Proceedings of ICAR 2003
The 11th International Conference on Advanced Robotics
Coimbra, Portugal, June 30 - July 3, 2003

Fine Motion Planning for Shared Wheelchair Control: Requirements and Preliminary Experiments

E. Demeester M. Nuttin D. Vanhooydonck H. Van Brussel

Department of Mechanical Engineering
Katholieke Universiteit Leuven
Celestijnenlaan 300b, B-3001 Heverlee, Belgium
Eric.Demeester@mech.kuleuven.ac.be

Abstract

Many elderly and disabled people today experience difficulties when manoeuvring an electric wheelchair. In order to help these people, several heuristic navigation algorithms have been devised in the past. However, in order to make the chair fast and reactive, the wheelchair's dimensions, non-holonomic and dynamic constraints are often only approximately taken into account. Consequently, these robotic wheelchairs may and do fail in executing the very same manoeuvres with which the elderly and disabled have problems.

A possible approach to tackle this problem, is to use a fine motion planner that takes the kinematic and dynamic constraints explicitly into account. This paper discusses the requirements for such wheelchair fine motion planners. Also, an enhanced version of a previously developed planner for autonomous robots is presented, and it is evaluated for use on wheelchairs.

1 Introduction

Wheelchairs are often used in places where fine manoeuvres are necessary to reach a certain goal. Examples of such places are bathrooms, elevators, and corridors in houses and public buildings. The relatively large size of wheelchairs as compared to their environment requires the user to be very dexterous in manoeuvring the chair. Moreover, these manoeuvring tasks are complicated by the non-holonomic characteristics of most wheelchairs.

However, many elderly and disabled people are not able to control their wheelchair properly, or they find it a very tiresome and frustrating task. Therefore, many algorithms have already been devised in the past to help people in their daily manoeuvring tasks: OMNI, Bremen Autonomous wheel chair, RobChair, Senario, Drive Assistant, VAHM, Tin man, Wheelesley, and Navchair (see e.g. [1], [2], [3]). Often, these algorithms model the wheelchair's geometry and kinematics only approximately. As a consequence, these algorithms contain a lot of heuristics and parameters to generate a safe behaviour on the desired platform. Therefore, a lot of tuning is necessary, but the parameters do not always have a comprehensive, physical meaning. This in turn limits these algorithms' portability to other wheelchair platforms. Since the non-holonomic character of most wheelchairs is often not taken into account, these algorithms furthermore are not successful in every situation. For example, a drive-through-door algorithm based on heuristics might not work if the chair approaches the door at an angle rather than from directly in front of the door (see e.g. [2]). However, it is the authors' opinion that robotic assistive wheelchairs should at least be able to execute the very same manoeuvres as wheelchair users have to.

Figure 1: *The robotic assistive wheelchair Sharioto*

2 Fine motion planning and shared wheelchair control

The fine motion planner which will be described later in this paper is being developed in the context of our shared control research for wheelchairs. Shared control is the situation in which the control of an (assistive) device is shared between a user and an "intelligent" controller of that device. This device may be a wheelchair, a tele-operated robot, an airplane, a robotic travel aid for the visually impaired, a robotic assistive manipulator, or any other device where robot and human co-operate in a task. The purpose of shared control is to combine the strengths of both the human and the controller. Whereas humans excel in global planning and coarse control, exact motion planning is one of the tasks that is performed better by robots. Therefore, fine motion planning fits very well in the shared control framework.

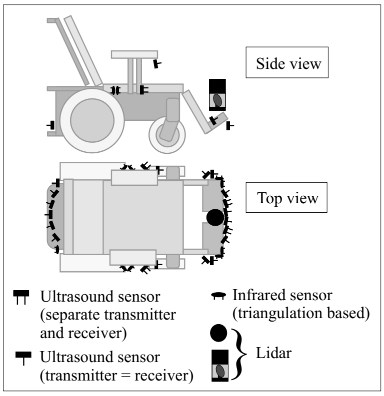

Figure 2: *Positioning of the exterioceptive sensors on Sharioto*

The approach to shared control followed here differs from previous approaches in the fact that a path planner is used as the basic navigation assistance algorithm. Most of the assistive robotics literature focuses on heuristic algorithms because the chair should behave fast and reactive. Indeed, a fine motion planner requires a map of the environment, and it takes time to calculate both this map and a path in it. However, the question remains whether a wheelchair should be very fast. This may be a requirement for an autonomous robot, but a wheelchair driver usually expects a smooth and comfortable driving behaviour. The chair should behave safely and deterministically. So if an obstacle is encountered, the best behaviour may be for the chair to slow down or to stop while waiting for new input from the user, rather then taking autonomously the decision to avoid the obstacle. Moreover, fine motion planners are usually only necessary to plan paths in small, local environments, say $4m$ x $4m$, so that time requirements are less stringent.

The advantage of fine motion planning is that the goal generation is decoupled from the motion generation. The fine motion planner focuses on the motion problem alone, i.e. robot dimensions, kinematics, and dynamics. Most of the previous wheelchair research has embedded both the motion problem and the generation of a goal position or direction in the same algorithm. For example, a dock-at-table algorithm embeds both the generation of a goal position (somewhere on the table) and the motion problem (how to get there). As a consequence, every time a new manoeuvre has to be added to the functionality of the wheelchair, a new algorithm has to be devised. And every time again, the motion problem has to be taken care of. Using a fine motion planner, only the generation of the goal position should be devised for every new manoeuvre, because the motion problem has been taken care of once and for all.

In the near future, we want to use this fine motion planner as an on-line evaluation mechanism for the driver's current driving capabilities (see figure 3), in order to decide in real-time whether or not more assistance should be provided or proposed to the user.

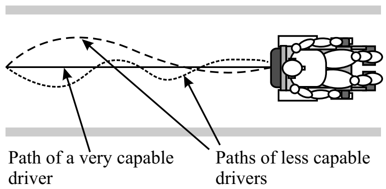

Figure 3: *Evaluate on-line the user's driving behaviour*

Not only do we want to assess the user's driving behaviour, we also want to predict where (s)he wants to go to. In order to assist people without irritating and frustrating them, their intentions and plans should be estimated as accurately as possible. Currently, a probabilistic framework using Bayes' rule is being developed to estimate these intentions on-line. This framework makes use of a "user model", i.e. a model that predicts which joystick signals $\mathbf{v_c}$ the user would give if (s)he wanted to go to a certain goal pose $\mathbf{p_e}$ with a certain end velocity $\mathbf{v_e}$ in a certain environment s, starting from pose $\mathbf{p_c}$:

$$P_{user}\left(\mathbf{v_c}\middle|\begin{bmatrix}\mathbf{p_e}\\ \mathbf{v_e}\end{bmatrix}, \mathbf{p_c}, s\right) \tag{1}$$

The use of fine motion planning comes with other advantages. A planner, possibly in combination with force feedback, may be used for teaching wheelchair drivers how to drive a wheelchair. Currently, especially elderly people may be helped during the training phase if they know they cannot damage the environment by colliding with it.

2.1 Previous work on fine motion planning

Many of the existing fine motion planners rely upon accurate, static models of the environment. An overview of these techniques, such as road-map, cell decomposition and potential field methods can be found in [4]. However, these algorithms fail if the world model changes on the fly or if the robot's dynamics should be taken into account (this is called "kinodynamic" planning).

Therefore, new planning techniques have been devised. Randomized or probabilistic planning techniques for example trade an amount of completeness against a major gain in computing efficiency, which makes them suitable for robots with many degrees of freedom (see e.g. [5]). These techniques have also been applied to kinodynamic planning. Also, rapid replanning techniques have been devised for use in dynamic environments (see e.g. [6]).

Before evaluating any planner in the context of shared wheelchair control, it is necessary to state which requirements are valid for wheelchair fine motion planning.

3 Requirements for wheelchair fine motion planning

A wheelchair cannot just be seen as an autonomous robot with a person on it. Not only does the chair have a different shape (long rectangle) as compared to classical autonomous robots (often round or square), several additional constraints have to be taken into account, exactly because human and robot are working together in order to transport the human to a certain place (cf. [7]). Some of the most imperative requirements for wheelchair fine motion planners are:

(Re)planning in real-time In order for the planner to be useful and reactive, it should be able to (re)plan a path in "real-time", which allows the controller to take recently sensed environmental information into account. On the platform we use, new sensor data arrive every 200 ms, and every 40 ms new signals are sent to the motors. In order to satisfy these real-time constraints, it may be better to produce a suboptimal but feasible path on time, rather than an optimal path with a certain delay.

Exactness The *fine motion* planner should be exact, in the sense that it should take the wheelchair's dimensions and its non-holonomic kinematic constraints into account (as opposed to a *gross motion* planner that only takes these constraints approximately into account, e.g. by using a grow-up space).

Completeness The planner should be complete, i.e. if a path to the goal configuration exists, then the planner should find one.

Robustness The planner should be robust to false or inaccurate sensor measurements, and to inaccurate motor control due to wheel slippage for example. Consequently, it may be necessary to plan paths that have a maximum clearance to obstacles.

Dynamics Because the planner is to be used on wheelchairs, one of the most important requirements is to produce a comfortable and smooth driving behaviour. In general, wheelchair manufacturers take several measures to ensure a comfortable drive behaviour. Firstly, a "restrictor" plate is adopted. This is a plate that restricts the joystick's position to a certain area, which ensures that the linear and angular velocity cannot be high at the same time, thereby guaranteeing (in hardware) a smooth driving behaviour during turning. Secondly, parameters regarding maximum velocity and acceleration can be programmed. A velocity profile is applied to the raw joystick signals that makes sure that these velocity and acceleration thresholds are not exceeded. A typical velocity profile and restrictor plate are shown in figure 4.

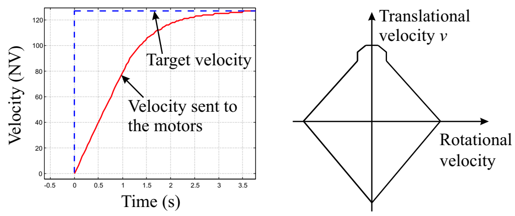

Figure 4: *(a) A typical velocity profile and (b) Sharioto's restrictor plate*

Moreover, when users drive a wheelchair, the frequency content of their joystick signals is limited. Typically, the most important frequencies do not exceed 5-6 Hz. If possible, the path planner should somehow take these constraints into account and emulate a similar driving behaviour.

Adaptability As explained above, the planner might be used in the future as the basis for a user model. Consequently, the planner should be able to plan paths for different wheelchair types (car-like or differentially driven), and for different user interfaces, both switch-based (e.g. sip and puff systems or scanning interfaces) and proportional (e.g. hand or chin joysticks), see also [8].

4 Path planning algorithm

The path planner described here was originally developed at our department for an autonomous robot [9]. The planner was enhanced by using the A* algorithm to search the tree instead of a breadth-first strategy in order to speed up the algorithm. In section 5.2, the planner's performance will be evaluated for use on a wheelchair.

In the first planning phase, the free configuration space is calculated. Then, the tree of possible paths is searched with A*. In a last step, the obtained path is smoothed.

4.1 Calculation of the free configuration space

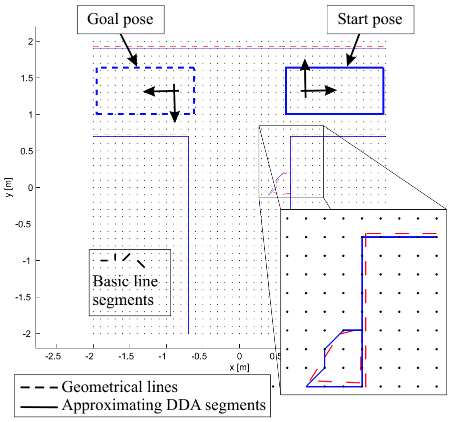

Figure 5: *A possible map of the chair's environment, its start and goal pose, and the approximation by short line segments*

This step takes the dimensions of the wheelchair into account. All collision free configurations in the sensed map (also called the "free configuration space" or "free C-space") are calculated before performing the actual planning.

In order to do so, the planning area is disretized and limited to 4 by 4 meters, with 40 disretizations for both x- and y-axis, and 40 discrete angles. The total number of configurations is therefore 40^3 (64 000). The planner's input consists of an environmental map with line segments. First, every line in the world model is approximated by four types of short line segments (with orientations 0 deg, 45 deg, 90 deg and -45 deg) using the DDA[1] integrator described in [10]. Figure 5 depicts an example of a possible map of the chair's environment, and shows the approximation of this map by the short line segments.

Next, the total free C-space is calculated. For each of the four basic line segments shown in figure 5, the robot configurations that collide with the line segment are calculated off-line. The on-line calculation of the complete free configuration space then reduces to a fast OR-convolution of the configurations occupied by all short line segments.

The total free C-space for the approximated map presented in figure 5 is shown in figure 6, together with the robot's collision free orientations at position $(-0.2, -0.4)$. This free C-space can now be searched for possible paths to the goal pose.

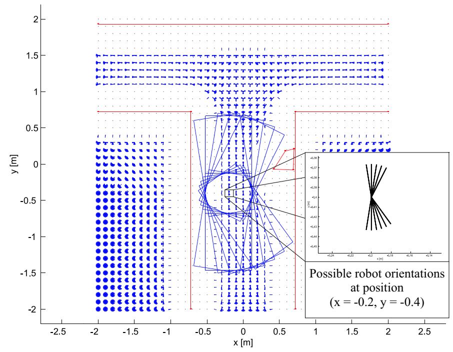

Figure 6: *Complete free configuration space for the approximated map in figure 5*

4.2 Tree search with A*

This step performs the actual planning and takes the wheelchair's kinematic and dynamic constraints into account. Therefore, it makes the algorithm easily adaptable to different wheelchair types and user interfaces. Moreover, the planning method is complete, since the A* algorithm will keep searching for a solution until it either finds a solution or it is certain that no solution exists. The real-time constraints are taken into account as much as possible by calculating the successor nodes off-line.

4.2.1 Off-line calculation of the successor nodes. The nodes that can be reached from a starting node are called "successors". In order to speed up the search, the relative position of the successors is calculated off-line for each of the 40 discrete orientations.

To determine a node's reachable configurations, the control variables v and ω are discretized as follows: linear velocity $\{-v, v\}$ and angular velocity {10 values ranging from -80% of ω_{max} to +80% of ω_{max}}. The reason for using only 80% of ω_{max} will be explained in section 4.3. ω_{max} is determined as follows. Since our wheelchair is differentially driven, it is not limited by a minimal turning radius. However, to obtain smooth trajectories and to make the here described method more general for all car-like vehicles, we introduce a constraint to the maximum angular velocity ω for a

[1] "Digital Differential Analyzer"

given minimal turning radius ρ_{min} (say, $1m$) and a linear velocity v (chosen to be 0.3 m/s):

$$\omega_{max} = \frac{|v|}{\rho_{min}} \tag{2}$$

Twenty paths can be constructed starting from a given configuration, by integrating the combinations of the velocities over a certain path distance l_{inc}, taking the non-holonomic constraints into account. Only those discrete configurations that lay close (within 20% of the cell size) to one of the 20 paths are selected as discrete successor points. l_{inc} is set to three times the discretization size. Figure 7 shows the successors for robot configuration [0 0 6] (x = 0, y = 0, $\theta = 6 \cdot 9$ deg = 54 deg).

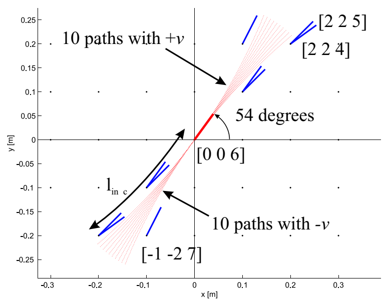

Figure 7: *Determination of successor nodes for orientation 6 (54 deg)*

4.2.2 Construction of the graph.

The graph is constructed using the A* algorithm which minimizes a certain cost function. At this moment, the cost function combines the path distance $\sum \Delta l$, the sum of all rotations $\sum \Delta\theta$, and the number of reversals nr_{rev}. A reversal is a configuration where the tangent to the path changes sign. Distance and rotation are scaled with the straight line distance $dist_{sg}$ from start to goal and with the orientation difference rot_{sg} between start and goal respectively. The cost factors are combined according to the following rule:

$$cost_{path} = \alpha_{dist} \cdot \frac{\sum \Delta l}{dist_{sg}} + \alpha_{rot} \cdot \frac{\sum \Delta\theta}{rot_{sg}} + \alpha_{rev} \cdot nr_{rev} \tag{3}$$

Figure 8 shows the planned path for cost parameters $\alpha_{dist} = 1$, $\alpha_{rot} = 2$, and $\alpha_{rev} = 2$.

4.3 Path smoothing

In section 4.2.1, only those discrete configurations were selected as successors whose distance to any of the feasible paths is less than 20% of the cell size. Consequently, the consecutive points on the final path do not lie on a smooth curve.

Therefore, in a final step, the actual successors are calculated sequentially for each path node by integrating the motion equations for linear velocities $\{-v, v\}$

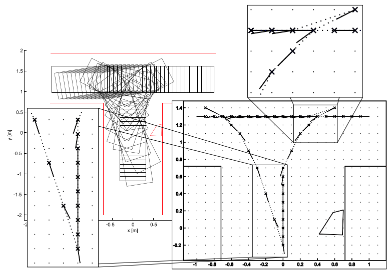

Figure 8: *Unsmoothed path*

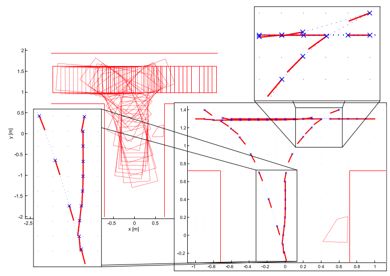

Figure 9: *Smoothed path of figure 8*

and angular velocities {25 values ranging from $-\omega_{max}$ to $+\omega_{max}$}. In order to guarantee that the discrete configurations are reachable in this step, only angular velocities till 80% of ω_{max} were adopted for the generation of the successor nodes in section 4.2.1 .

5 Hardware platform and preliminary experiments

5.1 Hardware platform

Our wheelchair "Sharioto" (depicted in figure 1) is a standard electric wheelchair which is differentially driven. Joystick and motors communicate via a CAN bus. Figure 2 depicts the positioning of several sensors that have been added to the chair. During a two and a half year multi-disciplinary research project, two novel sensors were developed: a LIDAR (Light Detection and Ranging sensor), and 4 ultrasound sensors without a dead zone. In the project, user requirements were collected from several user groups in hospitals. At the moment, a hand and chin joystick are available, and experiments with voice control have been performed. A laptop with Pentium III 418 MHz processor reads the sensors using two PCMCIA cards (DAQ-700 from National Instruments) and connects to both the joystick and the motors via a CAN-bus driver.

The laptop performs the navigation assistance. Velocity commands from the user interface are redirected via the laptop, corrected, and sent to the motors. In the project, three navigation algorithms have been developed, namely stopping in time, avoiding obstacles, and docking at a table. These were tested during experiments done with patients suffering from paresis, tremor, and spasticity. For more information, see [7] and [8].

5.2 Preliminary experiments

The algorithm explained in section 4 has been tested in simulation by using environmental maps generated with Matlab and is currently being implemented on the wheelchair. Sharioto's laptop needs 3340 ms to plan the path shown in figure 9. The number of expanded nodes amounts to 4639. The time required to construct the complete C-space is proportional to the number of line segments. For the 130 short line segments in figure 5, the construction of the C-space takes 460 ms. The search in this C-space requires 2600 ms, and the path smoothing 280 ms.

The only parameters that have to be tuned are the weights of the different cost factors in the cost function (3). By tuning these costs, different drive behaviours may be emulated, which is necessary if the planner is to be used as a basis for a user model. For example, cost factors that penalize rotations to the right may be added to the cost function in order to emulate the drive behaviour of a person that tends to drive to the left (because of his or her pathology).

Though the planner seems promising for use on autonomous robots, the application to wheelchairs is not straightforward. The wheelchair's dynamic behaviour should be modelled more accurately and the real-time performance is not yet satisfactory. Further, it can be seen that the smoothing step is rather computationally expensive and may yield collisions with line segments, because in the smoothing step no collision checking is done.

6 Conclusions and future work

This paper discusses the requirements for wheelchair fine motion planning in the context of shared control. Preliminary results have been shown with a fine motion planner that takes the dimensions and the kinematic constraints of a wheelchair into account. This algorithm's performance will be used as a reference for further developments on similar fine motion planners.

In the near future, we want to take the wheelchair's dynamics more accurately into account so that no smoothing is necessary. Also, efforts will be done to adhere more to the real-time constraints of the planner.

Acknowledgments

This research has been carried out with the financial help of the Flemish government under the ITEA project AMBIENCE, GOA project 99/04, and the Inter-University Attraction Poles, Office of the Prime Minister, IUAP-AMS.

References

[1] "Reinventing the wheelchair - autonomous robotic wheelchair projects in europe improve mobility and safety", *issue of IEEE Robotics and Automation Magazine*, vol. 7, no. 1, March 2001.

[2] R. C. Simpson, S. P. Levine, D. A. Bell, L. A. Jaros, Y. Koren, and J. Borenstein, "Navchair: An assistive wheelchair navigation system with automatic adaptation", *Lecture Notes in Artificial Intelligence*, vol. 1458, pp. 235–255, 1998.

[3] H. Yanco, *Shared User-Computer Control of a Robotic Wheelchair System*, PhD thesis, Massachusetts Institute of Technology, September 2000.

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

[5] J. Barraquand, L.E. Kavraki, J.C. Latombe, T.Y. Li, R. Motwani, and P. Raghavan, "A random sampling scheme for path planning", *International Journal of Robotics Research*, vol. 16, no. 6, pp. 759–774, 1997.

[6] A. Stentz, "The focussed D* algorithm for real-time replanning", in *Proc. of the Int. Joint Conference on Artificial Intelligence*, August 1995.

[7] M. Nuttin, E. Demeester, D. Vanhooydonck, and H. Van Brussel, "Shared autonomy for wheel chair control: Attempts to assess the user's autonomy", in *Autonome Mobile Systeme AMS'2001*, Stuttgart, Germany, October 2001, pp. 127–133.

[8] M. Nuttin, D. Vanhooydonck, E. Demeester, and H. Van Brussel, "Selection of suitable human-robot interaction techniques for intelligent wheelchairs", in *11th IEEE Int. Workshop on Robot and Human Interactive Communication ROMAN 2002*, Berlin, Germany, September 25-27, 2002, pp. 146–151.

[9] J. Vandorpe, *Navigation Techniques for the Mobile Robot LiAS*, PhD thesis, Katholieke Universiteit Leuven, January 1997.

[10] Y. Koren, *Computer Control of Manufacturing Systems*, International Student Edition (McGraw-Hill), 1985.