GLOBAL NEARNESS DIAGRAM NAVIGATION (GND)

Proceedings of the 2001 IEEE
International Conference on Robotics & Automation
Seoul, Korea • May 21-26, 2001

J.Minguez L.Montano T.Simeon† R.Alami†
Computer Science and Systems Engineering †LAAS (CNRS)
CPS, University of Zaragoza, SPAIN Toulouse, FRANCE
{jminguez,montano}@posta.unizar.es {nic,rachid}@laas.fr

Abstract

This paper presents the Global Nearness Diagram (GND) navigation system for mobile robots. The GND generates motion commands to drive a robot safely between locations, whilst avoiding collisions. This system has all the advantages of using the reactive scheme Nearness Diagram (ND), while having the ability to reason and plan globally (reaching global convergence to the navigation problem). This framework has been extensively tested using an holonomic mobile base equipped with a laser range-finder. Experiments in unknown, unstructured, dynamic and complex environments are reported to validate the system.

1 Introduction

The development of a robust navigation system, which can work in different environments, and can adapt to everyday situations, is still an open research area in the field of robotics.

The construction of environmental models is highly coupled to navigation. This task is dependent on the natural and environmental conditions.

Focusing our attention on motion generation, we can divide navigation systems into three categories [22]: Model-based navigation systems, Hybrid systems and Reactive schemes.

- The *Model-based navigation systems* construct a model of the environment used directly to extract the motion commands. This model is based on the specific characteristics of the world (indoor/outdoor, static/dynamic, structured/unstructured...).
- The *Reactive navigation schemes* are restricted to the iteration between perception (usually the system inputs), and action (usually the system outputs). This constrains their solutions to a local section of the environment, and non optimal solutions are obtained. On the other hand, these reactive schemes have been demonstrated to be extremely well-adapted to very complex and dynamic environments, which model-based navigation systems cannot cope with.
- The *Hybrid systems* integrate both schemes in the sense that each one works independently, but they interact to perform the navigation task.

The difference between reactive systems and hybrid systems is that reactive schemes deal directly with perceptions, in order to generate motion commands, while hybrid systems build a model that interacts with the reactive scheme.

This work was partially supported by projects CICYT TAP97-0992-C02-01, DPI2000-1272 and Departamento de Educación y Cultura de la Diputación General de Aragón (Ref P29/98).

The main difference between model-based systems and hybrid systems, is the motion commands generation process. The former builds a model which is directly used to generate the motion commands. On the other hand, hybrid systems have two very well distinguished tasks: the model builder and the reactive navigation scheme, the latter generating the motion commands.

We focus our attention on reactive schemes, and their evolution over the last years. Early reactive navigation methods, firstly attempted to solve problems related to their internal behavior and drawbacks. Secondly, reactive methods evolved in order to deal with their lack of global reasoning and planning (towards hybrid methods), see Section 2 for an extended discussion on this topic.

In this paper, we present the evolution of the reactive method Nearness Diagram Navigation (ND) [1], towards the Global Nearness Diagram (GND). The GND is a navigation scheme that assures global convergence to the reactive navigation problem, inside the physical limits imposed by a model dynamically built. The GND is shown to be a very powerful navigation system, because it has all of the advantages of the reactive method ND, while incorporating global reasoning, which allows it to avoid trap situations.

The paper is organized as follows: Section 2 presents the related work, and the system requirements is introduced in Section 3. Sections 4 and 5 present two navigation systems (mND and mpND), and Section 6 shows how they cooperate to form the complete navigation system (GND). In Section 7 a comparison with other methods is presented, and in Section 8 we draw our conclusions.

2 Related Work

A reactive navigation scheme (also known as collision avoidance approach), is an algorithm that takes as input perceptions of the environment. The outputs are the motion commands that drive the robot towards the final location, while avoiding collisions. The reader is directed to [1] for an extended discussion and taxonomy of these methods.

The evolution in time followed by the common reactive navigation methods can be divided into two steps. In the first one, the methods evolved to cope with their internal drawbacks and limitations (eliminate oscillations, local trap situations, unstable motion, motion constraints, robot geometry...). In the second step, the methods evolved to deal with their lack of global reasoning, trying to increase its local nature (in the direction of hybrid methods).

The methods as they evolved cannot be considered to be purely reactive, because they go farther than only dealing with perceptions. On the other hand, they cannot be considered exclusively hybrid, because the devices introduced to increase the local nature, do not make complete sense outside of the navigation field, and are completely oriented

to improve the reactive method behavior.

We will discuss the evolution of five techniques (see Fig. 1): Potential Field Methods (PFM [2]), Vector Field Histogram (VFH [8]), Dynamic Window Approach (DWA [13]), Elastic Band (EB [15])) and Nearness Diagram (ND [1]).

Potential Field Methods (PFM)

The PFM [2] are obstacle avoidance methods that make a physical analogy to generate collision free motion. The obstacles and goal generate forces that are respectively repulsive and attractive. The motion commands are computed from these forces. The PFM technique has been widely used and studied by a large number of researchers [3], [4], [5], [17], [23], among others. These methods are in the first step of the evolution, because some inherent limitations settle in [6], are still a subject of research.

Vector Field Histogram (VFH)

The VFH [8] is an obstacle avoidance method, that selects the motion direction from a precalculated set of solutions (valleys), switching among three different laws. Later VFH+ [9] was presented, where internal problems and drawbacks of the original VFH were overcame. VFH+ took into account the width of the robot and the robot trajectory. Less oscillatory results were obtained, and it was possible to commit to a direction due to improved motion selection. Recently, the VFH* [10] was presented which basically deals with the local nature of the VFH+. VFH* uses a look-ahead verification to analyze the consequences of heading towards the candidate directions. The consequences are quantified by cost functions, allowing for the selection of the one which minimize some criteria. Trap situations are avoided by calculating a number of steps in advance of the algorithm's execution.

Elastic Band EB

The Elastic Band [15] and [16] is a framework that provides many of the benefits of reactive systems without sacrificing global planning. A path is provided by a global planner. Incremental adjustments to the path are based on the sensory data while maintaining the path in the free space. The concept of *bubble* is introduced to implement the elastic band efficiently. Later, [17] introduced a new formalism of this concept in a *Reed and Shepp* metric system, taking into account the kinematic constraints of the robot. Recently, the elastic strip (ES) framework has been presented [18] and [19]. Here, several local replanning operations are integrated in this framework to deal with moving obstacles, to improve the behavior in dynamic environments. While the elastic strips can be used to obstacle avoidance for mobile robots, it has been shown to work extremely well in high-dimensional configuration spaces.

Dynamic Window Approach (DWA)

In the mid-90's, some researches made an effort to incorporate vehicle dynamics into the collision avoidance problem, choosing motion commands rather than a travel direction (SAFA [11] and CVM [12]). But it was the DWA [13], the method that won more popularity in the scientific community. The DWA formulates the problem as a constrained optimization in the velocity space. Constraints are derived from physical limitations of the robot's velocities and from the sensor data (that indicates the presence of obstacles). The original DWA was formulated to synchro-drive robots. Recently, the GDWA [14] was presented, where the original DWA was reformulated to holonomic mobile robots,

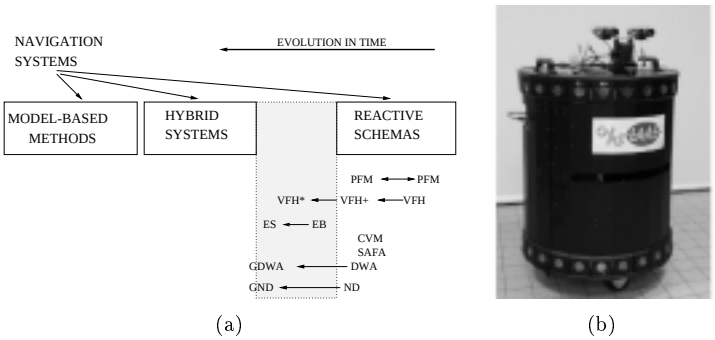

Fig. 1. a) Reactive schemes evolution. b) Nomad XR4000 platform.

and the cost function was slightly modified to improve the robot behavior [18]. Moreover, connectivity of the space was explored, allowing trap situations to be avoided.

Nearness Diagram (ND)

The ND [1] is a reactive scheme that performs a high level information extraction and interpretation of the environment. Subsequently, this information is used to generate the motion commands. In a first step, the ND extracts a description of regions which are free of obstacles, selects one of them, evaluates the robot security and chooses one of the five general situations defined. Secondly, it generates the motion commands with one of the five laws adapted to the general situations. The contribution of the ND scheme among other reactive methods can be seen in [1]. This paper describes the evolution from the ND towards the GND.

3 System Requirements

The aim of this work is to create a navigation system that drives the robot robustly among locations. We have identified three requirements, that have to be accomplished when designing a navigation system that executes motion tasks in an autonomous way:

1. **Information integration**: it is necessarily to integrate information from different perceptions into a model of the environment. Two reasons motivate this: firstly, it gives a framework to have an incremental global reasoning. Secondly, past perceptions may be used to avoid obstacles not perceived at the current moment (sensor constraints).
2. **Dynamic environments reaction**: when the environment changes dynamically, the description of the environment has to model instantaneously this change. If not, the robot will avoid parts of the space known to be free of obstacles, or will not avoid perceived obstacles.
3. **Trap situations solution**: There are a lot of obstacle configurations that produce trap situations. The most typical is the U-shape obstacles and they are common for all the reactive methods. Moreover, there are some symmetric environments where the reactive methods can produce alternate solutions. These environments create cyclic behaviors in the robot motion.

The evolution from the purely reactive navigation system ND, through to the final navigation system GND, has gone a way to accomplish these three requirements.

4 Mapping ND (mND)

In Section 3, three requirements were outlined in order to design an effective navigation system. We now go on to

present the mND. It consists of a navigation system that uses a dynamically built model of the environment, and a reactive scheme to generate the motion commands. With this method we want to fulfill the first two requirements, related to information integration and the dynamic environments reaction.

No assumptions about the environment are made (static/dynamic, structured/unstructured...). The sensor used to perceive the environment is a laser range-finder.

The model of the environment is constructed by merging the information in an occupancy-grid that represents the working space. The laser sensory data is introduced directly into the grid model without any pre-processing, and is updated at each servo tic.

The grid has three types of cells: *occupied*, *free* and *unknown*. A point measured by the laser gives an *occupied* cell in the grid. Lines between the sensor and the measured points are projected to the map as *free* cells. Initially, all the cells of the map are set to *unknown* (never perceived). The Bresenham algorithm [20] is used to optimally update the map, in order to achieve real-time performance.

The occupancy-grid represents a finite subsection of the environment centered around the robot. A local region is defined in the center of the grid. When the robot escapes from this local region, the entire grid moves to encompass the robot within the local region. This allows the robot to move within the local region without having to move the complete grid. Grid displacements are always multiples to the cell's dimension, and rotation is not needed. Thus, error propagation associated to the measures in the cells is avoided.

Once the model has been built, it is used as input by the reactive navigation method, instead of directly using perception (see Fig. 4). For robust navigation, the approach relies on the fact that the robot's surroundings are constantly sensed, and that the map is updated at high rate.

It is important to remark that the last perception introduced in the map has no odometric errors with respect to the robot's location, and only sections not perceived accumulate them. Moreover, spurious measures are eliminated from map while introducing the new perceptions. Assuming that the robot performs instantaneous forward motions (that usually coincides with the main visibility sensor direction), and little slippage occurs during motion, this framework results in a very adequate method of **integrating the information** at different times (always in the obstacle avoidance context).

Moreover, the **the environment's dynamic** is reflected in the model as it is perceived, which is a consequence of updating the entire area covered by the last perception.

4.1 Experimental Results

We have extensively tested this navigation system on the XR4000 platform in LAAS (CNRS) shown in Fig. 1b. This base moves with an 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 range-finder with a field of view of 180°, a range of 32 meters, and an accuracy of up to 3cm.

To perform the experiments, the dimensions of the map are 10 by 10 meters, and cell dimensions are 5 by 5 centimeters, which gives a grid of 200 by 200 cells. The process of updating the map with a laser measure (360 points), and moving the grid when necessary, takes approximately 100ms. The ND takes less than 50ms which gives a cycle-time of 150ms. These times are well suited to real-time collision avoidance. The maximum translational velocity set for the experiments was $v_{max} = 0.5\frac{m}{sec}$, and the maximum rotational one was $w_{max} = 1.57\frac{rad}{sec}$.

Fig. 2a presents a real experiment where a human walked between the robot and the selected passage. In this case, the environment's dynamic has to be automatically introduced because:

- If the human is not automatically integrated into the model, the reactive method will not have time to react.
- If the last of the robot's perceptions of the human are not eliminated from the model, the passage will remain closed, and the reactive method will avoid the free space.

From left to right, in Fig. 2b the robot moves towards the center of the passage. In Figs. 2c,d,e the human appears in the scene. In Fig. 2f the human enters in the security zone and the robot starts an avoidance manoeuvre, while moving towards the passage. In Fig. 2g, the human completely blocks the passage and the robot continues to avoid him. In Fig. 2h, the human has moved passed the passage, which appears now open for the robot to enter. It now turns towards the passage while continuing to avoid the human. In Fig. 2i the human has finally left the security zone, and so the robot recovers its motion towards the center of the passage.

The experiment shows that the human is automatically integrated into the model, so the reactive scheme avoids it instantaneously. Moreover, past human's perceptions are automatically eliminated, and the passage remains open after the person moved passed it. As consequence, the reactive scheme instantaneously directs the robot through the passage.

5 Mapping-Planning ND (mpND)

The mND is a framework which integrates the information in a model of the environment. The reactive scheme generates the motion commands to avoid collisions. Two advantageous properties are extracted from this coupling:

1. The model integrates past perceptions, thus the reactive method is able to avoid obstacles not perceived at the current moment.
2. The model reflects the environment's dynamics when it is perceived, so the reactive method reacts instantaneously to change.

However, due to the lack of global reasoning in the system, it still has limitations when dealing with trap situations.

The mpND is a navigation system that uses the mND scheme, but exploits the information of the connectivity of the space with a planning algorithm. With this method, we want to fulfill the third requirement, as stated in Section 3 (trap situations).

A minima-free navigation function NF1 [21] is built, using a wave propagation technique, over the configuration space calculated from the grid model. Finally, a path free of collisions, that connects the initial and final configurations, is obtained by a gradient-search technique. The main reasons for the use of this planning algorithm are: its grid-based navigation function (adapted to the grid-model); and its simplicity and efficiency, which allows for the computa-

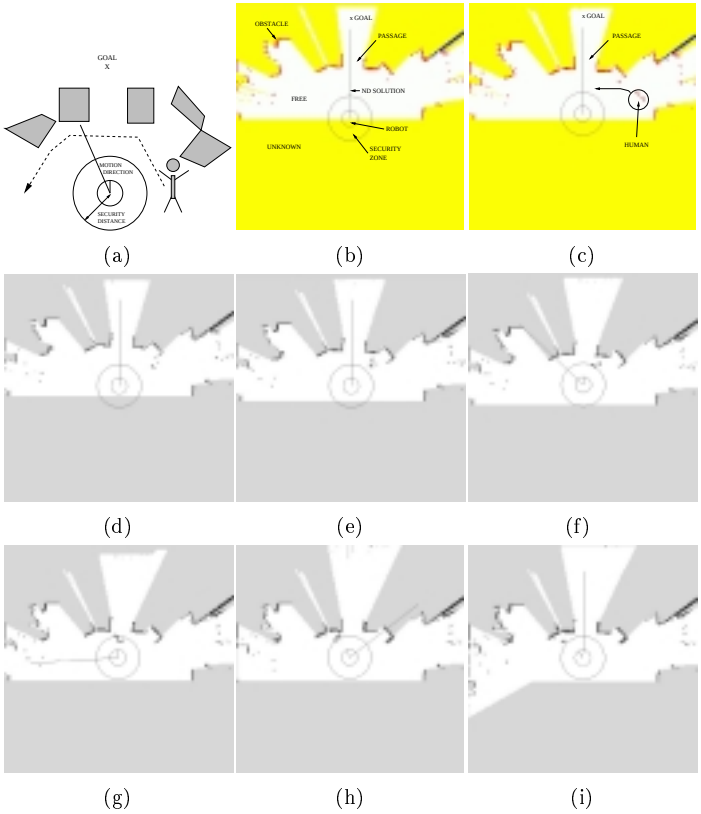

Fig. 2. mND real experiment

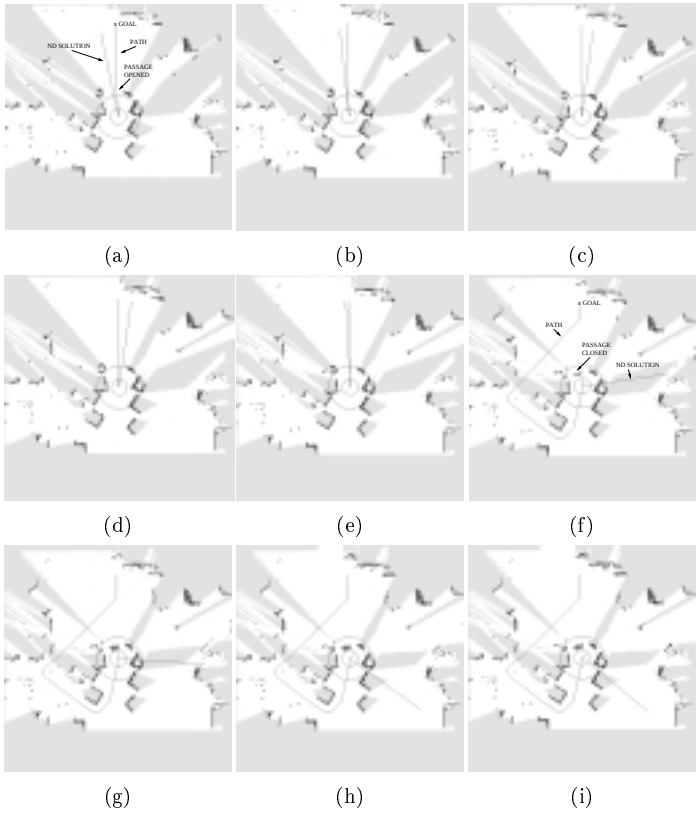

Fig. 3. mpND real experiment.

tion of this navigation function at each servo-tic during the robot control loop.

The path solution gives two important pieces of information:

1. *If it is or not possible to reach the final configuration from the actual robot configuration (global reasoning).*
2. *The* **instantaneous path direction** *in order to reach the final configuration (global reasoning).* The instantaneous path direction is the main direction of the first part of the path (in our current implementation the first meter is used to calculate it).

The case of unavailable trajectories will be discussed in next section. Once the path is calculated, it has to be linked to the reactive navigation method. The ND is modified to drive the robot towards the instantaneous path direction (when repeated at each servo tic assures convergence to the goal location), instead of directing the robot towards the goal location. See in Fig. 4 the complete mpND navigation scheme.

There are no obstacle configurations that produces **trap situations** (when a solution exists inside the grid-model), because the instantaneous path direction has the information needed to get the robot out of these situations. The reactive method only has to direct the robot towards this instantaneous direction to avoid trap situations. Moreover, the symmetries of the environment do not produce cyclic behaviors, because the possible alternate solutions are discriminated by the instantaneous path direction.

5.1 Experimental Results

The same platform and settings of the mND (Subsection 5.1) are used here. From mND to mpND, only the NF1 module is added, which introduces a time penalty of 100 ms. The complete servo-tic is then 250 ms, which is well-suited to achieve real-time performance.

Fig. 3 shows a real experiment where the robot is forced to fall into a trap situation. The navigation system has to react automatically to this situation and drive the robot out of it.

In Fig. 3a can be seen that the robot had to cross a passage to reach the final location. While the robot was traveling through the passage (Figs. 3b,c,d,e), a human blocked it. The robot was then inside of a big U-shape obstacle, what produced a trap situation, see Fig. 3f. Automatically, the new path calculated (and thus the instantaneous path direction) pointed out of the U-shape configuration. The ND generates the motion commands to follow this direction, see Fig. 3g,h,i. The result was that the robot was able to get out of the trap situation.

6 Global Nearness Diagram Navigation (GND)

Some researchers have signalled that it is possible to generate motion with a classical planner. But other researchers have used this result to generate reactive motion, using the planner in an iterative way. From our point of view this reasoning is not always valid. There exist two situations, where a planning algorithm does not find a solution, in order to connect the initial and final robot's configurations. If these situations appears, it is not possible to close the motion control loop, because it is not possible to generate the motion commands to follow the path.

Two situations produces this case:

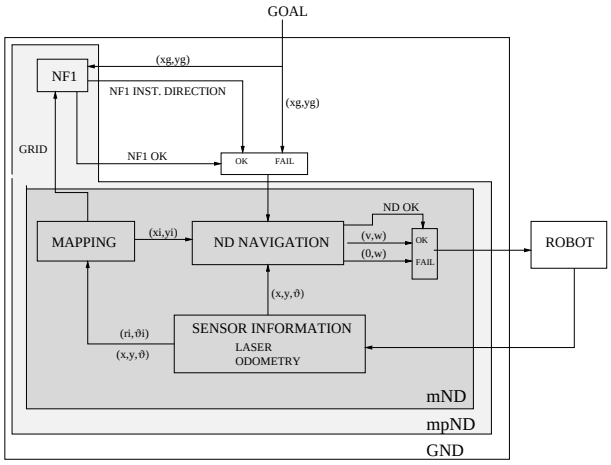

Fig. 4. GND navigation scheme.

1. *The final configuration is not in free space C_{free} [21]* (final location in collision with an obstacle). This is a very typical situation in unknown environments, where goals are iteratively placed for exploration. When the environment is incrementally discovered, the goal can be within an obstacle. In dynamic environments a mobile object can move, or even stop, at the goal location. Even in static and completely known environments, this situation appears when the goal moves to within an obstacle due to the robot drift.
2. *The robot or the goal are completely surrounded by an obstacle.*

One could think that these situations could be avoided by replacing the goal location. From our point of view, it is not the task of the navigation system to modify a final location imposed by an external agent, because the consequences can drastically determine the success of the global task.

Due to this shortcoming, we realize that when these situations occurs, the system should be able to continue its navigation task (close the motion control loop). To cope with this limitation, we developed the Global Nearness Diagram Navigation (GND). It combines the two schemes presented (mND and mpND) to achieve the complete navigation task, see Fig. 4 for the complete GND system.

The GND works as follows. First the mpND is used until a failure flag is produced in the NF1 module (a path connecting the initial and final configurations of the robot does not exist). Then the mND takes control and generates the motion commands. Now there are two possibilities: 1) the path is available (the control is passed to the mpND); 2) there is a ND failure. This last situation happens when there is no free walking area to move (the robot is completely surrounded by obstacles). The motion commands stop and rotates the robot about its center. This behavior updates the map in all directions. This continues until the environment changes, and the control can be transmitted to the mND or mpND.

The GND inherits the properties of the mND related to the **information integration** and the **dynamic environment reaction**. Moreover, the mpND is used when possible, thus avoiding the problem of **trap situations**. The motion commands in the GND are generated by the ND, which avoids many of the problems of other reactive schemes. These properties together make the GND a navigation system which is very well adapted to deal with unknown, dynamic, unstructured, dense and very complex environments.

6.1 Experimental Results

The GND system has been extensively tested with the XR4000 platform in LAAS (CNRS). In all the experiments the environment was unknown, and was incrementally explored. We have chosen two illustrative experiments to show the system behaving in dynamic, unstructured and complex environments (see Fig. 5). Due to the difficulty in reflecting the environment's dynamics, we decided to show the complete robot path and some snapshots of the experiment, rather than directly showing the sensory data and blurring thus the graphics.

Experiment 1: This experiment was designed to show the robot getting out consecutively of three trap situations, produced by changes in the environment's structure. The first snapshot shows the initial state of the robot and the environment, where the robot had to cross a passage to reach the goal location. When the robot arrived at the end of the passage, the right passage was opened (the robot could not see it), and the main passage was closed. Automatically the robot turned to get out of this trap situation. This part of the experiment can be seen step by step also in Fig. 3. When the robot was leaving the passage, it perceived the right passage and reacted to move the robot inside. Then the human closed this passage. The robot automatically reacted to get out of this new trap situation, and ended getting out of the global trap situation. Subsequently the robot resumed the motion towards the goal location.

Experiment 2: This experiment shows the robot in a typical populated environment. The first snapshot shows the initial state of the robot and the environment, where the robot had to cross a room to reach the final location. Humans were walking, building and modifying the environment randomly to disturb the robot's motion. During the experiment, the robot got trapped and had to move back to find the solution. The snapshots shows the highly dynamic nature of the environment.

7 Comparison with other methods

We next discuss the improvements of the GND over other navigation systems. We have chosen the more recent methods of the four techniques introduced in Section 2: Potential Field Methods (in a general fashion), the Vector Field Histogram (VFH* [10]), the Elastic Strip [19] and the Global Dynamic Window Approach (GDWA [14]).

The reader is directed to [1] for a comparison in purely reactive terms. The discussion here is oriented towards the three requirements introduced in Section 3: information integration, dynamic environments reaction and trap situations solution.

Potential Field Methods PFM

Many special solutions to the inherent limitations of PFM [6] still appear in the literature [23]. We have decided to orient the discussion with the PFM in a general fashion. In reactive terms, the ND (and thus the GND) solves all the inherent limitations of PFM except the trap situations (see [1] for a detailed discussion). Moreover, the GND avoids the trap situations when possible, solving also

these last undesirable situations.

Vector Field Histogram VFH*

The VFH* uses a grid map of the environment [7] to integrate information. To discuss the drawbacks and advantages of each approach, in terms of information integration and dynamic environments reaction, is out of place, mainly due to the different nature of the sensors used (ultrasonic sensor in [7] and laser in GND). In terms of trap situations, the VFH* uses a look-ahead verification to analyze the consequences of heading towards a candidate direction. As far as we understand, when using a look-ahead verification to increment the local nature of the method, it is necessarily to fix a maximum number of steps (named goal depth in VFH*), which translates in a maximum distance inspected (named total projected distance in VFH*). To select this distance could be a trade off between the speed and the validity of the method, because it represents the maximum reach of the local nature of the method (measured in robot distance traveled). On the other hand, the GND assures global convergence inside of the map grid used.

The main advantage of a look-ahead verification is when dealing with platforms with low computational capabilities. The look-ahead then is well adapted because even by reducing the projected distance, good results can be obtained. Running the GND in real time requires high computational capacity, otherwise one can reduce the size of the grid, which drastically affects the reach of the solution.

Elastic Strip ES

The elastic strip framework [19] has been shown to work very well in high-dimensional configuration spaces. The discussion here is focused in low-dimensional configuration spaces, that is the case of this paper. Two strategies were introduced in [18] and [19] to deal with dynamic environments. The former is to impose constraints on the internal forces acting on two adjacent configurations, to allow a mobile obstacle to pop through the elastic strip. The second one is to maintain a set of alternative routes to chose when the elastic becomes invalid.

The elastic framework is based in the existence of a path that is not always available (see Section 6). As shown in [18], the elastic strip framework could fail in very tight or cluttered environments, where the GND is well-adapted due to the properties of the reactive method ND.

Global Dynamic Window Approach GDWA

The evolution of the GND has been inspired by the GDWA [14], [18]. We direct the comparison to the advantages/disadvantages of the model used, and to some implementation details.

- *Model*: The GDWA uses an occupancy-grid that represents the configuration space of the robot, and remains fixed in a global reference. We understand that the motivation to represent the configuration space is to not completely rebuild it at each step of the algorithm. While keeping the occupancy grid fixed in space, the same navigation function can be reused for every robot location, as long as the environment does not change. The GND uses an occupancy-grid to represent directly the working space, which moves centered around the robot's position.

To use a model that moves with the robot means that the dimension of the model does not depend on the distance traveled (the robot was at all times surrounded by the grid). Moreover, it ensures that the instantaneous surroundings of the robot are directly represented by the model. We would like to signal that the operation to displace a complete grid can be very efficiently implemented in terms of memory (in our current implementation takes about 10ms a 200 by 200 cells, i.e. 10m by 10m grid). On the other hand, the solution has to be found inside of the grid.

In GDWA, the data measured by the laser is translated into configuration space obstacles [21], that are represented in an occupancy grid. From our point of view, this framework does not represent fully the environment's dynamics. The reason is that in a laser measure, the obstacle point can be translated into the configuration space. But the line joining this point to the sensor, that has free space information, cannot be used to update the configuration space. So, only configuration space obstacles is updated, but not the free space. The consequence is that the robot avoids parts of the space that are free of obstacles (see experiment in Section 4, where the passage will remain blocked to the GDWA after the person crosses it). This information of the free space is lost in the GDWA while it is exploited by the GND.

- *Implementation details*: The GDWA computes the navigation function for a subsection of the configuration space, referred to as localized navigation function (localized NF1). The subsection of the configuration space is increased while looking for a solution. This heuristic saves a lot of computational time currently lost by the GND (that computes the NF1 for the complete grid), and should be added to the GND. The multi-resolution GDWA is also presented [18], to deal with the impact of computational complexity of the size of the occupancy grid used.

The GDWA extracts information from the NF1 by examining the neighborhood of the grid cells that corresponds to the robot location. As shown in [18] and [14] some unnatural behaviors were found, because the border effects of the NF1, and because the solution can only be multiple of 45°. The GND calculates the complete path using a gradient-search technique. Subsequently, the path is tightened using a recursive algorithm and the instantaneous path direction is calculated. Using the instantaneous path direction, avoids the border effects of the NF1 and the constrained solutions, and thus no unnatural behaviors were found.

8 Conclusions

This paper presents a new navigation system that links global information with a local reactive scheme to generate motion. The GND uses the sensory information to build a grid representation of the environment. A planning algorithm is used to extract global information from the model. Finally, the reactive scheme ND uses the computed path and the model to generate collision free motion while directing the robot towards the final location.

We also present how some reactive techniques evolved in the last years, to discuss the advantages/disadvantages of this system among other existing methods. Experimental results in unknown, unstructured, dynamic and complex environments are also shown.

Acknowledgement

We thank R. Alami and T. Simeon of LAAS/CNRS (FRANCE) for accepting J.Minguez in their working group. Moreover, we thank S. Fleury of LAAS/CNRS (FRANCE), for her help with algorithm implementation on the XR4000 Nomad platform.

References

[1] J.Minguez L.Montano "Nearness Diagram Navigation. A New Real Time Collision Avoidance Approach" *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)*, Takamatsu, Japan, 2000.

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

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

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

[5] 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.

[6] 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.

[7] J.Borenstein and Y.Koren "Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance" *IEEE Transactions on Robotics and Automation*, pp 535-539, 1991.

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

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

[10] I.Ulrich and J.Borenstein "VFH*: Local Obstacle Avoidance with Look-Ahead Verification" *IEEE International Conference on Robotics and Automation*, pp2505, San Francisco, USA, 2000.

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

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

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

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

[15] 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.

[16] Sean Quinlan "Real-Time Modification of Collision Free Paths" *PHD thesis, Stanford University* December, 1994.

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

[18] Oliver Brock "Generating Robot Motion: The Integration of Planning and Execution' *PHD thesis, Stanford University* November, 1999.

[19] O.Brock and O.Khatib "Real-Time Replanning in High-Dimensional Configuration Spaces Using Sets of Homotopic Paths" *IEEE International Conference on Robotics and Automation*, pp 2328, San Francisco, USA, 2000.

[20] J.Foley, A.Van Dam, S.Feiner, J.Hughes, "Computer Graphics, principles and practice" *Addison Wesley*, edition 2nd, 1990. The system programming series.

[21] J.C.Latombe "Robot Motion Planning" *Kluwer Academic Publishers*, 1991.

[22] D.Maravall J.de Lope and F.Serradilla "Combination of Model-based and Reactive Methods in Autonomous navigation" *IEEE International Conference on Robotics and Automation*, pp 2328, San Francisco, USA, 2000.

[23] L.Chenqing M.Ang H.Krishman L.Yong "Virtual Obstacle Concept for Local-minimum-recovery in Potential-field Based Navigation" *IEEE International Conference on Robotics and Automation*, pp 983, San Francisco, USA, 2000.

(a) (b)

Fig. 5. a) Experiment 1. b) Experiment 2.