The Shared Control Dynamic Window Approach for Non-Holonomic Semi-Autonomous Robots

Pablo, Iñigo-Blasco, Robotics and Computer Technology Lab, Univ. of Seville, pabloinigo@atc.us.es, Spain
Fernando, Díaz-del-Río, Robotics and Computer Technology Lab, Univ. of Seville, fdiaz@atc.us.es, Spain
Saturnino Vicente Díaz, Robotics and Computer Technology Lab, Univ. of Seville, satur@atc.us.es, Spain
Daniel Cagigas Muñiz, Robotics and Computer Technology Lab, Univ. of Seville, dcagigas@atc.us.es, Spain

Abstract

A shared control method, called *Shared Control Dynamic Window Approach,* is presented. It is inspired in the Dynamic Window Approach (DWA) for autonomous mobile robots. It takes user commands by means of the control interface and provide the most suitable and kinodinamically feasible trajectory that guarantees obstacle avoidance. It provides navigation assistance to drive vehicles in unstructured environments and other scenarios where dynamic constraints play an important role . In order to keep an intuitive control for the user, the intrusiveness of the method varies gradually and proportionally with the danger of collision. Preliminary experiments with users driving vehicles in a simulated world validate the method. Its implementation is public under General Public License.

1 Introduction

Driving a vehicle in unstructured environments may be a challenging task for a human pilot. The difficulty and dangerousness increases when the driving has to be performed at high speed, the vehicle has non-holonomic constraints or it does not have a fast dynamic response. Although a computer can provide some high navigation capabilities in terms of precision, perception and reaction time, a human mind can perform better complex reasoning and can interpret properly complex scenarios, for instance predicting other drivers' intentions. Shared control methods arise from the need of improving vehicle navigation but following the guidelines of a human pilot through user interfaces such as joysticks.

Some navigation assistance methods have been developed by Car industry, such as: forward collision detection, automatic braking, lane guidance systems, etc. These methods do not improves the capacity of avoiding obstacles. On the contrary, shared control is a kind of navigation assistance that have been mainly investigated in the field of intelligent wheelchairs. It can help handicapped users colliding with obstacles, causing damages to themselves, to other people, to the vehicle or to the environment. Advantages of shared control methods go beyond the assistance of handicapped people. They can be also useful for driving kinodynamic constrained and complex-driving vehicles, like cars at high speeds. Moreover, they may improve the learning of driving since novel human pilots typically need a long time to do it [1].

This work introduces a shared control method to control a kinodinamically constrained ground vehicle by means of a joystick or other Local Control Interface (LCI). The need for shared control in wheelchairs and the experience of our Lab in this field [2], [3] have inspired our present proposal. It enables unexperienced or handicapped pilots to drive vehicles safely in challenging scenarios. The resulting method is specially interesting when it is applied to vehicles moving at high speed and with a slow dynamic response (see 1).

Paper is organized as follows: section 2 introduces some of the previous works in this field. Section 3 states a general approach about how reactive local planners can be used as shared-control systems and summarizes classic DWA method whereas 4 discuses different collision metrics. Later, in section 5 our method is described. Implementation and experimental validation are presented in section 6, while conclusions in section 7.

2 Previous work

Works [4]–[6] survey the most important shared control methods for wheelchairs, which can be classified in three main approaches: the “goal prediction based” and the “behavior based” ones and the “continuous shared control”. The first one is based on guessing where the user wants to go. This concept is commonly known as “prediction of intent”. “Behavior based” approaches use a set of navigation behaviors (ie: manual mode and autonomous mode) that are activated in different contexts (crossing a door, navigating a corridor, etc.). Finally, in “continuous shared control” approaches, the desired trajectory given by the LCI is combined with other obstacle avoidance criteria. This approach gives the user a direct influence in the performed trajectory and not only in the a final position objective.

The use of goal predictors enables any automatic navigation method to be used as a shared control method. The desired trajectory can be guess using landmarks such as artificial rails on the ground [7], [8]. However, this has two main drawbacks: the loss of autonomy that the user suffers, and the need of inferring the final user goal through a programmatic system. This implies that the performed trajectory may be completely different to what

ISBN 978-3-8007-3601-0

the user expected. Conversely, according to some studies [9], navigation assistance should appear (in a gradual an continuous way) only when it is needed, and should be the less intrusive as possible (wheelchair users are reluctant to lose the control of his/her wheelchair)

Behavior based approaches such as [10][11] alternates autonomous navigation methods with manual control. The main drawback of these approaches is that the automatic change of navigation modes may produce confusion to the user. A non clear switching between the autonomous mode and the manual one may even produce an erratic steering. Once again, the assistance should be predictable and appear in a gradual and seamlessly manner.

Other approaches such as [12] proposes the use of an elastic bands planner. This method enables the user to introduce small deviations of the global trajectory that is being performed by means of the LCI, which can be considered a combination of the “goal prediction” and the “continuous shared control” approaches.

Other navigation assistance approaches are based on the Reactive Navigation Paradigm. These methods compute the movement of the vehicle from the LCI commands and the position of the obstacles. In this sense, they can be considered as “goal prediction based” approaches. Methods [5], [13]–[15] are inspired on the potential field paradigm, whereas [5] proposes that the LCI generates an artificial short-term goal and obstacles produce repulsive forces.

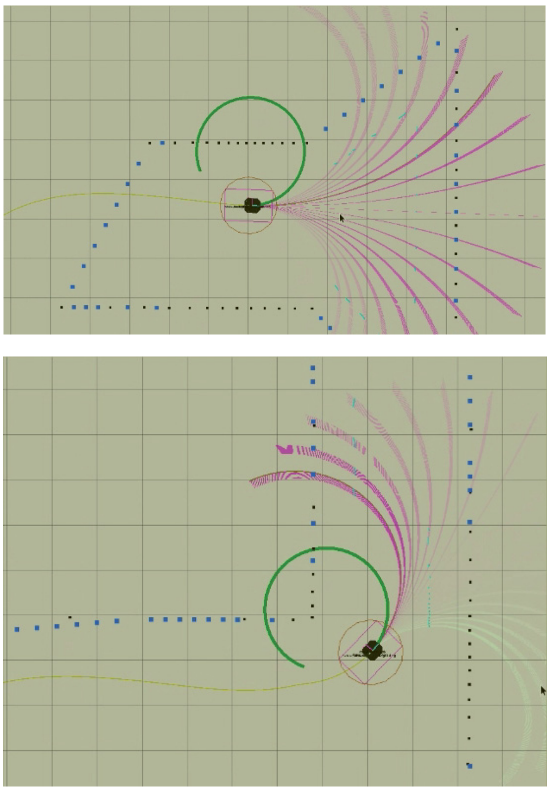

Figure 1. Two captures of Shared Control working. Thick circular arcs indicate estimated user intentions. The rest of arcs are kinodynamycally feasible trajectories, from which Shared-DWA method selects the best trajectory. An important drawback of these methods is that they do not usually take into account the non-holonomic vehicle constraints. This can produce a collision when the user copes some challenges like moving close to a wall or trying to cross a door.

Recently, methods [16], [17] takes into account the kinematic constraints, so they follow the “continuous shared control” approach. However, these methods are only suitable for slow movements (like backwards maneuvering), since vehicle dynamics is neglected.

The other field that has inspired our work is Intelligent Cars. Works [18][19] survey an important part of the current research work on navigation assistance for intelligent cars. A promising recent work [20] proposes a shared control method for intelligent cars based on the geometric computation of path homotopies from a Voronoi diagram (computed from the input obstacles). However, the computation of the homotopy does not take into account explicitly the kinodynamic constraints of the vehicle and, additionally, it need a previously known final goal position.

3 Motion planning and the DWA method

The Shared-DWA method is based on ideas from the wheelchair assistance navigation field and from vehicle assisted driving [21], [22]. The consideration of the kinodynamic constraints of the vehicle is one of the novelty features that it provides. Thus, the “shared control problem” is a particular case of the “obstacle avoidance problem”, that is, the concept of goal position in the workspace must be generalized to a goal trajectory in the action space. This enables the conversion of many local planners in shared control methods by introducing the required modifications. In order to illustrate the proposed idea, the DWA (Dynamic Window Approach) method has been chosen to be redefined as a shared control system, although it is possible to redefine almost any other obstacle avoidance methods using our approach.

Let us define the obstacle avoidance problem statement like this: Let R be a robot that moves in a workspace W with a state $x \in X$ and whose dynamic behavior is defined by $\dot{x} = F(x,u)$, where $u \in U$ is the applied action. The set of known or sensed obstacles is given by $o \subset W$, and the mission goal is $g \in G$. A motion planner takes the state, the goal and the obstacles and provide the output, that is, the set of actions that the robot should follow in order to reach the goal. The goal space G has been intentionally defined in an abstract way because many representations are possible.

Conversely to most autonomous obstacle avoidance methods, which typically considers the goal like a position in the workspace, the goal position is unknown in a LCI-based shared control system. Each LCI command is only a partial view of user's intention while the final goal position is only in the pilot's mind. The sequence of LCI commands represents conceptually a

ISBN 978-3-8007-3601-0

flow of desired actions that the user has thought as a way of reaching the goal position (user acts as a global planner). From authors point of view, the goal of a shared control system should be a commanded trajectory instead of a goal position. It must be noticed that this general approach may be considered a formalization of the "continuous shared control" approaches (see section 2).

The *Dynamic Window Approach* (DWA) is a reactive planner that performs a search of an optimal action in a subset V_d of the action space called "The Dynamic Window" (see 2). V_d is the subset of dynamically possible speeds $V_d \subset [v_{min}, v_{max}] \times [\omega_{min}, \omega_{max}]$, given the current state $x \in X$ and the action bounds, which are linear and angular acceleration limits of the dynamical system: $[a_{min}, a_{max}]$ and $[\alpha_{min}, \alpha_{max}]$. Note that a_{min} is the maximum braking capacity while a_{max} is the maximum acceleration. By convenience, this window is assumed to be symmetric around its center (in practice it does not). Hence V_d bounds depend on $[a_{min}, a_{max}]$, $[\alpha_{min}, \alpha_{max}]$, and on the sampling period δt. For example the upper bound for linear speed in V_d would be: $max(v + a_{max} \delta t, v_{max})$. The subset V_d is sampled in a regular grid fashion, which gives a discrete set of actions $\check{V}_d$. A filter to discard non-admissible actions is applied to $\check{V}_d$, which result in a subset of admissible actions $\check{V}_a \subset \check{V}_d$. An action $(v_{ij}, \omega_{ij}) \in \check{V}_d$ is admissible if it is possible to perform an emergency braking along the circular trajectory defined by (v_{ij}, ω_{ij}) without colliding with any obstacle.

Original DWA defines goals as points in the workspace, $g \in G \equiv W$. For a goal and a set of obstacles $o \subset W$, DWA finds the best action $(v_{best}, \omega_{best}) \in \check{V}_a$ as the one that minimizes the aggregated cost function G, which is defined as:

$$G(v, \omega, o, g) = k_{clearance} \cdot clearance(v, \omega, o) + k_{velocity} \cdot vel(v) + k_{heading} \cdot heading(v, \omega, g)$$

Function G depends on 3 components, weighted with the free parameters $k_{velocity}, k_{clearance}, k_{heading}$. The $clearance(v, \omega, o)$ function describes and evaluates the distance to the obstacles. The $vel(v)$ function evaluates the cost on the speed. In original DWA method, faster speed commands were preferred against slower ones, while other variants are possible. The $heading(g, v, \omega)$ function quantifies how well the vehicle will be orientated to the goal.

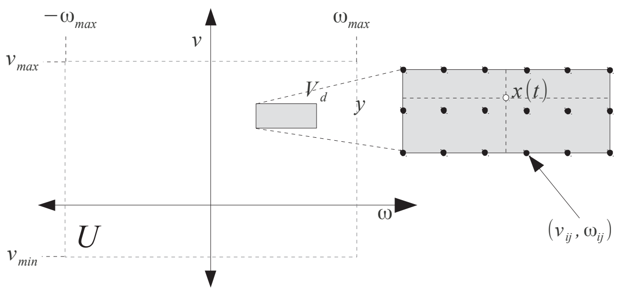

Figure 2: Main concept of the DWA algorithm. A dynamic window V_d is a subset of the kinematically possible action space U. Sampled actions in V_d represent a set of kinodynamically feasible trajectories of the robot in the Workspace W.

4 Distance to collision metrics for obstacle avoidance

In order to avoid collisions, DWA uses two mechanisms: the Clearance Cost Function (CCF) and the Non-Admissibility Filter (NAF). From the authors point of view, the capacity of obstacle avoidance of such mechanisms may worsen in some typical scenarios. Due to this, we propose a novel distance to collision metric to improve the reliability of the obstacle avoidance task in the DWA and in the Shared-DWA.

CCF in DWA has two main limitations: first, it only considers the distance to the obstacle along a circular trajectory. Second, it neglects the dynamic constraints of the vehicle. Thus CCF may not describe properly the danger of collision of the circular trajectory. An example of the CCF limitations can be seen in pure turn trajectories for shaped robots. Here, the clearance cost got is always zero neglecting the possible collisions of this movement. Because of these limitations, the main responsibility in the obstacle avoidance falls on NAF. NAF makes DWA algorithm totally safe in theory. However, it is not a continuous metric, but a discrete one (of just two opposite values). Therefore, small errors when sensing the state or the obstacles may convert an inadmissible trajectory to an admissible one and viceversa. Due to this, sometimes the best fitted action computed by DWA is recklessly similar to a non-admissible action in the same or in a similar circular path. In our experiments with a real robot into intricate zones, robot usually fell into non-admissible states that sometimes ended up in collisions.

To sum up, the question is: What should a distance to collision metric consider? A general answer could be to compute how far the current state is from an inevitable collision state [23]. However, computing the distance to inevitable collision states may not be affordable in terms of computational complexity. This is because of the computational cost of exploring all the possible future trajectories. During last decades, most of the collision to distance metrics have assumed a simplification: the vehicle would follow a known path in the near future.

Classical methods such as "Virtual Force Field" (VFF) [24] or "Vector Field Histogram" (VHF) [25] used the linear euclidean distance in the workspace. DWA [26] and Curvature Velocity Method (CVM) [27] generalized this idea to measure the distance along circular paths. The "Ego-Kinematic Space" [28] combined linear and angular distances to obstacles along circular admissible trajectories. The "Trajectory Parameter-Space" [29] proposed a coherent metric in the C-Space for arbitrary shaped paths, but it does not consider vehicle dynamic constraints.

Bearing in mind previous discussion, we propose a distance to collision metric that takes into account the dynamic constraints of the vehicle and considers both angular and linear distances to the closer obstacle.

ISBN 978-3-8007-3601-0

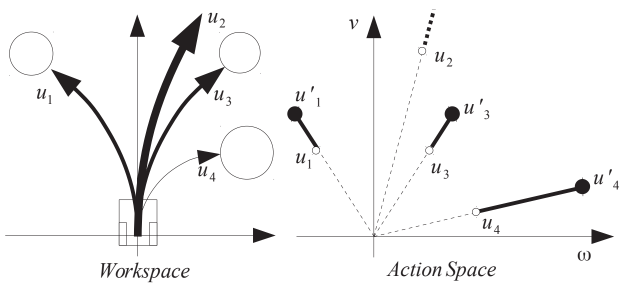

Figure 3: Candidate actions u_i have different distance cost depending on: the distance to the obstacle, the linear and angular speeds of the action, and the dynamic limits of the vehicle. Left image shows a set of candidate actions in the workspace (arrow width is proportional to their linear speeds). Right image shows these candidate actions (empty blobs) in the action space, and the nearest non-admissible actions u'_i (black blobs) with the same curvature.

This metric answers to the following question: "What is the distance d_U between the candidate action u to the nearest inevitable collision action u_{inev} along the same circular path?". That is, $d_U(u)=\|u-u_{inev}\|$.

Two challenges appear: computing the inevitable collision action for the same circular path, and using a suitable distance metric in the action space.

The nearest inevitable collision trajectory $u_{inev}=(v_{inev},\omega_{inev})$ can be easily computed by means of the classic equations of the uniform acceleration motion (as NAF does). Let a_{min} be the linear maximum deceleration, d_{linear} the arc distance to the obstacle, α_{min} the angular deceleration brake and $d_{angular}$ the orientation distance to the obstacle (along the circular path). Thus, the maximum speed such that the vehicle can do an emergency brake will be given by one of these two braking formulas: $v_{inev}=\sqrt{2\cdot a_{min}\cdot d_{linear}}$, $\omega_{inev}=\sqrt{2\cdot\alpha_{min}\cdot d_{angular}}$.

The distance metric of the action space has to cope with the problem of the different units of the linear ($m\cdot s^{-1}$) and angular velocities ($rad\cdot s^{-1}$). A proposal is a conservative norm using the worst case between the linear and the angular non-admissibility degrees[1]:

$$d_{\mathcal{U}}(v,\omega)=\begin{cases}0 & \text{if } v\geq v_{inev}\vee\omega\geq\omega_{inev}\\ 1 & \text{if } v=0\wedge\omega=0\\ min\left(\dfrac{v_{inev}-v}{v_{inev}},\dfrac{\omega_{inev}-\omega}{\omega_{inev}}\right) & \text{otherwise}\end{cases}$$

One of the characteristic of this approach is that the distance values are normalized in [0,1]. The distance value of any feasible action is always lower than 1. Figure 3 illustrates this idea. The danger of collision of a trajectory depends on the distance to the obstacle and also on the vehicle kinodynamic constraints. For instance, u_4 would be a safer action than u_3 despite that its linear distance is lower if the vehicle were able to execute high angular decelerations.

[1]This expression only considers forward movements (backwards movement can be obtained with the same procedure).

5 The Shared-DWA method

In order to find the best kinodynamically feasible action, Shared-DWA method tries to follow the commanded goal trajectory from the LCI while guaranteeing avoidance of obstacles (see 1). Thus, DWA cost functions must be redefined.

In order to fix ideas, let us consider a simple dynamical model of a planar wheeled mobile robot, where $x\equiv(x_1,x_2,\varphi)$ represents the Cartesian coordinates and the orientation w.r.t. x_1 axis. Action $u\equiv(v,\omega)$ is composed of the linear and angular speeds. Let $j=(j_x,j_y)\in J$ be an LCI command, which is mapped to the target circular path (instead of as a goal position in the workspace), that is, there is a function: $f_{mapping}(j,x)\rightarrow g\equiv(v_g,\omega_g)$. Replacement of goal space G makes Shared-DWA look much more a velocity control system. Conversely, the original DWA statement was a position control system .

The Shared-DWA design follows the next two principles:

- In non dangerous scenarios, users should have full control of the system like in manual navigation. Due to this, the selected actions in $(v,\omega)\in V_d$ must be the closer to the goal action $g=(v_g,\omega_g)$. This minimizes the intrusiveness of the system according to the principles explained at [30] .
- If a collision may occur, the navigation assistance degree must be increased.

Admissibility cost plays the role of CCF and NAF in DWA. It regulates the influence of the velocity heading functions according to their safety. It can be defined as the complementary of previous d_U, that is:

$$Adm(v,\omega)=(1-d_U(v,\omega))$$

The aggregated cost function must be different from that of DWA, because the non admissibility function is not required. The admissibility factor serves to regulate the intrusiveness of the automatic control. If the action is safe, the cost is only related with the previous explained factor. Otherwise the cost get increased until reaching 1 when the action is non-admissible. To sum up:

$$cost=Adm+(1-Adm)(k_{head}\,heading_{cost}+k_{vel}\,velocity_{cost})$$

The heading function describes the desired direction that the vehicle should take at the original DWA. In Shared-DWA, the heading function measures how well the curvature fits the target curvature. Hence, the heading function is reformulated according to:

$$heading_{cost}(v,\omega,v_g,\omega_g)=\frac{|atan2(v,\omega)-atan2(v_g,\omega_g)|}{\pi}$$

The velocity cost function for Shared-DWA describes how an action fits the desired velocity. This is evaluated as the linear error between the candidate v and the target velocity v_g normalized by the maximum possible velocity v_{max}:

$$vel_{cost}(v,v_g)=\frac{|v-v_g|}{v_{max}}$$

When the speed of the vehicle increases, the non-admissibility degree for most of the actions get increased. In this situation, it is possible to reach a state where almost all the actions have cost 1, which can be called "non-admissibility saturation". Approaching this state is not recommendable because it would limit too

ISBN 978-3-8007-3601-0

much the user movement freedom. This situation is common when running fast in a corridor. If it happened, it would be very difficult to do a 90º turn.
In order to have a rich set of candidate actions, experience has shown us that linear speed must be decreased when approaching this situation. This can be done by introducing an extra regulation to the velocity, like this:

$$vel(v, v_g) = (1-D)\frac{|v-v_g|}{v_{max}} + D\frac{|v|}{v_{max}}$$

D is a global dangerousness metric of the current navigation context. It can be defined like a reduction function for any speed in $\check{V}_d$:

$$D = \sum_{(v,\omega)\in \check{V}_d} \frac{clearance(v,\omega)}{count(\check{V}_d)}$$

When global dangerousness is high, system intrusiveness is increased. With this regulation the non-admissibility saturation is not reached and the admissible candidate actions hold diverse enough.

6 Implementation and Experimental validation

Shared-DWA implementation is freely available for the community as open source under the terms of the GPL license. The developed system support frequencies higher than 50Hz in a Intel Core I7 2GHz processor with a resolution of the dynamic window of 20×15. The method uses *Continuous Collision Detection (CCD)* in order to compute the non-admissibility degree. The implementation can consider robot shape using precomputed lookup tables or a circular robot shape computing the collision at real time. Due that sampling frequency of the laser ranger is 20 Hz (with 120 samples), both methods can be efficiently computed in 0.05 s. During the execution, user gets force feedback with the factor D described in the previous section. This gives the user information about the global danger of the current state and encourages him/her to decrease the speed.
Test have been done using the simulator “Stage” [31]. This simulator has been extended to support dynamic constraints (angular and linear acceleration limits) .
This software is built on top of the following open source software: ROS (http://www.ros.org) and The Robot Navigation Template Library (RNTL) (https://sites.google.com/site/robotnavigationtemplatelibrary/) developed by the authors.
Some experiments that validate Shared-DWA algorithm are summarized below. Four human users tested the system in the simulated environment. The simulator experiments have been performed by users showing two different views to drive the vehicle: a ortogonal-top view perspective, or from a first-person view with a simulated on-board camera. Users were asked to do two trials: first driving the vehicle using a joy-pad without any assisted help (here named manual control); second, driving with the Shared-DWA assistance. This makes possible to compare both methods. The users had no previous experience about how to control this vehicle.
Three vehicles with different kinodynamic characteristics were tested. Vehicle A is similar to a bicycle, vehicle B to a motorcycle, and vehicle C to a very agile motorcycle (see Table 1).

	v_{max}	a_{brake}	$a_{forward}$	ω_{max}	ω_{acc}
A	$4\,m\,s^{-1}$	$6\,m\,s^{-2}$	$3\,m\,s^{-2}$	$1.5\,rad\,s^{-1}$	$50\,rad\,s^{-2}$
B	$8\,m\,s^{-1}$	$6\,m\,s^{-2}$	$6\,m\,s^{-2}$	$3\,rad\,s^{-1}$	$50\,rad\,s^{-2}$
C	$10\,m\,s^{-1}$	$15\,m\,s^{-2}$	$7\,m\,s^{-2}$	$3\,rad\,s^{-1}$	$50\,rad\,s^{-2}$

Table 1. Kinodynamic description of the vehicles.

Manual control. Results showed that vehicles B and C were very hard to control with manual control in both views (first-person view and top-view). Users collided in less than 10 meters and found severe problems to perform 90º turns. They also had problems trying to manage the vehicle at slower speeds since they were not accustomed to the LCI. Better results were achieved with vehicles B and C if velocities were $v_{max}=4\,ms^{-1}$ and $\omega_{max}=1.5\,rad\,s^{-1}$. After several minutes of training they were able to drive the vehicle with less than one collision per minute using the global view. Finally, they had some problems steering the vehicle along a desired path. In this case, users mentioned that they were not able to react quickly to obstacles (specially in the first-person view) and also they pointed out that they did not feel comfortable with a joypad controlling a non-holonomic vehicle at high speed.
Shared-DWA control. Users could manage without collisions the three robots reaching velocities near to the maximum and accomplishing some requested missions. For vehicle A, users got similar results than in manual control (after the training process) and satisfaction degree were pretty similar. However some of them complained about the slower speeds (in comparison with manual control) when they drove through narrow corridors or close to the walls. The results were also positive when managing vehicle B. They were able to achieve different requested goals in the scenarios without colliding. However, at the first trials they complained because sometimes directions chosen by the system were different to the desired. This was because system intrusiveness increased and users lose much more control over the vehicle. After 5 minutes of training two of the users were able to overtake this problem by commanding slower speeds at the intersections. Results and satisfaction degree was higher for vehicle C, where system intrusiveness decreased because of its higher braking capability. Therefore they were able to take hard 90º turns at high speeds without colliding (mean speed of $v_{max}=9\,ms^{-1}$) .

7 Conclusions and Future work

A novel navigation assistance method for controlling a kinodynamically constrained vehicle has been proposed. It makes possible avoid collisions when driving vehicles at high speed in unstructured scenarios. Moreover a state-space collision distance metric was presented,

ISBN 978-3-8007-3601-0

which is very useful to regulate the navigation assistance intrusiveness in a continuous and gradual way (avoiding context switching). Preliminary testing with 4 users in a simulated environment showed that the method could be specially useful for semi-autonomous teleoperation of a mobile robot whose shape and collision distance were hard to perceive properly by a human using a first-person camera. Future work includes usability tests with real vehicles and considering non-circular trajectories. The last would make possible to take hard turns at high speed, by performing an automatic aperture trajectory even if the user is not familiar with the vehicle kinodynamic constraints.

Acknowledgments

This work was partially supported by projects Vulcano (TEC2009-10639-C04-02) and Biosense (TEC2012-37868- C04-02); funded by the Spanish Ministry of Science and Innovation, and by project GameUp: Game-based Mobility Training and Motivation of Senior Citizens, funded by Ambient Assisted Living (AAL) Joint Programme 2011 call of the European Union.

References

[1] J. Hall and R. West, “Role of formal instruction and informal practice in learning to drive.,” no. Ergonomics, 39(4), 693–706., 1996.

[2] S. Vicente Diaz, C. Amaya Rodriguez, F. Diaz Del Rio, A. Civit Balcells, and D. Cagigas Muniz, “TetraNauta: a intelligent wheelchair for users with very severe mobility restrictions,” in *Proceedings of the International Conference on Control Applications*, 2002, vol. 2, pp. 778–783.

[3] A. C. B. F. D. del R. J. Sevillano and G. Jiménez, “SIRIUS: A Low Cost High Performance Computerized Wheelchair,” in *Proc. of the Int. Workshop on Medical Robots, Pág. 23-30. Vienna. October 1996*, 1996.

[4] R. a. R. Cooper, “Intelligent control of power wheelchairs,” *IEEE Eng. Med. Biol. Mag.*, vol. 14, no. 4, pp. 423–431, 1995.

[5] C. Urdiales, J. M. Peula, M. Fdez-Carmona, C. Barrué, E. J. Pérez, I. Sánchez-Tato, J. C. Toro, F. Galluppi, U. Cortés, R. Annichiaricco, C. Caltagirone, and F. Sandoval, “A new multi-criteria optimization strategy for shared control in wheelchair assisted navigation,” *Auton. Robots*, vol. 30, no. 2, pp. 179–197, Oct. 2010.

[6] R. Tang, “A Semi-autonomous Wheelchair Navigation System,” no. February, 2012.

[7] T. Carlson and Y. Demiris, “Human-wheelchair collaboration through prediction of intention and adaptive assistance,” *2008 IEEE Int. Conf. Robot. Autom.*, pp. 3926–3931, May 2008.

[8] T. Carlson and Y. Demiris, “Collaborative control for a robotic wheelchair: evaluation of performance, attention, and workload.,” *IEEE Trans. Syst. Man. Cybern. B. Cybern.*, vol. 42, no. 3, pp. 876–88, Jun. 2012.

[9] H. Road, “Who’s intelligent? Wheelchair, driver or both?,” *IEEE Int. Conf. Control Appl. Glas. Scotland, U.K., Sept. 2002*, pp. 760–765, 2002.

[10] T. Carlson and Y. Demiris, “Collaborative Control in Human Wheelchair Interaction Reduces the Need for Dexterity in Precise Manoeuvres,” *Proc. ACM/IEEE HRI-08 Work. Robot. Help.*, no. March, pp. 59–66, 2008.

[11] T. Rofer, C. Mandel, and T. Laue, “Controlling an automated wheelchair via joystick/head-joystick supported by smart driving assistance,” *2009 IEEE Int. Conf. Rehabil. Robot.*, pp. 743–748, Jun. 2009.

[12] Q. Zeng, “A collaborative wheelchair system,” 2008.

[13] A. C. Lopes, U. Nunes, L. Vaz, and L. Vaz, “Assisted navigation based on shared-control, using discrete and sparse human-machine interfaces,” *Eng. Med. Biol. Soc. EMBC 2010 Annu. Int. Conf. IEEE*, vol. 2010, pp. 471–474, 2010.

[14] G. Bourhis and M. Sahnoun, “Assisted Control Mode for Smart Wheelchair,” vol. 00, no. c, 2007.

[15] J. Abascal, B. Bonail, Á. Marco, R. Casas, and J. L. Sevillano, “AmbienNet: an intelligent environment to support people with disabilities and elderly people,” in *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, 2008, pp. 293–294.

[16] E. B. Vander Poorten, E. Demeester, E. Reekmans, J. Philips, and J. De Schutter, “Powered Wheelchair Navigation Assistance through Kinematically Correct Environmental Haptic Feedback,” pp. 3706–3712, 2012.

[17] M. Sato and T. Tomizawa, “Development of a collision-avoidance assist system for an electric cart,” *2011 IEEE Int. Conf. Robot. Biomimetics*, pp. 337–342, Dec. 2011.

[18] “HLDI Bulletin - Volvo collision avoidance features: initial results,” *Highw. Loss Data Institute.*, vol. 29, no. 5, 2012.

[19] J. Kolodko, T. Noël, Michael Parent, and Ljubo Vlacic, “Cooperative Autonomous Driving,” no. March, pp. 44–49, 2005.

[20] S. Anderson, “Constraint-based planning and control for safe, semi-autonomous operation of vehicles,” *Intell. Veh. …*, pp. 383–388, Jun. 2012.

[21] B. S. Ashley and M. Blommer, “Crashless Cars : Making Driving Safer,” pp. 3–6, 2009.

[22] A. Eidehall, *An automotive lane guidance system*, no. 1122. 2004.

[23] T. Fraichard and H. Asama, “Inevitable collision states — a step towards safer robots?,” *Adv. Robot.*, vol. 18, no. 10, pp. 1001–1024, Dec. 2004.

[24] J. Borenstein and Y. Koren, “Real-time obstacle avoidance for fast mobile robots,” *IEEE Trans. Syst. Man. Cybern.*, vol. 19, no. 5, pp. 1179–1187, 1989.

[25] Y. Koren and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” *Proceedings. 1991 IEEE Int. Conf. Robot. Autom.*, pp. 1398–1404, 1991.

[26] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *Robot. Autom. Mag. IEEE*, vol. 4, no. 1, pp. 23–33, 1997.

[27] R. Simmons, “The Curvature-Velocity Method for Local Obstacle Avoidance The Curvature-Velocity Method,” *Work*, no. April, 1996.

[28] J. Minguez and L. Montano, “Extending reactive collision avoidance methods to consider any vehicle shape and the kinematics and dynamic constraints,” *IEEE Trans. Robot.*, no. Figure 1, pp. 1–13, 2008.

[29] J. Blanco, J. Gonzalez, and J. Fernandez-Madrigal, “The Trajectory Parameter Space (TP-Space): A New Space Representation for Non-Holonomic Mobile Robot Reactive Navigation,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 1195–1200.

[30] P. D. Nisbet, “Who’s intelligent? Wheelchair, driver or both?,” pp. 760–765, 2002.

[31] R. Vaughan and B. Gerkey, “Really reusable robot code and the player/stage project,” *Eng. Exp. Robot.*, 2007.

ISBN 978-3-8007-3601-0