

Valley

A set of sectors $S = \{s_i\}_{i=1,\cdots,k}$ constitutes a valley (with s_l and s_r the left and right *extreme sectors*, respectively), if they satisfy the following two premises:

1. All the sectors in S are adjacent and there are no discontinuities among them.
2. Let s_{nl} and s_{nr} be the adjacent sectors to s_l and s_r respectively, not belonging to S (named *adjacent sectors* to the valley). There are two discontinuities between (s_l,s_{nl}) and (s_r,s_{nr}) and both sectors satisfy:

$$PND_{s_{nl}} - PND_{s_l} > l \text{ or } PND_{s_{nr}} - PND_{s_r} > l$$

In [19] a formal definition of valley can be found. Fig. 2b shows an example of the valleys mapped on the environment.

A special case occurs when the robot has the goal location between an obstacle and itself; the sector that holds the goal location (s_{goal}) can not belong to a valley. When this situation is detected, $PND_{s_{goal}}$ to zero is set, creating an artificial valley in the goal sector.

There are two types of discontinuities (gaps) depending on the premise 2, the rising and the descending discontinuity. A discussion of it is out of the scope of this paper, and the reader is directed to [19].

There are *wide* and *narrow* valleys. A valley is *wide* if its number of sectors is greater than s_{max} (a parameter, in our current implementations $s_{max} = \frac{n}{2}$), and *narrow* otherwise.

Valleys represent regions among obstacles.

Selected Valley

The criteria to choose a valley is to select the one which has the rising discontinuity, with minimal sectorial distance from the sector that holds the goal location (s_{goal}). Depending on the information source and the geometry of the robot, some strategies can be proposed to determine, if it is possible to reach the gap of the selected free walking area. At this point we have developed an algorithm that works for circular robots. Let us name the region represented by the selected valley, the selected free walking area (see Fig. 2).

3.3 RND ANALYSIS

The analysis of the RND evaluates the robot safety situation. Some concepts must be introduced beforehand:

Definition 3: **Security distance:** Minimum tolerable distance (d_s) to an obstacle (starting from it, the robot is not safe).

Definition 4: **Security nearness:** Maximum tolerable nearness (n_s) to an obstacle. It is the value in the RND, starting from it, the robot is too close to an obstacle, calculated as: $n_s = d_{max} - d_s$ (see Fig. 2d).

Safety Situations

There are two safety situations in which the robot can be: Low Safety (LS) and High Safety (HS).

The robot is in Low Safety situation (LS) if, in the RND, at least one sector exceeds the security nearness. It means that the distance between the boundary of the robot and the obstacle, is lower than the security distance, this being a potential risk for the robot (something is inside of the security zone, Fig. 3). Otherwise, the situation is High Safety (HS), Figs. 4 and 5.

3.4 GENERAL SITUATIONS

At this point, the environmental analysis has obtained some regions of the environment and a description of the robot safety. This information is now used to define the five general situations for the robot. They cover all the possibilities among the robot location, goal location and obstacle configuration, but they have to be checked in rigorous order.

There are two LS situations:

1. **LS1**: The robot is in Low Safety 1, if there is at least one sector that exceeds the security nearness in the RND, only on one side of the rising discontinuity, which is closer to the goal sector, of the selected valley. This situation happens when the robot is too close to the obstacles, only on a side of the gap, which is closer to the goal, of the selected free walking area (see Fig. 3a).
2. **LS2**: The robot is in Low Safety 2, if there is at least one sector that exceeds the security nearness in the RND, on both sides of the rising discontinuity, which is closer to the goal sector, of the selected valley. This situation occurs when the robot is too close to the obstacles, on both sides of the gap, which is closer to the goal, of the selected free walking area (see Fig. 3b).

Note that both LS situations depend on the relative locations among the robot, obstacles and the free walking area selected (the goal location is only used to select the free walking area).

There are three HS situations:

3. **HSGV**: The robot is in High Safety Goal in Valley if the goal sector belongs to the selected valley. This situation occurs when the goal location is inside of the selected free walking area (see Fig. 4).
4. **HSWV**: The robot is in High Safety Wide Valley if the selected valley is wide. This situation happens when the goal is not inside of the free walking area selected, but it is wide (see Fig. 5a).
5. **HSNV**: The robot is in High Safety Narrow Valley if the selected valley is narrow. This situation occurs when the goal is not inside of the free walking area selected, but it is narrow (see Fig. 5b).

Note that the HS situations depend only on the goal location, with respect to the selected free walking area or its shape.

4 NAVIGATION STRATEGY

The navigation strategy is based on two important assumptions: One, it is very difficult to solve the problem of local navigation with a unique motion heuristic, due to the structural complexity that can present an environment (even in a local fashion). Two, the direct use of the goal location in a motion heuristic has a pernicious effect (except in an obvious situation without apparent complexity).

The ND uses five laws of motion, adapted to the five situations obtained in the analysis and interpretation step, in which the goal location is directly used only in one of them. The ND calculates the translational and rotational velocity (**v**,**w**) in each sample time as motion commands for an holonomic mobile robot (see Fig. 1a).

4.1 TRANSLATIONAL VELOCITY

The translation velocity is divided in the local frame of the robot in module and direction **v**=(v,θ). We next show how they are calculated.

A.1 TRANSLATIONAL VELOCITY DIRECTION

For each situation a solution sector $s_\theta \in \mathbb{R}$ is calculated. The direction of motion θ is obtained as the bisector of s_θ. Due to the fact that $s_\theta \in \mathbb{R}$, infinite virtual sectors are created around the robot that can be chosen as solution, so any direction of motion can be assigned, being $\theta \in [-\pi, \pi]$.

For a realistic implementation of the method, it is desirable to fix $\theta \in [-\pi/2, \pi/2]$, that is $s_\theta \in [\frac{n}{4}, \frac{3*n}{4}]$ to prohibit instantaneous backwards motion.

Navigation in Low Safety Situations

In LS the robot is in danger of colliding, so the solution has to put the robot in a secure situation.

The navigation in each LS situation is next shown:

1. **LS1 Navigation**: In LS1 there are obstacles closer to the security distance, only on one side of gap, which is closer to the goal, of the selected free walking area. The solution has to produce a motion that brings the obstacles out from the security zone, while moving towards the selected gap (see Fig. 3a). The solution sector is calculated by:

$$s_p = Abs(s_i - s_j) * p + \frac{s_{max}}{2}$$

$$s_\theta = s_i + sign(s_i - s_j) * s_p$$

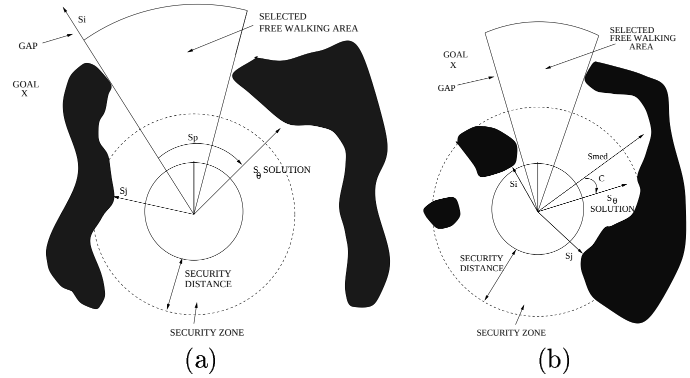


Fig. 3. a) LS1 situation example. b) LS2 situation example.

where:

- s_i: sector corresponding to the rising discontiuity (gap) of the selected valley (free walking area).
- s_j: sector with the highest value in the RND, that exceeds the security nearness, on a side of the rising discontinuity of the selected valley. It corresponds to the closer obstacle.
- p: experimentally tuned parameter, whose value depends on the transitions among the general safety situations, and assure a smooth behaviour among them. In our current implementations $p \in [1.5, 2.5]$.

In LS1 Navigation, the main objective is to push the robot away from the closest obstacle, while moving towards the gap of the free walking area.

2. **LS2 Navigation**: In LS2 there are obstacles closer to the security distance on boths sides of the gap, which is closer to the goal, of the selected free walking area. The solution has to produce a motion to centre the robot between both obstacles, while moving towards the selected gap (see Fig. 3b). The solution sector is computed as:

$$s_{med} = \frac{s_i + s_j}{2}$$

$$s_\theta = s_{med} \pm c$$

where:

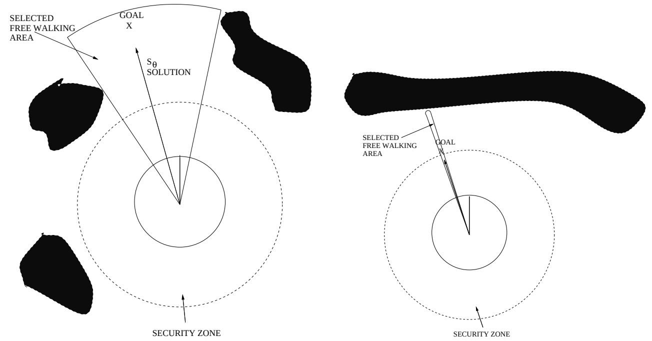
- s_i, s_j: are the sectors with highest values in the RND, that exceed the security nearness, on both sides of the rising discontinuity of the selected valley. They correspond to the two closer obstacles.
- c: is a correction value used to centre the robot between the two closer obstacles. It depends on the closer obstacle distance, and on the distance difference of the two closer obstacles. This quantity c is added or subtracted in function of the sector that holds the closer obstacle, to put the robot at the same distance of the two obstacles (most safe situation).

In LS2 Navigation, the main objective is to maintain the robot at the same distance from the two closer obstacles, while moving towards the selected gap.

Navigation in High Safety Situations

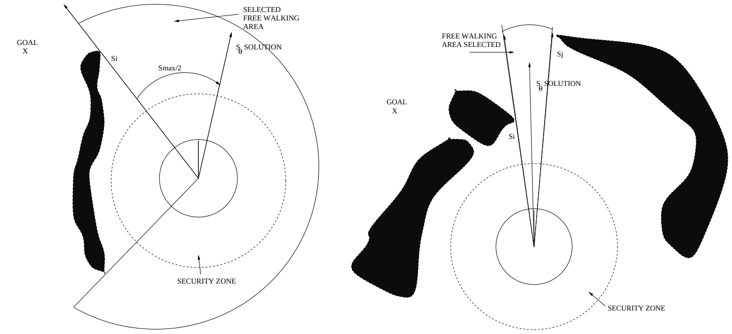
In High Safety the robot is not in danger of colliding, so the solutions are chosen inside of the free walking area selected.

The navigation in each HS situation is shown next:



(a) (b)

Fig. 4. a) HSGV situation example. b) HSGV situation example (artificial valley case).



(a) (b)

Fig. 5. a) HSWV situation example. b)HSNV situation example.

3. **HSGV Navigation**: In HSGV the goal location is inside of the free walking area. The solution is to move the robot towards the goal (see Fig. 4). The solution sector is calculated by $s_\theta = s_{goal}$.
This is the situation that uses directly the goal location to calculate the motion commands. Note that in this situation the robot is not in danger of colliding, and the goal is inside of a free walking area (that is a situation not dangerous for the robot and without apparent complexity).
In Fig. 4 two examples of the HSGV situation are shown. In Fig. 4a the robot moves directly towards the goal location (that is inside of the selected free walking area). Fig. 4b is the special case where an artificial valley is created.
This navigation drives the robot directly to the goal.
4. **HSWV Navigation**: In HSWV the goal sector is not inside of the free walking area selected, but it is wide. The solution is to have a *contour following* behavior (see Fig. 5a). The solution sector is calculated by:

$$s_\theta \quad = s_i \pm \frac{s_{max}}{2}$$

where:
- s_i: sector corresponding to the rising discontinuity, which is closer to the goal sector, of the selected valley. It corresponds to the gap, closer to the goal, of the selected free walking area.

This navigation gives as result a motion along side the obstacle
5. **HSNV Navigation**: In HSNV the goal sector is not inside of the free walking area selected, but it is narrow. The solution is to move towards the center of the free walking area (see Fig. 5b). The solution sector is calculated by:

$$s_\theta \quad = \quad \frac{s_i + s_j}{2}$$

where:
- s_i, s_j: are the sectors of the two discontinuities of the selected valley. It corresponds to the two gaps of the free walking area

This navigation directs the robot among the obstacles.

A.2 VELOCITY MODULE

The velocity module is calculated depending on the safety situation of the robot. Let v_{max} be the maximum translational velocity. Let d_{obs} be the distance to the closer obstacle to the robot bounds, and d_s the security distance. Let be $\theta \in [-\pi/2, \pi/2]$ the direction of the translational velocity calculated.
1. High Safety: $v = v_{max} * Abs(1 - \theta/\frac{\pi}{2})$
2. Low Safety: $v = v_{max} * \frac{d_{obs}}{d_s} * Abs(1 - \theta/\frac{\pi}{2})$

With this velocity control, the robot moves at maximum speed until one obstacle enters in the security zone, then the robot reduces its speed proportionally to the closer obstacle distance, until the security zone is cleared. Moreover, hard changes in direction reduce the translational velocity module, while the robot turns towards the instantaneous motion direction.

4.2 ROTATIONAL VELOCITY

The rotational velocity is calculated from the direction of the translational velocity. It is desirable that the robot is aligned with the instantaneous motion direction. Let be w_{max} the maximum rotational velocity. Let be $\theta \in [-\pi/2, \pi/2]$ the direction of the translational velocity calculated.

$$\mathbf{w} \quad = w_{max} * \theta/\frac{\pi}{2}$$

This produces hard turns of the robot, when sudden changes in the angle of the translational velocity (the robot faces as soon as possible the motion direction), and smooth turns when the changes are small.

5 EXPERIMENTAL RESULTS

The Nearness Diagram has been implemented and tested on the Nomad XR4000 shown in Fig. 6a. This base moves at omnidirectional translational velocities of up to $1.2\frac{m}{sec}$, and accelerations of up to $1.5\frac{m}{sec^2}$. It is equipped with a SICK laser rangefinder with a field of view of 180°, a range of 32 meters, and an accuracy of up to $1cm$.

The algorithm takes $10msec$ to work in each sample time for one laser measure (corresponding to one lap). In our current implementation we are using the last 20 laser measures (corrected to the actual robot location, used as a short-term memory), that gives a time algorithm of $125msec$ (because of the sectorization process, and the correction of the last measures), that is well-suited to have a real time approach.

Next three experiments are shown (see Figs. 6b,c,d). They have been chosen to show how, problems of other

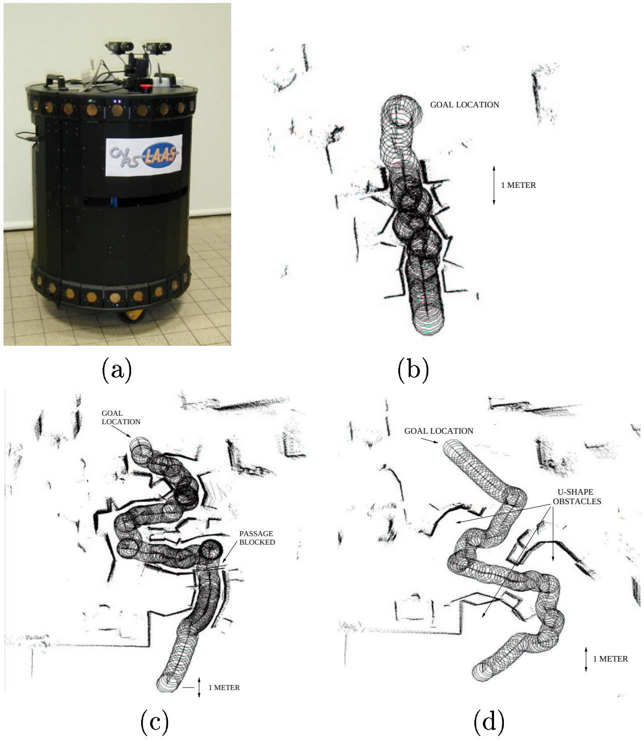


Fig. 6. a) Nomad XR4000. b) Experiment 1. c) Experiment 2. d) Experiment 3.

approaches, are avoided by the ND. This will be discussed in the next section.

The maximum translational velocity set for the robot was $v_{max} = 0.5\frac{m}{sec}$, and the maximum rotational one was $w_{max} = 1.57\frac{rad}{sec}$. In all the experiments only the goal location was known a priori for the robot.

- **Experiment 1**: The robot had to cross a passage with hard asymmetries and a very reduced place to manoeuvre. Directions towards close obstacles were chosen every time to find the right way. No oscillations appeared throughout the whole run.
- **Experiment 2**: In this experiment the environment was dynamically built. In the first part of it, when the robot was in the first corridor, we closed the passage. The robot detected that it was blocked and it stopped (it was the best thing that could be done working with a local method). Then, the passage was opened and the robot continued its motion. In the whole travel, the robot was obliged to choose directions of motion far from the goal location (which demonstrates the insensibility to the goal location), and directions towards obstacles. It had also to react to the new obstacles we were placing.
- **Experiment 3**: The robot had to travel to reach the goal avoiding three typical U-shaped structures placed in the environment. The robot did not enter and get trapped because the U-shape structures were completely visible by the robot. That demonstrates the fact that, having a high level structure (regions) to reason and calculate the motion commands, enables trap situations to be avoided with the information available at the current moment.

6 COMPARISON WITH OTHER METHODS

A discussion of the improvements that the method does, against other collision avoidance approaches, is next presented. We have chosen the Potential Field Methods (in a general fashion), the Vector Field Histogram [10], the Elastic Band [16] and the Dynamic Window Approach [14]. These methods cover all the groups discussed in Section 2.

POTENTIAL FIELD METHODS (PFM)

In [18] were presented the inherent problems of this methods, which are taken as the basis for the following discussion:

It is a well-known problem of the PFM to get easily trapped into U-shape structures. The ND only introduces the robot in a U-shape structure if it is partially visible, and the partially visible part holds the goal direction. If the U-shape structure is completely visible, the ND avoids this situation in an elegant way, because no regions (valleys) appear on the inside of the U-shape structure (see experiment 3).

Very dense or asymetric environments can produce unstable motion or trap situations when using a PFM. The ND does not present problems on introducing, and driving the robot, among very close obstacles (if it is possible, that is checked), see experiment 1. It is also very insensitive to displacements in the goal location, whilst moving among very close obstacles, because the goal location is not directly used to calculate the motion commands in this situation (see experiment 2). Moreover, the ND does not present oscillations in very narrow corridors, because the motion generation in this case is oriented to avoid them (see experiment 1).

In a general way, we can conclude that the ND does not suffer from the inherent problems of the PFM, except the U-shape obstacles (where a very insensitive behavior is found). With the experience that we have in our laboratory with the PFM [8], we have not been able to find an environment, where our PFM implementations succeed and the ND fails.

VECTOR FIELD HISTOGRAM (VFH)

The VFH [10] is an obstacle avoidance method, that selects the motion direction from a precalculated set of solutions (valleys), switching among three different laws. This set of solutions is obtained with an empirical threshold. Its tuning should be different when navigating from very dense environments (as shown in experiment 1), to less dense. A fixed tuning should produce that some open areas could be lost (valleys), or on the other hand, some obstacles could not be tacked into account. In the ND, the regions (which are not used as sets of solutions), are obtained using a robot parameter (the diameter for a circular one), and none of these are lost, if they exist.

The VFH does not take into account the width of the robot. A low pass filter is used, to empirically compensate the robot's width, and to smooth the polar histogram. However, even with a well-tuned filter, the robot could have tendency to cut corners. The ND takes directly into account the width of the robot, and filters are not used. Moreover, we have not observed it cutting corners because of the designed control laws (see experiment 2, 3).

In [11] the VFH+ was presented. It solves most of the problems of the VFH, but it can not direct the robot towards an obstacle, as it was possible in the original VFH method. This constraint makes the method not well-suited for very dense environments (where directions towards the obstacles are required every moment). The ND has the ability to select directions towards the obstacles when it is necessary (see experiment 1, 2).

ELASTIC BAND (EB)

The elastic band [16], [6] is a method that needs a collision free path to the goal, elaborated by a motion planner. Subsequently, subjected to artificial forces, the elastic band deforms in real time to a short and smooth path, that maintains clearance from obstacles.

The band is well-suited to deal with known or partially known environments, where an approximate trajectory can be calculated. On the other hand, its behaviour is not good with unknown and very dynamic environments. The problem in these environments is that the band is deformed, but always connected. This preserves the topology of the trajectory, but it easily gets blocked when U-shape or unexpected obstacles appear, or when moving obstacles removes the band far from the goal. The ND reacts well to dynamic and unknown environments, where it searches for places to move among the obstacles (see experiment 2 and 3, where we were unable to reach the goal with our Elastic Band implementation [6]).

DYNAMIC WINDOW APPROACH (DWA)

The dynamic window approach [14], [15] is a collision avoidance method where, the search for motion commands (translational and rotational velocity of the robot), is carried out directly in the space of velocities. The motion commands are selected from a maximisation of an objective function. This objective function balances the goal heading, the forward robot velocities and the obstacle clearance. From our point of view this can be a problem under certain circumstances, because the solution depends on the weigthing coefficients heuristically chosen. Moreover, this way to find the motion commands avoids any interpretation of the precalculated set of solutions, and therefore of the predictability of a solution. The navigation strategy of the ND is implemented in a geometrically way, so the solutions obtained are completly predictable.

Another difficulty we have found, in most of the collision avoidance approaches (specially in the PFM and the EB), is the tuning of the internal parameters. It seems complicated to find the best values, to have a good reaction, to the most of the collision avoidance situations. The ND only has one parameter heuristically chosen (p parameter), being easy to find a good value that does not determine the final solution, and which is only used in one of the five navigation laws.

7 CONCLUSIONS

The ND is a new collision avoidance approach, that uses some high level description of the environmental information, to generate the motion commands, with one among five different laws.

The method is well-suited to navigate in unknown, unstructured and dynamic environments. The real strength of the method is when dealing with very dense and complex environments (that is the case in a typical populated indoor environments), where problems of other approaches are avoided.

Further work is directed to extend the ND to nonholonomic robots, and to robots with shapes.

ACKNOWLEDGEMENT

We thank R. Alami and T. Simeon of LAAS/CNRS (FRANCE) for their valuable comments and discussions. Moreover, we thank S. Fleury of LAAS/CNRS (FRANCE), for her help in the algorithm implementation on the XR4000 Nomad platform .

References

[1] O.Khatib "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots" *The International Journal of Robotics Research*, 5(1), 1986.

[2] B.H.Krough "A Generalised Potential Field Approach to Obstacle Avoidance Control" *Robotics Research: The Five Years and Beyond. SME Conference Proceedings* Bethlelem, 1984.

[3] R.B. Tilove "Local Obstacle Avoidance for Mobile Robots based on the method of Artificial Potentials" *Research Publication GMR* 6650, 1989.

[4] K. Azarm, G. Schmidt "Integrated mobile robot motion planning and execution in changing indoor environments" *Proceedings 1994 IEEE/RSJ/GI Int. Conference on Intelligent Robots and Systems* Vol. 1, pp. 298-305, Munchen, Germany, 1994.

[5] A. Masoud, S Masoud, M. Bayoumi "Robot Navigation using a Pressure Generated Mechanical Stress Field, The Biharmonical Potential Approach" *IEEE International Conference on Robotics and Automation*, pp.124-127. California, USA, 1994.

[6] Maher Khatib "Sensor-based motion control for mobile robots" *PHD thesis, LAAS-CNRS* December, 1996.

[7] J.Borenstein Y.Koren "Real-time Obstacle Avoidance for Fast Mobile Robots" *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 5, Sept./Oct., pp.1179-1187.

[8] L. Montano and J.R.Asensio "Real-Time Robot Navigation in Unstructured Environments Using a 3D Laser Rangefinder" *IEEE-RSJ International Conference on Intelligent Robots and Systems, IROS'97*, pp526. Grenoble, France, 1997.

[9] W.Feiten R.Bauer G.Lawitzky "Robust Obstacle Avoidance in Unknown and Cramped Environments" *IEEE International Conference on Robotics and Automation*, pp2412, 1996.

[10] J.Borenstein and Y.Koren "The Vector Field Histogram (VFH)-Fast Obstacle Avoidance for Mobile Robots" *IEEE Journal of Robotics and Automation*, 7(3), 1991.

[11] I.Ulrich and J.Borenstein "VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots" *IEEE International Conference on Robotics and Automation*, pp1572, Leuven, Belgium, 1998.

[12] M.Hebert C.Thorpe A.Stentz "Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon", *Kluwer Academic Publishers* 1997.

[13] R.Simmons "The Curvature-Velocity Method for Local Obstacle Avoidance" *IEEE International Conference on Robotics and Automation*, pp3375, Mineapolis, USA, 1996.

[14] D.Fox W.Burgard S.Thurn "The Dynamic Window Approach to Collision Avoidance" *IEEE Transactions on Robotics and Automation*, 4:1, 1997.

[15] O.Brock O.Khatib "High-Speed Navigation Using the Global Dynamic Window Approach" *IEEE International Conference on Robotics and Automation*, pp341, Michigan, USA, 1999.

[16] S.Quinlan O.Khatib "Elastic Bands: Connecting Path Planning and Control", *IEEE International Conference on Robotics and Automation*, Vol 2, pp. 802-807, Atlanta, USA, 1993.

[17] J.C.Latombe "Robot Motion Planning" *Kluwer Academic Plublishers*, 1991.

[18] Y.Koren J.Borenstein "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation" *IEEE International Conference on Robotics and Automation*, pp. 1398-1404, California, USA, 1991.

[19] J.Minguez L.Montano "Nearness Diagram Navigation (ND): A New Real Time Collision Avoidance Approach for Holonomic and no Holonomic Mobile Robots" *Internal Report RR-00-14 University of Zaragoza*,pp 60, February 2000.