Using Virtual Potential Fields for Electric Wheelchair Guidance

Vicente Díaz, S., Amaya Rodríguez, C., Civit Balcells, A., Díaz del Río, F.
Robotic and Computing for Assistive Technologies Research Team.
Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla.
Avda. Reina Mercedes s/n. 41012. Sevilla. Spain.
satur@atc.us.es

Abstract- **TetraNauta is an electric wheelchair guidance system intended for people with heavy motion impairments (such as persons with tetraplegia), It is specially useful when impairments also affect to wheelchair steering as it is able to automatically guide wheelchairs between different points in an known environment (a hospital, a school, etc), conditioned with track marks painted on the floor. It also provides a semiautomatic navigation mode, where control is shared between user and navigation system. It is intended for learning wheelchair manipulation and as an aid in places where navigation is difficult or dangerous (i.e. for crossing narrow corridors).**

I. INTRODUCTION

Some electric wheelchair users have strong motion impairments restricting the needed movements for driving a wheelchair. Problems may arise driving along prolonged courses: they can get tired as force, accuracy and movement control needed to use the steering device must be maintained for a long time.

Several automatic control systems have been developed to free users from continuously attend wheelchair movement. Some of them have lead to very interesting solutions based on autonomous motion wheelchairs, but with serious drawbacks when porting from research prototypes to commercial wheelchair. An important drawback is their technical complexity, that makes them too expensive to implement in practice [1], [**¡Error! No se encuentra el origen de la referencia.**]. The TetraNauta project [2], [3], [6] is a low cost, fully open steering system to allow people with heavy motor impairments to move in a known environments (hospitals, schools, home, etc.). It minimizes the steering effort, especially in long paths, while attending to user safety. In such environments, most wheelchair motions repeatedly follow a limited number of trajectories along a fairly controlled environment. TetraNauta tries to reduce user efforts by limiting the degrees of freedom in wheelchair motion: it must go over predefined paths following track marks painted on the floor. TetraNauta system can be divided in two parts:

- *Traffic management subsystem.* It is a central computer that communicates with TetraNauta wheelchairs to automatically obtain traffic state (position, motion information, etc.). Moreover, system operator or field sensors can report it some traffic related incidences: temporary restricted areas, closed doors, obstacles, etc.
- *Wheelchair motion control subsystem.* Every TetraNauta wheelchair has an embedded computer, the motion control module, and several functional modules controlling different parts of the wheelchair: power module, steering device module, etc.

This paper describes how the motion subsystem uses Virtual Potential Fields based techniques to automatically guide wheelchair or to share guidance commands with user.

II. MOTION CONTROL OPERATION

The primary goal of TetraNauta system was to automatically move disable people from different points in an known environment: an hospital. But it can also move a wheelchair under direct user control or it even can be used as training platform for novice users. User can switch among operation modes according to his/her needs or skills. There are three operation modes:

- ***Automatic navigation.*** The wheelchair follows predefined paths, so that user (or assistant) just selects the desired destination. The motion control unit performs all the necessary actions to arrive at destination. User cannot directly control any intermediate movement once destination has been given to the wheelchair.
- ***Manual navigation.*** In this mode, TetraNauta behaves like a conventional electric wheelchair, taking all direction and speed commands from user.
- ***Semiautomatic navigation.*** The purpose of this mode is to help user in learning to operate a conventional electric wheelchair. TetraNauta works as a training system, making the painted track marks behave as rails or barriers. It performs an automatic like navigation, but allowing the user to manually interfere on wheelchair movements. The system applies a shared control: users can modify movements generated by automatic navigation with the steering device (i.e. the joystick). Different levels of semiautomatic navigation can be used according to user skill. As practice increases, he/she can switch to a level with more responsiveness to user actions. Semiautomatic navigation can also be used to help user in crossing doors or narrow corridors.

Every TetraNauta wheelchair has a general map of the environment wherein it navigates in automatic mode. It also knows its actual position in the map using on board sensors that detect absolute positioning marks.

User must select the destination point using the user interface to start navigation. Next step is to create a *travel plan*: given an origin and destination locations, the Traffic Management Subsystem find the better path attending to several factors: shortest journey, deadlocks prevention, high traffic avoidance, etc. If wheelchair cannot communicate with the Traffic Management Subsystem, it generates itself the travel plan using the last received traffic state information.

0-7803-7474-6/02/$17.00 ©2002 IEEE

After the Motion Control Subsystem receives the *travel plan*, it executes it making necessary actions to reach destination.

Tasks executed in TetraNauta can be arranged using a multilevel organization common in traditional mobile robot systems [4]:

- **Task planning**. User performs this level in TetraNauta, as he/she always decides destinations.
- **Trajectory planning**. To generate a high level description (graph) of the path to follow is not a complex task, as initial absolute position is known and the system has a complete environment map. It can be done both by the Traffic Management or Motion Control Subsystem.
- **Trajectory generation**. The Motion Control Subsystem must generate trajectory topological descriptions from their graph description. A significant part of this work is already done, as trajectories follow lines and marks painted on the floor. The system performs trajectories by following lines and detecting cross and intersections with a CCD camera at the bottom part of the wheelchair.
- **Robot control**. This level controls the wheelchair power unit to keep the wheelchair moving through the topological trajectory defined by the track marks. Semiautomatic navigation deals with user intentions at this level. In this paper, we propose techniques based on *virtual potential fields* (VPF) to do it.

Next section describes how VPFs can be applied both to automatic and semiautomatic navigations.

III. NAVIGATION

Virtual potential field (VPF) control techniques give us a simple and intuitive way to implement the shared control needed for semiautomatic navigation. It is based on the idea that any painted line generates a virtual potential field that attracts or repels the wheelchair, according to line colour. VPFs generate virtual forces affecting to the wheelchair and guiding its motion. User intentions, taken with the steering device, are treated as one more force by navigation system (supposing that we are operating in semiautomatic navigation). "*User virtual force*" is added to virtual forces generated by VPFs associated to painted track marks. The resulting force contains both user intentions and automatic navigation intentions. In automatic navigation, the only considered virtual forces are those coming from VPFs.

A. The coordinate system.

A state vector $\mathbf{q} = (x, y, \theta)^T$ is used to define the state of any rigid system moving in a plane at time T. The vector defines robot position and orientation. $(x, y) \in \Re^2$ are Cartesian coordinates locating the TetraNauta robot at point q_0 (see Fig. 1). The orientation coordinate, $\theta \in (-\pi, \pi]$, is the robot angle with an external fixed reference system $\Re(O, i, j)$. This state definition has been widely used in literature. Another state definitions can be found in [7], [8], [9].

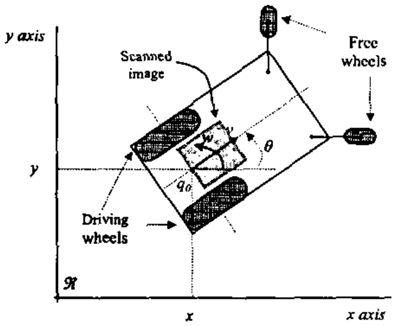

Fig. 1. Wheelchair coordinate system.

Wheelchair scans the X-Y plane (the floor) with a camera attached to its underside (Fig. 1). It is located under user seat, getting fixed size image of the environment where the robot is moving on. The image can contain a part of the guiding line to follow.

Fig. 2 shows a sample images taken by the camera when the robot is over a line. As the camera is close to the floor, a short line section can be scanned but it is enough to make an accurate navigation.

V_R is a new Cartesian coordinate system defined for referencing elements within the image. The x-axis is the axis joining the driving wheels. $\mathbf{q}_0$ is its center and y-axis is perpendicular to x-axis, so that its positive values are toward the front moving direction of the robot.

Points $\mathbf{q}_{ini} = (x_{ini}, y_{ini})^T \in R^2$ and $\mathbf{q}_{dest} = (x_{dest}, y_{dest})^T \in R^2$ are the start and end point of the line to follow, while $d_{\min}$ is the minimum distance between $\mathbf{q}_0$ and L', the detected line section. $\mathbf{q}_{\min} = (x_{\min}, y_{\min})^T \in R^2$ is the point in L' nearest to $\mathbf{q}_0$. Finally, δ is the sloping angle between L' and x-axis, and $e_\delta = \delta - \frac{\pi}{2}$ is the "error angle" with marching direction. With this coordinate system, control techniques used to follow the line should try to make $e_\delta \to 0$, and $d_{\min} \to 0$.

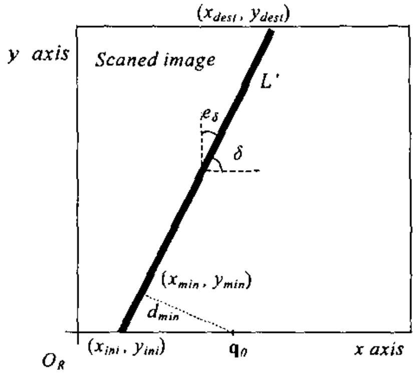

Fig. 2. Sample of images taken by the camera.

B. Guiding TetraNauta with VPFs.

For following a line, at least an AVPF (attractive VPF), with the line as the attractive element, is needed. It affects to robot motion so that line is finally reached. We pretend not just to reach the line, but also to follow it. And additional AVPF is needed, but with a line point as attraction element, attracting the robot toward a destination location in the line.

This seems a good solution for implementing automatic navigation mode of TetraNauta. But semiautomatic navigation also needs lines acting as repulsive elements. So we also need RVPF (repulsive VPF) where line act as a repulsive element.

$U_{att,line}$ and $U_{att,dest}$ are the names of the potential fields attracting to a line and a destination point (the end of the line). In automatic navigation, we combine their effects to get U_{att}, the total VPF influencing the wheelchair. In semiautomatic navigation, an additional type of VPF, U_{rep}, is used for modeling repulsive effects of lines.

In following sections, a generic definition of these VPFs is presented, followed by their adaptation to our coordinated system.

C. AVPF toward the line ($U_{att,line}$).

This VPF attracts the wheelchair toward the line. Two important problems must be solved:

The first is to choose a defining function for the AVPF. The simplest functions of those defined in literature, are parabolic [12] or conic [13] functions. The main benefit of conic functions is that they are constants in all the working space (except at destination point), while parabolic function has excellent stabilization features. We use a combined model [14] mixing parabolic and conic functions. It is the most satisfactory for TetraNauta as once the wheelchair has reach the line, it needs to keep it over the line to make a good tracking. On the other hand, a comfortable navigation, with constant velocity and free of brusqueness, is very important because wheelchair users are disabled persons.

Our combined AVPF function is given by:

$$U_{att,line}(\mathbf{q}) = \begin{cases} (2p_0 k_{p0})p_{line}(\mathbf{q}) - (p_0^2 k_{p0}) & \text{if } p_{line}(\mathbf{q}) \geq p_0 \\ k_{p0} p_{line}^2(\mathbf{q}) & \text{if } p_{line}(\mathbf{q}) < p_0 \end{cases} \quad (1)$$

where $\mathbf{q}$ is the robot location. We shall call union distance to the positive constant p_0. k_{p0} is a positive proportionality constant. Finally, we have to define the function $P_{line}(\mathbf{q})$.

Definition of $P_{line}(\mathbf{q})$ is the second problem to solve. The original definition of a VPF states that a point is the VPF generating element: the drain point. $P(\mathbf{q})$ is defined as the Euclidean distance to between the drain point and $\mathbf{q}$. Now we have not a single drain point, but a drain line. To manage this, $P_{linea}(\mathbf{q})$ is defined as the minimum Euclidian distance between the wheelchair and the line.

The force generated by the AVPF can be computed by:

$$\vec{F}_{att,line}(\mathbf{q}) = \begin{cases} -\dfrac{2k_{p0}p_0(\mathbf{q}-\mathbf{q}_{min})}{p_{line}(\mathbf{q})} & \text{if } p_{line}(\mathbf{q}) \geq p_0 \\ -2k_{p0}(\mathbf{q}-\mathbf{q}_{min}) & \text{if } p_{line}(\mathbf{q}) < p_0 \end{cases} \quad (2)$$

Where $\mathbf{q}_{\min}$ is the nearest point of the line.

D. AVPF toward the destination point ($U_{att,dest}$).

This AVPF is defined so that the destination point generates an attracting force to the wheelchair. The destination point is over the line to follow, and usually matches with the end of its visible segment.

We use a combined AVPF definition function very similar to $U_{att,line}$. It differs only in the target position to reach: it is a single point $\mathbf{q}_{dest}$. The generated force is given by:

$$\vec{F}_{att,dest}(\mathbf{q}) = \begin{cases} -\dfrac{2k_{p1}p_1(\mathbf{q}-\mathbf{q}_{dest})}{p_{dest}(\mathbf{q})} & \text{if } p_{dest}(\mathbf{q}) \geq p_1 \\ -2k_{p1}(\mathbf{q}-\mathbf{q}_{dest}) & \text{if } p_{dest}(\mathbf{q}) < p_1 \end{cases} \quad (3)$$

Where p_1 is again the *union distance* and k_{p1} is a positive proportionality constant. We also define $P_{dest}(\mathbf{q})$ as the Euclidian distance between the wheelchair and the desired final destination.

E. Line generated RVPF ($U_{rep,line}$)

A line generating a RVPF may have two different configurations:

- It may be *completely repulsing*, creating an impassable obstruction to the wheelchair, so that it can only walk around it.
- If the line is defined as *partially repulsing* the wheelchair can cross it, but feeling some resistance. The degree of resistance can be also configured.

We ought to use a computationally inexpensive RVPF function expressing both partially and completely repulsion. A force function [6] meeting these requirements is:

$$\vec{F}_{rep}(\mathbf{q}) = \begin{cases} 4A\dfrac{(\mathbf{q}-\mathbf{q}_{\min})}{p_0^2} & \text{if } p_{obs}(\mathbf{q}) \leq \dfrac{p_0}{2} \\ \dfrac{4A(\mathbf{q}-\mathbf{q}_{\min})}{p_0}\left(\dfrac{1}{p_{obs}(\mathbf{q})} - \dfrac{1}{p_0}\right) & \text{if } \dfrac{p_0}{2} < p_{obs}(\mathbf{q}) \leq p_0 \\ 0 & \text{if } p_{obs}(\mathbf{q}) > p_0 \end{cases} \quad (4)$$

Where $\mathbf{q}_{\min}$ is the robot's nearest obstacle point, p_0 is the obstacle (line) influence distance and A is the obstacle maximum repulsion potential. Moreover, $p_{obs}(\mathbf{q})$ is the minimum distance from the robot to the line.

IV. AUTOMATIC NAVIGATION IN TETRANAUTA

In *automatic navigation*, the user just tells the final destination to system and it makes all required movements to reach it. Wheelchair control must be completely automatic, avoiding any user contribution to its local movements[1] (apart from stopping to change a new destination or select a new navigation mode).

In TetraNauta, this kind of navigation uses local sensing of lines painted on the floor. The sensing system is based in a low cost CCD color camera.

Analyses of camera images get topological parameters used in VPF functions for the guiding system.

Fig. 3 shows a sample situation to demonstrate how the guiding system uses VPF, and Fig. 2 could be one of the scanned $n \times m\, cm^2$ images, being n and n the x and y-axis dimensions, respectively.

Image processing would detect a line section between points $\mathbf{q}_{ini} = (x_{ini}, y_{ini})$ and $\mathbf{q}_{dest} = (x_{dest}, y_{dest})$, and slope error[2], e_δ.

$\mathbf{q}_0$ stands for the wheelchair positions, but we shall call it $\mathbf{q}$ to maintain notation, so that $\mathbf{q} = \left(\frac{n}{2}, 0\right)^T$ and d_{min} is the minimum distance between $\mathbf{q}$ and line L. With these inputs, our purposes are that $\mathbf{q} \to \mathbf{q}_{dest}$, $d_{min} \to 0$, and $e_\delta \to 0$.

While the camera captures only a line section, we see that this segment maintains nearly the same position as we navigate toward $\mathbf{q}_{dest}$. The same effect occurs if the wheelchair is stopped above the line. If control works fine, succeeding captured images are very similar, except at the end of the line, when we will really see $\mathbf{q}$ approaching to $\mathbf{q}_{dest}$ (Fig. 3)

So, it is important both to try to arrive at $\mathbf{q}_{dest}$, that is equivalent to impose a path following[1] movement [10], [11], and try to force $d_{min} \to 0$ and $e_\delta \to 0$, compelling the wheelchair to go on over the line following the desired path aligned with line direction.

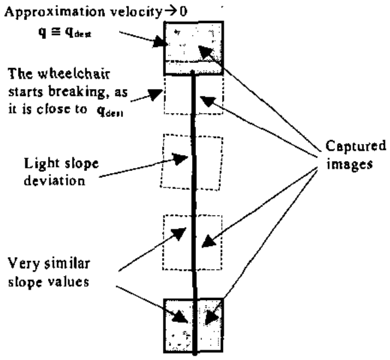

Fig. 3. Sample of a sequence of scanned images in automatic navigation.

We combine effects of two attractive VPF to meet both objectives: a line AVPV makes the wheelchair to move toward the line ($d_{min} \to 0$), while an point AVPF moves it toward the end of the line ($\mathbf{q} \to \mathbf{q}_{dest}$). The global effect of both VPFs is that wheelchair moves towards the desired location, not in any way but following the line, because the line AVPF has not effect once the wheelchair is aligned with the line.

AVPFs defined in (1) and applied in (3) can be used to get the virtual force applied to the wheelchair, as:

$$U_{att}(\mathbf{q}) = U_{att,line}(\mathbf{q}) + U_{att,dest}(\mathbf{q}) \qquad (5)$$

And,

$$\vec{F}_{att}(\mathbf{q}) = \vec{F}_{att,line}(\mathbf{q}) + \vec{F}_{att,dest}(\mathbf{q}) \qquad (6)$$

Their defining functions (2) and (3) are continuous, positive, and derivable, so the resulting AVPF keeps its components properties. On the other hand, the AVPF has an absolute minimum in $\mathbf{q}_{dest}$ ($U_{att}(\mathbf{q}_{dest}) = 0$), because both $U_{att,dest}(\mathbf{q}_{dest}) = 0$, and $U_{att,line}(\mathbf{q}) = 0$, $\forall \mathbf{q}$ inside the line with the same slopping, including $\mathbf{q}_{dest}$. The force due to the total AVPF also has good stabilization features because it converges to 0 when the robot approaches to $\mathbf{q}_{dest}$.

Fig. 4. shows the AVPF shape in a $50 \times 50\, cm^2$ zone, when (5) is used in a line defined by points $\mathbf{q}_{ini} = (25, 49)^T$ and $\mathbf{q}_{dest} = (25, 0)^T$. Parameters are $k_{p0}, k_{p1} = 0.1\, v.p.u./cm^2$ and $p_0, p_1 = 10\, cm$.

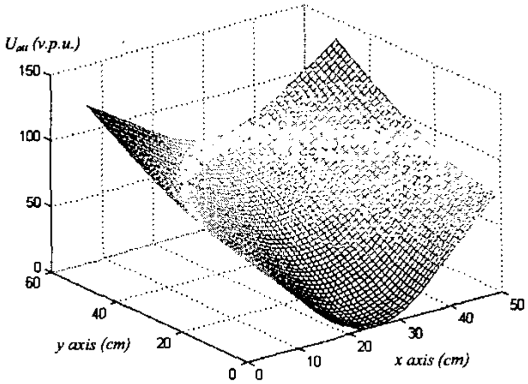

Fig. 4. Attractive virtual potential field used for automatic navigation. Potential values are in virtual potential units (v.p.u.).

V. SEMIAUTOMATIC NAVIGATION.

In semiautomatic navigation painted marks behave like rails or barriers. It is useful when user operates wheelchair for the very first time because a wheelchair operation is an unusual task he/she needs to learn. Moreover, learning process could be difficult if user skills are restricted by some kind of disable.

In this mode, user and navigation system share wheelchair guidance. Navigation system is also based in VPFs generated virtual forces that affects to movements. A simple and natural way to treat user intentions, taken with

---

[1] That is, all movements needed to follow the user selected route.

[2] The camera is lined up with the wheelchair, so that slope error is zero when it is parallel to the line.

some input device[3], is to generate with then an additional "virtual force" in the navigation system. All virtual forces are adding up to drive wheelchair movement including both user intentions and automatic navigation.

A standard joystick can be used to compute user virtual force. We can read and computed the x/y joystick position anytime using the method illustrated by Fig. 5.

Position is within a circle or radius r, so that we consider that its position defines a force vector $\vec{F}_u$, (appropriately pondered) produced by user intention.

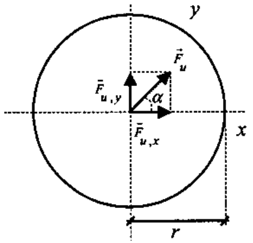

Fig. 5. Joystick can be moved up to a distance of r from its released position.

The total virtual force applied to wheelchair in semiautomatic mode is given by

$$\vec{F}_{SemiNav} = \vec{F}_{usser}(\mathbf{q}) + \vec{F}_{system}(\mathbf{q}) \tag{7}$$

where

$$\vec{F}_{user}(\mathbf{q}) = \vec{g}\vec{F}_u(\mathbf{q}) \tag{8}$$

$\vec{g}$ is an user adapted function: It could be modify depending on user skill level, route complexity, etc. We can use, for simplicity, the following linear function:

$$\vec{F}_{user}(\mathbf{q}) = k_u \vec{F}_u(\mathbf{q}) \tag{9}$$

$\vec{F}_{system}(\mathbf{q})$ is defined by the navigation submode. Two have been defined for semiautomatic navigation: *rail submode* and *learning submode*. User can change submode anytime when the system is in semiautomatic navigation

A. Rail submode.

It starts working once the wheelchair is on a main line[4]. The line attracts it aligning the wheelchair to its direction. Then the line acts like a rail: user is able to go forward or backward, or change velocity, but he/she cannot leave the line.Virtual force due to navigation system keeps wheelchair on the line:

$$\vec{F}_{system}(\mathbf{q}) = \vec{F}_{att,line}(\mathbf{q}) \tag{10}$$

Where $\vec{F}_{att,line}(\mathbf{q})$ is given by (2), with its parameters customized for this submode.

User virtual force has been defined so that user only generates forward forces. This is intended for security, keeping in mind that TetraNauta users usually have strong impairment. It is difficult for them to drive forward, and impossible, in many cases, to safety reverse drive. TetraNauta implements automatic π radians turns allowing users to easily to drive back.

User virtual force is compute from the y-axis projection of joystick position (see Fig. 5) and the orientation of the scanned line section. So, its module is given by:

$$\left|\vec{F}_{user}(\mathbf{q})\right| = \begin{cases} k_u \left|\vec{F}_{u,y}(\mathbf{q})\right| & if\ \vec{F}_{u,y}(\mathbf{q}) > 0 \\ 0 & if\ \vec{F}_{u,y}(\mathbf{q}) \le 0 \end{cases} \tag{11}$$

The user virtual force angle is equal to the scanned line inclination (Fig. 2), so that it has the same direction that navigation system force:

$$Arg\left(\vec{F}_{user}(\mathbf{q})\right) = \delta \tag{12}$$

The user can navigate on any line controlling velocity. He/She uses imput device to command wheelchair for turning back or choosing direction when it meets a line fork. Wheelchair operation can be tiring in long courses because user has to keep joystick position toward marching direction to keep wheelchair moving. A minor variation of rail submode, *sloped rail submode*, is provided to alleviate user from this. In *sloped rail submode*, user only needs joystick to choose direction in fork options, to turn back or to stop motion, while system keeps the wheelchair moving. Virtual force is given by:

$$\vec{F}_{SemiNav}(\mathbf{q}) = \vec{F}_{System}(\mathbf{q}) = \vec{F}_{att,line}(\mathbf{q}) + \vec{F}_{att,dest}(\mathbf{q}) \tag{13}$$

It is the expression used by automatic navigation (6), as both modes have identical behavior, apart from decisions on trajectory in forks or turns.

B. Learning submode.

Learning mode is primary intended to use TetraNauta as a training platform, but it also helps wheelchair steering in some circumstances (crossing narrow doors, corridors, etc.).

The training environment contains attracting and/or repulsing lines (painted with different colors). The lines modify user commands to help wheelchair control. Virtual force is given by:

$$\vec{F}_{learning}(\mathbf{q}) = \vec{F}_{att}(\mathbf{q}) + \vec{F}_{rep}(\mathbf{q}) + \vec{F}_{user}(\mathbf{q}) \tag{14}$$

where $\vec{F}_{rep}(\mathbf{q})$ and $\vec{F}_{user}(\mathbf{q})$ are given by (4) and (9), respectively, and

$$\vec{F}_{att}(\mathbf{q}) = \vec{F}_{att,line}(\mathbf{q}) + \vec{F}_{att,dest}(\mathbf{q}) \tag{15}$$

---

[3] Joystick, *mouthstick*, *chinstick*, etc.

[4] TetraNauta implements line searching algorithms to ease this operation.

$\vec{F}_{att,dest}(\mathbf{q})$ does not exist if the line is configured as a rail. There are four different types of line defined for this submode[5]:

- *Color 1.* Repulsive line. Repulsion level depends on user skill. The maximum level is used for novel users, becoming impassable lines. Repulsion level decreases as user improves. Repulsive lines can be used to make virtual corridors (Fig. 6).
- *Color 2.* Impassable repulsive line. It is used to disallow user to leave training environment.
- *Color 3.* Attractive line. The AVPF acts like a virtual valley along the line. It has no effect on user force if he/she directs joystick parallel to line. If joystick is directed toward line, system boosts user force effects (the wheelchair is "lowering the valley") while system weakens user force effects (the wheelchair is "rising from the valley") if joystick is directed off the line. Attraction level also depends on user skills. These lines are useful for crossing doors or corridors (Fig. 6).
- *Color 4.* Rail line. Wheelchair behaves line rail submode. It is also used for crossing narrow places.

If system does not detect any line it behaves like a standard electric wheelchair. So, user can activate learning mode as default operation mode and directly guide wheelchair movements except in dangerous or difficult zones (unevenness, stairs immediacy, doors, etc), where marks can help navigation.

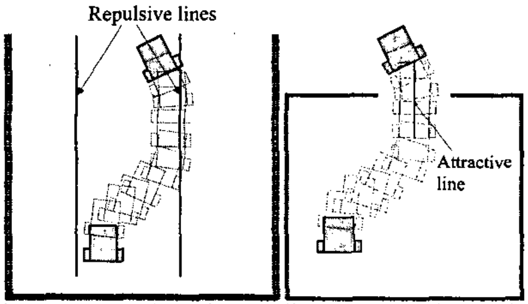

Fig. 6. Attractive and repulsive lines effect in semiautomatic navigation

CONCLUSIONS

This paper have presented Virtual Potential Fields as a good tool for implementing automatic guidance in the in the intelligent electric wheelchair area, and these can extent to the teleoperated mobile robots area, in general. Virtual Potential Fields are intuitive and easy to use. It has been shown that small software variations can greatly increase system functionality, making an automatic navigation system into a training platform for wheelchair operation.

REFERENCES

[1] G. Bourhis, P. Pino. "Mobile Robotics and Mobility Assistance for People with Motor Impairments: Rational Justification for the VAHM Project". *In: IEEE Trans. on Rehabil. Engin.* Vol. 4, No. 1, March 1996.

[2] A. Civit, J. Abascal. *"TetraNauta: A Wheelchair Controller for Users with Very Severe Mobility Restrictions". Improving the Quality of Life for the European Citizen,* I. Placencia, E. Ballabio (Eds.). ISBN: 1383-813X. IOS Press, Netherlands, (1998). Pg. 336-341.

[3] A. Civit, F. Díaz, S. Vicente, C. Amaya, G. Jiménez, J.L. Sevillano. "Arquitectura de controladores de sillas de ruedas". *Novática, Monografía: informática y discapacidades.* Pg. 16-20. Nov-Dec 1998.

[4] I.J. Cox. "Blanche- An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle". *IEEE Transactions on Robotics and Automation.* Vol. 7, Nº2, April 1991.

[5] R. Gelin, J.M. Detriche. "The sprint of COACH". *IEEE International Conference on System, Man and Cybernetic.* Le Touquet, Fance, 1993.

[6] S. Vicente Díaz. "Una aportación al guiado de sillas de ruedas eléctricas en entornos estructurados". *Ph. D. Thesis.* Universidad de Sevilla, July 2001.

[7] Y. Kanayama, Y. Kimura, F. Miyazaki, T. Noguchi. "A Stable Tracking Control Method for an Autonomous Mobile Robot". *Proceedings 1990 IEEE International Conf. Robotics and Automation,* Cinccinatti, Ohio, Pág. 384-389.

[8] C. Samson. "Path Following And Time-Varying Feedback Stabilization Of A Wheeled Mobile Robot", *Proceedings International Conference ICARCV'92,* Singapur RO-13.1. 1992.

[9] A. Micaelli, C. Samson. "Trajectory Tracking for Unicycle-Type And Two-Steering-Wheels Mobile Robots". *Informe interno Nº 2097, INRIA: Programme 4: Robotique, Image et Vision.* Centre de Sophia-Antipolis, Francia. Noviembre 1993.

[10] A. De Luca, G. Oriolo, C. Samson. *"Feedback Control of a Nonholonomic Car-Like Robot".* Chapter 4 of Planning Robot Motion, J.-P. Laumond Ed., Springer-Verlag, 1997.

[11] F. Díaz, G. Jiménez, J.L. Sevillano, S. Vicente, A. Civit. "A Path Following Control for Unicycle Robots". *Journal of Robotic Systems,* 18(7), 525-342 (2001).

[12] O. Khatib. "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots". *International Journal of Robotics Research.* Vol. 5, Nº1, Primavera 1986, Pág. 90-98.

[13] J.C. Latombe. *"Robot Motion Planning".* Kluwer Academic, Boston, 1991.

[14] J.R. Andrews, N. Hogan. *"Impedance control as a framework for implementing obstacle avoidance in a manipulator".* Editores: D.E. Hardt y W.J. Book, Control Of Manufacturing Processes And Robotic Systems, ASME, 1983. Pág. 243-251.

---

[5] In TetraNauta prototype, colors are green red, blue and black, but they can be changed in the image processing module to adapt them to floor background color.