Autonomous Motion Generation for a Robotic Wheelchair

J. Minguez L. Montesano L. Montano
I3A, Dpto. de Informática e Ing. de Sistemas
Universidad de Zaragoza, Spain
{jminguez,montesano,montano}@unizar.es

Abstract— **The video shows a robotic wheelchair autonomously driven by a sensor-based navigation system in an office scenario. The objective of the experiment displayed is to reach a location out of a room. This is achieved by a sensor-based motion system composed by three different modules: a modeler of the environment, a tactical planner and an obstacle avoidance method. The integration of these modules provides the wheelchair with the capability to move in unknown, dynamic and complex scenarios (where there is very little room to maneuver and that create the well-known trap situations). These are the work conditions that are difficult for many existing techniques.**

I. INTRODUCTION

Robots are being developed that operate under a wide variety of conditions including unknown, unstructured and dynamic scenarios. Mobility in such environments is a key issue to increase the degree of autonomy of a robotic platform since it is the basis to incorporate more functionalities. Indeed the generation of robust motion is becoming an important technological issue in many applications. A good example is the vehicle used in this work, a robotic wheelchair. The objective is to incorporate a motion system able to autonomously drive the vehicle among locations while avoiding collisions. Here the motion performance is very important since it is the basis of the application, and failures could put the end-user and people sharing the workspace at risk.

In autonomous robots, the mobility aspect is inherently related with at least three aspects: modeling, planning and reaction. The systems that incorporate these skills are usually called hybrid systems (hybrid architectures). In general, they combine a planner (deliberation) and a reactor (execution) working on a basis of a model. These systems are widely spread in the robotics community [1], [8], [10]. We outline in this paper our hybrid system and how we have integrated it in a robotic wheelchair [7], [5]. The video shows a real experiment in a difficult scenario: dynamic, dense, complex and cluttered with obstacle configurations that create trap situations and cyclic behaviors. The experiment demonstrates how the system carries out a robust, efficient and trustworthy navigation under these circumstances. In fact this is the novelty of the approach, to be able to navigate in these scenarios, which are a challenge for many existing techniques.

II. OVERVIEW OF THE SYSTEM

The hybrid motion system is formed by an architecture that integrates model construction, motion planning and obstacle avoidance functionalities:

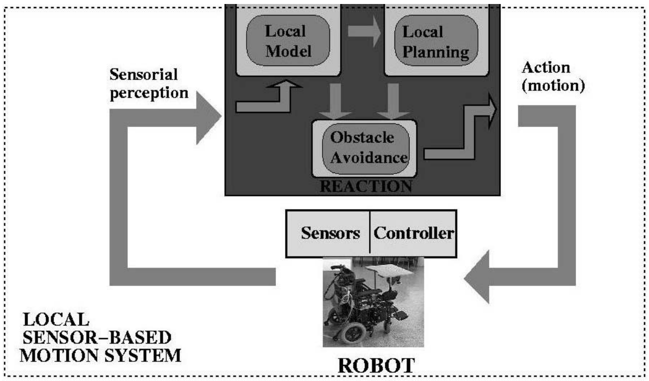

Fig. 1. Sensor-Based Navigation architecture

1) **Model builder**: construction of a model of the environment (to increase the spatial domain of the planning and used as local memory for obstacle avoidance) and tracking of the vehicle position.
2) **Planner**: extraction of the connectivity of the free space to increase the spatial domain of the solution (for instance used to avoid the cyclical motions and trap situations).
3) **Reactive motion**: computation of the collision-free motion.

Globally the system works as follows (Figure 1): given the information provided by the sensors, the model builder incorporates it into the existing model. Next, the model is used by the planner module to compute the course to follow to reach the goal. Finally, the avoidance module uses the information of the obstacles contained in the model and the information of the tactical planner to generate the motion (to drive the vehicle free of collisions towards the goal). The motion is executed by the vehicle controller and the process restarts with the new sensor measurements. In this context, the integration of functionalities plays a crucial role. It assures that the three modules work synchronously within the perception - action cycle providing the framework for coordination, failure detection and recovery. Next, we describe the design of the modules and the integration architecture.

III. MODULE DESCRIPTION

We describe in this Section the model builder module (Subsection III-A), the planner module (Subsection III-B), the obstacle avoidance method (Subsection III-C) and the architecture of integration (Subsection III-D).

A. The model builder

The function of this module is to integrate the sensor measurements to construct a local map of the environment. We choose a binary occupancy grid to model the obstacles and the free space. The grid travels centered with the robot and has a limited size to represent the portion of environment necessary to solve the navigation.

The design of this module includes two parts: (i) To improve the vehicle odometry, we use a scan matching technique with the information provided by the laser [2], [6]. Although these techniques do not guarantee global consistency, its precision is enough to build the local map needed by the other modules. (ii) To integrate a scan in the model, we update all the cells that correspond to the area scanned (free and occupied space). With this strategy, the last laser scan integrated in the grid does not have any odometry error with respect to the current location of the vehicle. Only the non updated cells accumulate errors, which are, however, mitigated by the scan matching technique. Furthermore, the grid rapidly reflects the changes in dynamic environments and the previous spurious measurements are eliminated with the new observations.

B. The planner

This module uses a dynamic navigation function (D^*Lite planner [9]) to obtain tactical information and avoid the trap situations and the cyclic motions. The principle of this planner is to locally modify the previous path (available from the previous step) using the changes in the scenario.

The module has two different parts. The first part is the computation of the obstacle changes in the configuration space (notice that the grid represents the workspace). This is efficiently carried out by defining operators between the previous map and the current one updated with the last perception. The map of changes is the input of the D^* Lite planner.

The second part of this module computes a path in the configuration space from the current vehicle location towards the goal. The $D*$ Lite planner models the navigation problem with a graph. Each vertex represents a location of the vehicle, and has associated one edge that points to the vertex with lower cost to reach the goal. Changes in the environment modify these costs. The planner locally explores and rearranges the edges affected by the changes that are relevant to compute the shortest path. This strategy is by far more efficient than exploring the complete grid to compute the path (up to two orders of magnitude [11]).

From this path we obtain the *tactical motion direction* (initial course of the path). Notice that this direction will not be used to direct the vehicle (since this degree of freedom will be handled by the reactive module), but as the main course of the motion.

C. The reactive module

The design of the ND method [4] is based on defining a set of situations to represent the navigational problem, and on how to act in each of them (actions). In real time, at each control cycle a situation is selected and the corresponding action computes the motion. In fact, this method employs a ”divide and conquer” strategy based on situations to simplify the difficulty of navigation. Thus, this technique is able to deal with complex navigation cases (usually these cases arise in environments where there is little room to manoeuvre like for example a narrow door). In particular, the ND method avoids most of the problems that other techniques present in these circumstances, like the local trap situations, the oscillating movements, or the impossibility to move towards certain zones with high obstacle density or far away from the goal direction (see [4] for a discussion on this topic). In addition, we used [3] to adapt this method to deal with the vehicle shape, kinematic and dynamic constraints of the vehicle.

D. Integration Architecture

The modules are integrated considering the limitations and restrictions imposed by the mechanical (sensors and actuators) and logical parts (computers) of the robot, and possible failures and recovery procedures in the software modules.

The three main modules are executed following the modeler-planner-reactor sequence driven by the flow of data between modules. This flow is unidirectional, from the modeling module toward the planner and obstacle avoidance modules. The modules are executed synchronously. This is important to avoid inconsistencies in time that would arise using asynchronous strategies (the model is used for local planning and obstacle avoidance and must be consistent in time with both modules). Furthermore, we assigned time outs to each module to close the motion control loop at the desired sensor rate. The purpose of these timeouts is to assure that the obstacle avoidance module is executed every cycle. This is important since the motion of the system is always generated by the avoidance method (assuring collision free motion). All the modules have been integrated in such a way that the control loop is always closed at $5Hz$ (sensor frequency) with a motion command available (there are no dead states).

IV. THE EXPERIMENT

We used a commercial wheelchair equipped with a SICK laser and two on-board computers (two $PentiumIII850Mhz$, one of them is used for motion control purposes and the other executes the architecture describe above). The wheelchair is a differential drive rectangular (1.2m×0.7m) vehicle. We set the maximum operational velocities to $(v_{max}, w_{max}) = (0.3\frac{m}{sec}, 0.5\frac{rd}{sec})$ due to the application context (human transportation).

The experiment outlined in the video is particularly difficult due to the vehicle used, the type of task and the nature of the surroundings. The wheelchair is a non holonomic robot with the driving wheels in the back part. Therefore, it cannot move in any direction and sweeps an ample area when it turns.

(a) (b) (c)

Fig. 2. These figures show the wheelchair moving in the office environment and the model computed with the planner and reactive method solution at a given time. (a) Moment when the trap situation was produced; (b) Moment when the wheelchair performs the avoidance of the chair on the path and the moving obstacle. (c) Model constructed and the solutions of the tactical planner and the reactive method.

In addition, the vehicle transports humans which requires the avoidance of abrupt movements and shaking behaviors. In other words, the vehicle has geometric, kinematic and dynamic constraints that must be taken into account. The laser sensor is placed in the front part of the robot (0.72m) and has a 180° field of view. Therefore, some obstacles are not visible from the current position (although they can collide with the vehicle). On the other hand the surroundings are unknown, since there are elements in the office like chairs, tables whose position cannot be established a priori. Although the walls could be known a priori, most of the time they are occluded by the furniture. This scenario is not prepared to move a wheelchair and in many places there is little room to move. Finally, people turn the scenario into a dynamic and unpredictable environment, modify the structure and may create global trap situations.

The objective of the experiment was to drive the vehicle to a location that was out of the office. All the scenario was unknown and the only a priori information was the goal location. Initially, the vehicle moved towards the closest visible door. During the motion, a person closed the right leaf of the door so that the wheelchair could not cross it. At this moment the vehicle was trapped in a large U-shape obstacle (Figure 2a). Rapidly the tactical planner computed the most suitable area to get out of this situation, and the obstacle avoidance method moved the vehicle to this zone. As a result, the vehicle quickly modified its way returning backwards and overcoming the trap situation. During the travel, the robot avoided collisions with the furniture and with a person who moved bothering the normal progression of the vehicle (Figure 2b). In particular, a chair was placed in the vehicle path. Rapidly the change was incorporated to the model and the reactive method performed the avoidance (Figure 2c). Finally, in the center of the office, the wheelchair detected a very narrow door but sufficiently wide to fit in. The obstacle avoidance method drove the vehicle toward this door, crossed it and left the office reaching the final position. Notice that crossing the door was difficult from the motion generation point of view and was successfully solved by the obstacle avoidance method.

V. CONCLUSION

We have presented in this paper results with a sensor-based navigation system integrated in a robotic wheelchair. The system is made up of three modules: a model builder, a planning method and a reactive navigation method. The advantage is that the system is able to generate sensor-based autonomous motion for vehicles in realistic scenarios, which are usually complicated, there is very little room to maneuver, are highly dynamic or create the well-known trap situations. We have demonstrated the usage with a wheelchair vehicle under these work conditions.

REFERENCES

[1] O. Brock and O. Khatib. High-Speed Navigation Using the Global Dynamic Window Approach. In *IEEE Int. Conf. on Robotics and Automation*, pages 341–346, Detroit, MI, 1999.

[2] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Intelligent and Robotic Systems*, 18:249–275, 1997.

[3] J. Minguez and L. Montano. Robot Navigation in Very Complex Dense and Cluttered Indoor/Outdoor Environments. In *15th IFAC World Congress*, Barcelona, Spain, 2002.

[4] J. Minguez and L. Montano. Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios. *IEEE Transactions on Robotics and Automation*, 20(1):45–59, 2004.

[5] J. Minguez and L. Montano. Autonomous Sensor-Based Motion Control in Unknown, Dynamic and Troublesome Scenarios. *Robotics and Autonomous Systems*, 52(4), 2005.

[6] J. Minguez, L. Montesano, and F. Lamiraux. Metric-based iterative closest point scan matching for sensor displacement estimation. *IEEE Transactions on Robotics (to appear)*, 2006.

[7] L. Montesano, J. Minguez, and L. Montano. Lessons learned in integration for sensor-based robot navigation systems. *International Journal of Advanced Robotic Systems (in press)*, 2006.

[8] R. Philipsen and R. Siegwart. Smooth and efficient obstacle avoidance for a tour guide robot. In *IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan, 2003.

[9] A. Ranganathan and S. Koenig. A reactive architecture with planning on demand. In *International Conference on Robotics and Automation*, page 14621468, Las Vegas, Nevada, 2003.

[10] C. Stachniss and W. Burgard. An Integrated Approach to Goal-directed Obstacle Avoidance under Dynamic Constraints for Dynamic Environments. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pages 508–513, Switzerland, 2002.

[11] A. Stenz. The Focussed D^* Algorithm for Real-time Replanning. In *Proceedings of the Intenational Joint on Artifcial Intelligence*, pages 1652–1659, 1995.