A rule-based assistive control algorithm for safe navigation for a powered wheelchair*

Catalin Stefan Teodorescu[1], Bingqing Zhang[1] and Tom Carlson[1]

Abstract— **This paper addresses the problem of safe navigation in an environment with randomly placed static obstacles. We convert a commercial powered wheelchair into a semi-autonomous vehicle with limited sight (environment awareness), by instrumenting it using off-the-shelf ultrasonic sensors and associated electronic boards (Arduino). In the continuity of our previous work where we had used stochastic dynamic programming to formulate optimization problems which led to relatively large size look-up tables that can be used as supervisory control, here we propose to extract rules using those results (that data). The advantage of this approach is a low-computational cost for future online implementation, and the drawback is a suboptimal policy. The feasibility is assessed by running simulations in a fairly realistic environment (Unity3D).**

I. INTRODUCTION

This is our latest work in a series of 3 papers dedicated to solving the problem of the nearest obstacle avoidance using the method of stochastic dynamic programming (SDP) [1]. To recall, our main contribution in [2] was the dynamic modeling of the physics (mechanics) of a 2-degree-of-freedom (DoF) vehicle moving down a variable angle slope. This was a control-oriented model, fairly simple but keeping significant, representative physical phenomena of interest. This was used in our follow-up paper [3] to address a model-based control problem, namely the obstacle avoidance of a vehicle with limited awareness of the environment (via ultrasonic sensors calibrated for a maximum of 2.8 meters range). Moreover, the driver's intention was modeled as a stochastic process, which we called the *blind driver* model. We formulated two optimization-based problems, the so-called *longitudinal* and *lateral control*, taking into account both the agreement with respect to user's intention as well as the vehicle's ability to safely stop/turn. We have shown the concept feasibility by running simulations in Matlab.

In this paper, we continue with the same control method presented in [2], [3] and:

- extend it towards both forward and backwards obstacle avoidance, whereas in [3] we have only dealt with the forward case;
- analyze SDP results (fairly large size look-up tables) and extract a simple rule-based control with low computation requirement; this can be straightforwardly implemented online using embedded boards (e.g. Arduino, STM32);
- run simulations in a more realistic environment, namely Unity3D.

All this work is intended to be implemented and tested in future, on the research platform depicted in Fig. 1. The clinical and social motivation is to enable a wider range of patients to be allowed to use powered wheelchairs, compared to today's practice, and more specifically people who suffer from cognitive (mental), vision, body impairments [4].

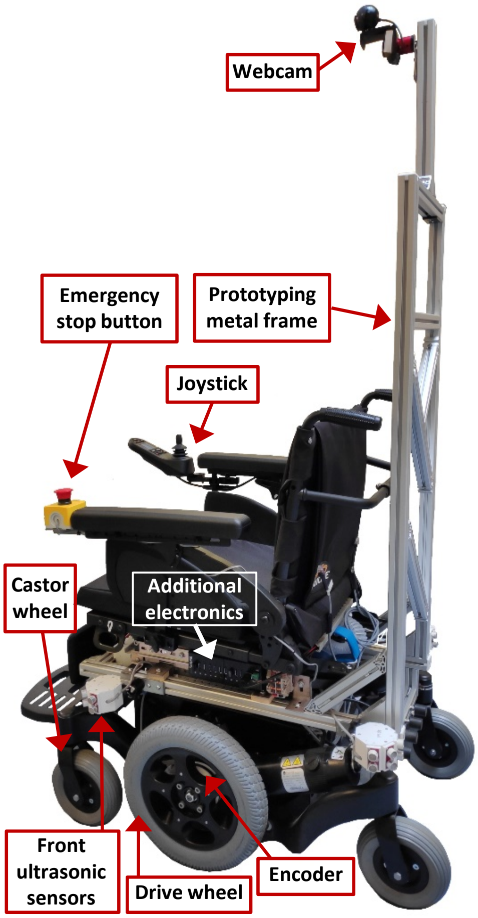


Fig. 1: Instrumented wheelchair.

Although historically the community of assistive technologies mainly exploited results from the robotics community [5], in our work we got inspiration from the automotive field.

*This work is funded by the project INTERREG VA FMA ADAPT – "Assistive Devices for empowering disAbled People through robotic Technologies" http://adapt-project.com/index.php. The FMA Program is a European Territorial Cooperation program which aims to finance ambitious cooperation projects in the border region between France and England. The Program is supported by the European Regional Development Fund (FEDER)

[1] Catalin Stefan Teodorescu, Bingqing Zhang and Tom Carlson are with Aspire Create, University College London, Royal National Orthopaedic Hospital, HA7 4LP, UK {s.teodorescu, bingqing.zhang.18, t.carlson}@ucl.ac.uk

We took the concept of stochastic dynamic programming from [6] and translated it towards a substantially different control problem.

This paper is organized as follows. In section II we present the discrete model involving behavior both deterministic (specifically the wheelchair dynamics) and non-deterministic (namely the user's intention, initial position of the closest obstacle). Next, in section III we put forward the latest state of the custom-made electronics added onto the instrumented wheelchair depicted in Fig. 1. After that, in section IV we show how we formulated the forward-backwards safe driving problem. Finally, section V is dedicated to simulations and the paper ends with Conclusions.

II. MODELING

The system dynamics is described by a 6-state variables system having the following coupled, nonlinear dynamics:

$$v_{\mathrm{d},k+1} = w_{v_{\mathrm{d},k}} \tag{1}$$

$$\omega_{\mathrm{d},k+1} = w_{\omega_{\mathrm{d},k}} \tag{2}$$

$$v_{k+1} = \sigma_1 v_k + \sigma_2 (v_{\mathrm{d},k} + v_{\mathrm{u},k}) \tag{3}$$

$$\omega_{k+1} = \sigma_3 \omega_k + \sigma_4 (\omega_{\mathrm{d},k} + \omega_{\mathrm{u},k}) \tag{4}$$

$$\theta^{\mathrm{b}}_{\mathrm{o},k+1} = \mathrm{atan2}\left(\begin{array}{c} d^{\mathrm{b}}_{\mathrm{o},k} \sin(\theta^{\mathrm{b}}_{\mathrm{o},k} - \Delta t\, \omega_k) + \Delta t\, v_k \sin(\Delta t\, \omega_k), \\ d^{\mathrm{b}}_{\mathrm{o},k} \cos(\theta^{\mathrm{b}}_{\mathrm{o},k} - \Delta t\, \omega_k) - \Delta t\, v_k \cos(\Delta t\, \omega_k) \end{array} \right) \tag{5}$$

$$d^{\mathrm{b}}_{\mathrm{o},k+1} = \sqrt{(d^{\mathrm{b}}_{\mathrm{o},k})^2 + (\Delta t\, v_k)^2 - 2\, d^{\mathrm{b}}_{\mathrm{o},k} \Delta t\, v_k \cos\theta_{\mathrm{o},k}} \tag{6}$$

where k stands for time index; superscript "b" for base frame (attached to the moving frame); subscript "o" for obstacle; Δt is the sampling time. The other notations are explained in the following.

For convenience, we converted the joystick 2-axes position into units of velocities. The first two eqs. (1)-(2) represent the driver's intention, expressed as the linear velocity demand v_{d} and the angular velocity demand ω_{d}. Their dynamics is governed by the stochastic variables $(w_{v_{\mathrm{d}}}, w_{\omega_{\mathrm{d}}})$ with associated probability distributions depending on a model. In our previous work [3] we have used the so-called *blind driver* model with uniform distribution. Other models can be mentioned, including novice driver, expert driver, tremor disorder, naughty child, etc. Pairs $(v_{\mathrm{d}}, \omega_{\mathrm{d}})$ are bounded by the admissible domain $\Omega_{v,\omega}$ which is an ellipsoid.

The middle eqs (3)-(4) are responsible of the deterministic wheelchair motion, expressed as the actual linear and angular velocities (v, ω), respectively. They are also bounded by the set $\Omega_{v,\omega}$ comprising both positive (vehicle advances forward, rotates clockwise as seen from above, respectively) and negative values (vehicle moves backwards, turns counterclockwise as seen from above, respectively). The interested reader can find all the details on how this model was derived in our previous work [2], including the nominal values for the parameters σ_1 to σ_4.

Finally, the location of the nearest obstacle sitting in the environment is specified by the absolute distance $d^{\mathrm{b}}_{\mathrm{o}} \in [d^{\mathrm{b,min}}_{\mathrm{o}}, d^{\mathrm{b,max}}_{\mathrm{o}}]$, where $d^{\mathrm{b,min}}_{\mathrm{o}} = 0.04$ meters and $d^{\mathrm{b,max}}_{\mathrm{o}} = 2.8$ meters; the angle $\theta^{\mathrm{b}}_{\mathrm{o}} \in (-\pi, \pi]$ radians, measured counterclockwise with respect to the forward direction of the advancing vehicle. This representation in polar coordinates is new compared to our previous work [3]. The motivation was twofold. Firstly, when the vehicle turns, the perception is that an obstacle positioned nearby changes its position by a little, compared to another obstacle sitting far away. For this reason we find it advantageous to work in polar coordinates. Later on, in the SDP formulation, when state variables are placed on an uniform grid, this would physically correspond to a circular grid in the real world (i.e. the environment), as illustrated in Fig. 2. In particular, in Fig. 2 notations Q_1 to Q_4 correspond to the 4 quadrants covering the entire range of $\theta^{\mathrm{b}}_{\mathrm{o}}$.

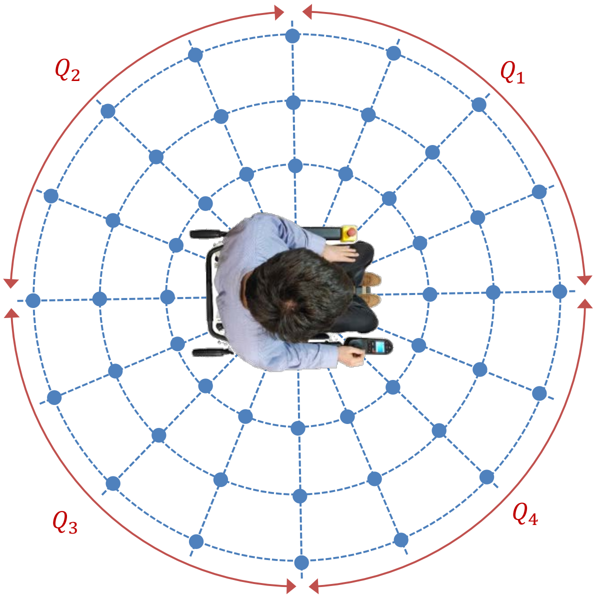


Fig. 2: Circular grid around the wheelchair used to describe the location of the nearest obstacle.

Second, the ultrasonic sensors measure distance $d^{\mathrm{b}}_{\mathrm{o}}$ directly, not the Cartesian position.

The control variables are the additional linear velocity v_{u} and angular velocity ω_{u}. They are intended to assist by correcting user's desired input $(v_{\mathrm{d}}, \omega_{\mathrm{d}})$ in order to adress safety concerns (obstacle avoidance). In particular, one control design requirement is to avoid contradicting user's intention:

$$\begin{array}{l} \mathrm{sign}(v_{\mathrm{d}} + v_{\mathrm{u}}) = \mathrm{sign}(v_{\mathrm{d}}) \text{ and} \\ \mathrm{sign}(\omega_{\mathrm{d}} + \omega_{\mathrm{u}}) = \mathrm{sign}(\omega_{\mathrm{d}}) \end{array} \tag{7}$$

This is an alternative to other approaches in the literature which neglect this criteria, for instance the *potential field* approaches [7], [8].

III. EXPERIMENTAL RIG

The instrumented wheelchair is an active research platform in constant evolution. In this section we would like to explain how the state variables of system (1)-(6) are measured online.

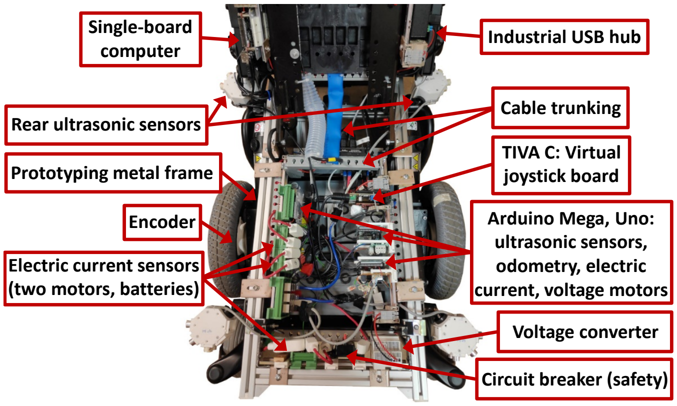

Single-board computer
Industrial USB hub
Rear ultrasonic sensors
Cable trunking
Prototyping metal frame
TIVA C: Virtual joystick board
Encoder
Arduino Mega, Uno: ultrasonic sensors, odometry, electric current, voltage motors
Electric current sensors (two motors, batteries)
Voltage converter
Circuit breaker (safety)

Fig. 3: Prototyping electrical components placed under driver's seat, attached to the inner side of the prototyping metal frame.

All the electronics was mounted under the driver's seat, and can be accessed by unfolding the wheelchair, see Fig 3.

A collaboration with INSA Rennes [9] allowed us to use a TIVA C board that is directly connected to the CAN bus of the wheelchair and have access to the joystick data $(v_{\mathrm{d}}, \omega_{\mathrm{d}})$ appearing in (1)-(2).

Part of the odometry, two industrial encoders have been mounted on each drive wheel, counting the rotation of the wheels. This can be converted into wheelchair Cartesian position and then estimate velocities (v, ω) from (3)-(4).

Twelve ultrasonic sensors were placed in an array covering 360°around the wheelchair. They allow measurement of the distance to the closest obstacle $d_{\mathrm{o}}^{\mathrm{b}}$ from eq. (6), while the angle θ_{o} can only be roughly known, with an uncertainty within the 45°beam range of each sensor.

IV. CONTROL

SDP suffers from the well-known *curse of dimensionality* (exponential increase of the required computation with each new state, control variable or uncertainty) [1]. Limited computation power makes control design on the 6-DoF plant model (1)-(6) impractical. For this reason we have proposed in [3] to decouple system dynamics and formulated two independent problems: the longitudinal and lateral control. Whereas in [3] we dealt with the forward navigation case, here we are interested in both forward and backwards safe navigation.

A. Longitudinal control

By making the assumption that the vehicle is moving in a straight line without turning ($\omega_{\mathrm{d}} \equiv 0$, $\omega \equiv 0$) the only danger can be the closest obstacle situated ahead or to the rear of the vehicle, but not to the sides ($\theta_{\mathrm{o}} \equiv 0$ or $\theta_{\mathrm{o}} \equiv \pi$). Thus we ended up with a 3-DoF system comprising eqs. (1), (3) and (6).

An issue often neglected or not clearly mentioned in the literature is when to enable a proposed control and when not. We address this by splitting state variables space (reachability domain) into 8 regions, see Fig. 4:

- region 1: $(v_{\mathrm{d}} < 0, v < 0, d_{\mathrm{o}}^{\mathrm{b}} > 0)$
- region 2: $(v_{\mathrm{d}} > 0, v < 0, d_{\mathrm{o}}^{\mathrm{b}} > 0)$
- region 3: $(v_{\mathrm{d}} < 0, v > 0, d_{\mathrm{o}}^{\mathrm{b}} > 0)$
- region 4: $(v_{\mathrm{d}} > 0, v > 0, d_{\mathrm{o}}^{\mathrm{b}} > 0)$
- region 5: $(v_{\mathrm{d}} < 0, v < 0, d_{\mathrm{o}}^{\mathrm{b}} < 0)$
- region 6: $(v_{\mathrm{d}} > 0, v < 0, d_{\mathrm{o}}^{\mathrm{b}} < 0)$
- region 7: $(v_{\mathrm{d}} < 0, v > 0, d_{\mathrm{o}}^{\mathrm{b}} < 0)$
- region 8: $(v_{\mathrm{d}} > 0, v > 0, d_{\mathrm{o}}^{\mathrm{b}} < 0)$

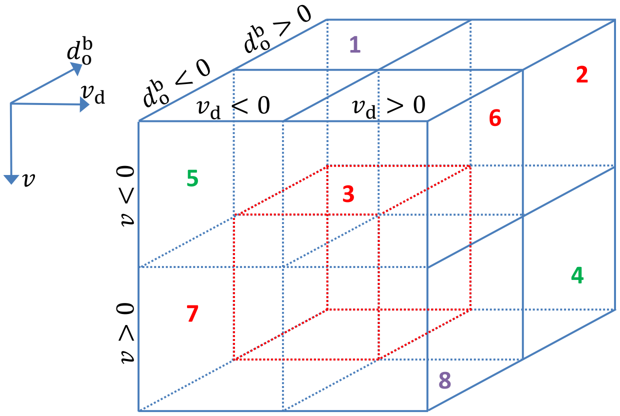

$d_{\mathrm{o}}^{\mathrm{b}}$
v_{d}
v
$d_{\mathrm{o}}^{\mathrm{b}} > 0$
$d_{\mathrm{o}}^{\mathrm{b}} < 0$
$v_{\mathrm{d}} < 0$
$v_{\mathrm{d}} > 0$
$v < 0$
$v > 0$
1
2
6
5
3
4
7
8

Fig. 4: State space divided into 8 rectangular cuboid shaped regions: *controllable* with green, *uncontrollable* with red, *no danger* with purple.

There are two situations that present no danger of hitting any obstacle. This happens when: (i) the vehicle is advancing backwards by following the associated instruction of the user, while the obstacle is situated in front of the vehicle (region 1); (ii) the vehicle is advancing forward, again following the corresponding instruction from the user, and the obstacle is situated to the rear of the vehicle (region 8). In these two situations, we do not need to enable any supervisory control, just let the user input control freely the wheelchair's behavior.

There are four situations when the wheelchair is uncontrollable (there is no reaction of the vehicle to inputs coming from the joystick). This happens: (i) when the vehicle advances forward but the joystick values correspond to requesting to advance backwards (regions 2 and 6); (ii) when the vehicle advances backwards but the joystick is set in position forward (regions 3 and 7). In all these situations, the wheelchair behaves as if we let go of the joystick and consequently starts braking.[1]

Finally, there are two situations when the wheelchair is controllable and a danger is present ahead (region 1) or at the rear (region 8). It is here that we need to enable assistive control, by formulating two optimization problems, one for each region. We solved them by SDP. We kindly guide the reader to check the problem formulation of our previous work [3] where we have treated only region 1. The same procedure can be applied for region 8. The outcome of these

[1] Note that this behavior corresponds to the default parameters set on this particular wheelchair and can be modified using the manufacturer's hardware dongle and software (dealer or OEM).

two SDP implementations are two look-up tables of fairly large size (in the order of tens of Megabytes) which can be fairly easy handled by prototyping hardware (e.g. Raspberry Pi, Odroid). However, their size can be problematic if an attempt is made to implement them on embedded boards with limited memory (e.g. Arduino, STM32). For this reason we propose to extract rules by making an analysis of the look-up table data.

Rule-based suboptimal control: Depending on the distance to the nearest obstacle $d_{\mathrm{o}}^{\mathrm{b}}$, there are two main tendencies that can be observed in the 3-dimensional look-up table data $v_{\mathrm{u}} = v_{\mathrm{u}}(v_{\mathrm{d}}, v, d_{\mathrm{o}}^{\mathrm{b}})$:

- at close distance values for $d_{\mathrm{o}}^{\mathrm{b}}$, the assistive control v_{u} will privilege safety thus counteract precipitated user's demand v_{d}; however, v_{u} will never contradict user's intention by virtue of condition (7);
- at far distance values for $d_{\mathrm{o}}^{\mathrm{b}}$, the user's demand will prevail and the assistive control effect will vanish.

Apart these two opposing tendencies, there is a transition region between them. The ability to extract some rules in this region depends highly on how fine (dense) the SDP grid was set up from the beginning. To simplify our presentation, we will use a linear interpolation function f depicted in Fig. 5, although more generally it is a polynomial. Note the maximum value of f is reached at the limit of visibility of the ultrasonic sensors $d_{\mathrm{o}}^{\mathrm{b}} = d_{\mathrm{o}}^{\mathrm{b,max}}$, while the minimum value of f is set to an arbitrary $d_{\mathrm{o}}^{\mathrm{b}} \geq d_{\mathrm{o}}^{\mathrm{b,min}}$ in order to allow safe stop. Regarding the latter, note the value set in Fig. 5 is rather too conservative, corresponding to a rather too careful design. However, it is compatible with the ideal (virtual) utrasonic sensor from section V. In a future work, this value should be lowered prior to the implementation on the actual vehicle.

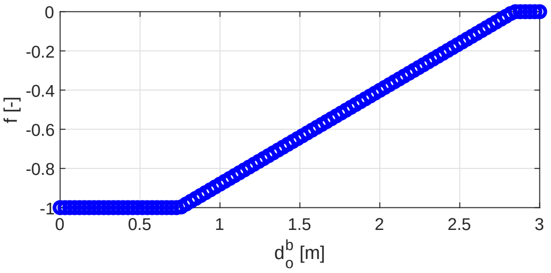


Fig. 5: Transition function $f(d_{\mathrm{o}}^{\mathrm{b}})$.

To summarize, the rule-based longitudinal control is presented as Algorithm 1. To simplify notations, we dropped out the index k everywhere.

```
$v_{\mathrm{u}} = 0$ //initialization: longitudinal control is disabled
if ($v_d > 0$ and $v \geq 0$ and $\theta_o^b > -22.5°$ and
  $\theta_o^b < 22.5°$) //region 4 and obstacle present ahead
  of the vehicle
or ($v_d < 0$ and $v \leq 0$ and ($\theta_o^b > 157.5°$ or
  $\theta_o^b < -157.5°$)) //region 5 and obstacle present to
  the rear of the vehicle
 then
    $v_{\mathrm{u}} = f(d_{\mathrm{o}}^{\mathrm{b}})\, v_{\mathrm{d}}$
end
```

Algorithm 1: Rule-based longitudinal control

B. Lateral control

In this case we make the assumption that vehicle is turning only, without advancing linearly ($v_{\mathrm{d}} \equiv 0$, $v \equiv 0$). Consequently, the nearest obstacle will always be situated at the same constant distance $d_{\mathrm{o}}^{\mathrm{b}}$. We end up with the 3-DoF system consisting of eqs. (2), (4) and (5).

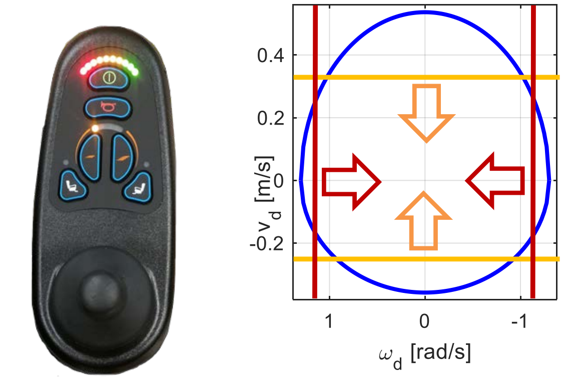
The same analysis presented in section IV-A was carried out. For brevity reasons, we present only the final result, namely the rule-based lateral control from Algorithm 2. As previously, to simplify notations, we dropped out the index k everywhere. Note that in Algorithm 2 we have used the same function f from Fig. 5 although more generally, they need not be the same.

C. Geometric interpretation

It is interesting to notice that Algorithms 1 and 2 have a simple visual interpretation. If we look on the joystick map $(v_{\mathrm{d}}, \omega_{\mathrm{d}})$, the effect of each control will be to reduce (shrink) the admissible domain of the cumulative request $(v_{\mathrm{d}} + v_{\mathrm{u}}, \omega_{\mathrm{d}} + \omega_{\mathrm{u}})$ by a certain amount. If we look on Fig. 6b:

- the longitudinal control acts on the vertical axis (see the straight boundary orange lines and arrows in Fig. 6b),
- the lateral control acts on the horizontal axis (see the straight boundary dark red lines and arrows in Fig. 6b)

A similar geometric analysis can be carried out with the control design of the team in INSA Rennes [4], [10].



(a) Physical wheelchair joystick

(b) Associated maximum admissible domain $\Omega_{v,\omega}$ in blue

Fig. 6: Reduction of the allowed control inputs sent to the power module, consequence of enabling the assistive (corrective) control.

To summarize, this assistive control acts as a virtual damper between the user and the obstacle. To justify this, we

```
doApplyLateralControl = false //initialization
if ($v_d > 0$ and $v > 0$) //vehicle advances forward and
 that state space is controllable
then
    if ($\omega_d > 0$ and $\omega > 0$ and $\theta_o^b \in Q_1$) //vehicle
     turns counterclockwise and there is an obstacle
     (danger) in the direction where it heads to
    or ($\omega_d < 0$ and $\omega < 0$ and $\theta_o^b \in Q_4$) //vehicle
     turns clockwise and there is an obstacle
     (danger) in the direction where it heads to
     then
        doApplyLateralControl = true
    end
end
if ($v_d < 0$ and $v < 0$) //vehicle advances backwards
 and that state space is controllable
then
    if ($\omega_d > 0$ and $\omega > 0$ and $\theta_o^b \in Q_3$) //vehicle
     turns counterclockwise and there is an obstacle
     (danger) in the direction where it heads to
    or ($\omega_d < 0$ and $\omega < 0$ and $\theta_o^b \in Q_2$) //vehicle
     turns clockwise and there is an obstacle
     (danger) in the direction where it heads to
     then
        doApplyLateralControl = true
    end
end
$\omega_u = 0$ //initialization: lateral control is disabled
if (doApplyLateralControl == true) then
    $\omega_u = f(d_o^b)\,\omega_d$
end
```

Algorithm 2: Rule-based lateral control

recall the damping force is defined as the product between a damping coefficient the actual speed. In our case, function $f(\cdot)$ acts as a variable damping coefficient.

V. SIMULATIONS

In this paper, we provide more realistic simulations compared to our previous work [2], [3].

Two tests have been performed to evaluate our proposed algorithm in terms of safety and manoeuvrability. A realistic wheelchair model is built in Unity3D where its kinematic constraints have been taken into account (see Fig. 7). For simpler implementation, an ideal (virtual) 360 degree ultrasonic sensor is used to detect the surrounding obstacles at a frequency of 20 Hz. The velocity commands sends are sent from ROS to Unity3D every 0.08 seconds.

A. Safety Test

In order to test whether our proposed shared-control algorithm ensure safety, a simple safety test has been implemented. Six scenarios have been designed to test longitudinal and lateral control by placing a cylinder in different positions with respect to the wheelchair. The user expresses his driving intention by pressing the corresponding arrow keys on the keyboard. In this test, we deliberately drive the wheelchair towards the obstacle and record the resultant behavior.

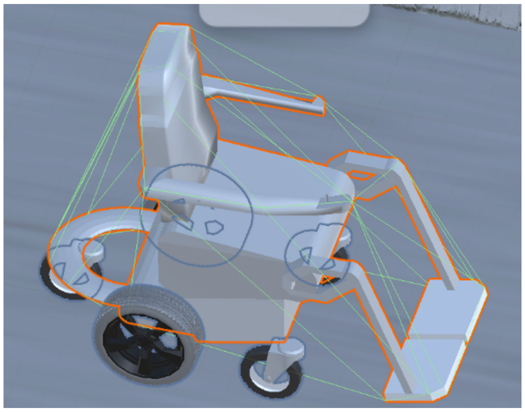


Fig. 7: Instrumented wheelchair model in Unity3D

B. Maneuverability Test

In order to test the maneuverability of our algorithm, we designed a more complex navigation scenario by building a maze with four static cylinder obstacles (Fig. 8). During the test, an operator drives the wheelchair through the maze from a starting position to an end line. The completion time and number of collisions were recorded to evaluate the performance. In addition, to guarantee the comparability, the operator attempted to avoid the obstacles in the same pattern (e.g. avoid the first obstacle from the left, then avoid the second one from the right etc.) through three repeated trials.

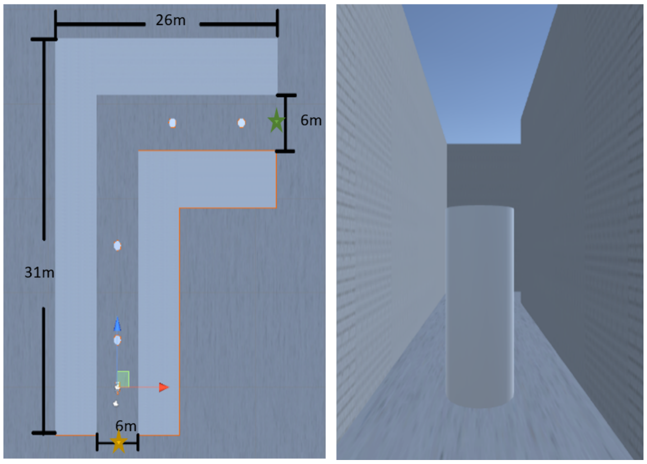


Fig. 8: Maze for manoeuvrability test, walls have thickness of 6m and static obstacles have radius of 0.5m. Yellow star indicate the start position and green star shows the goal. (a) Top view; (b) Wheelchair view

C. Results

No collisions were observed during the safety test. For longitudinal and lateral control, the wheelchair simply decreased its linear and angular velocities, respectively, and stopped correctly at the minimum allowed distance.

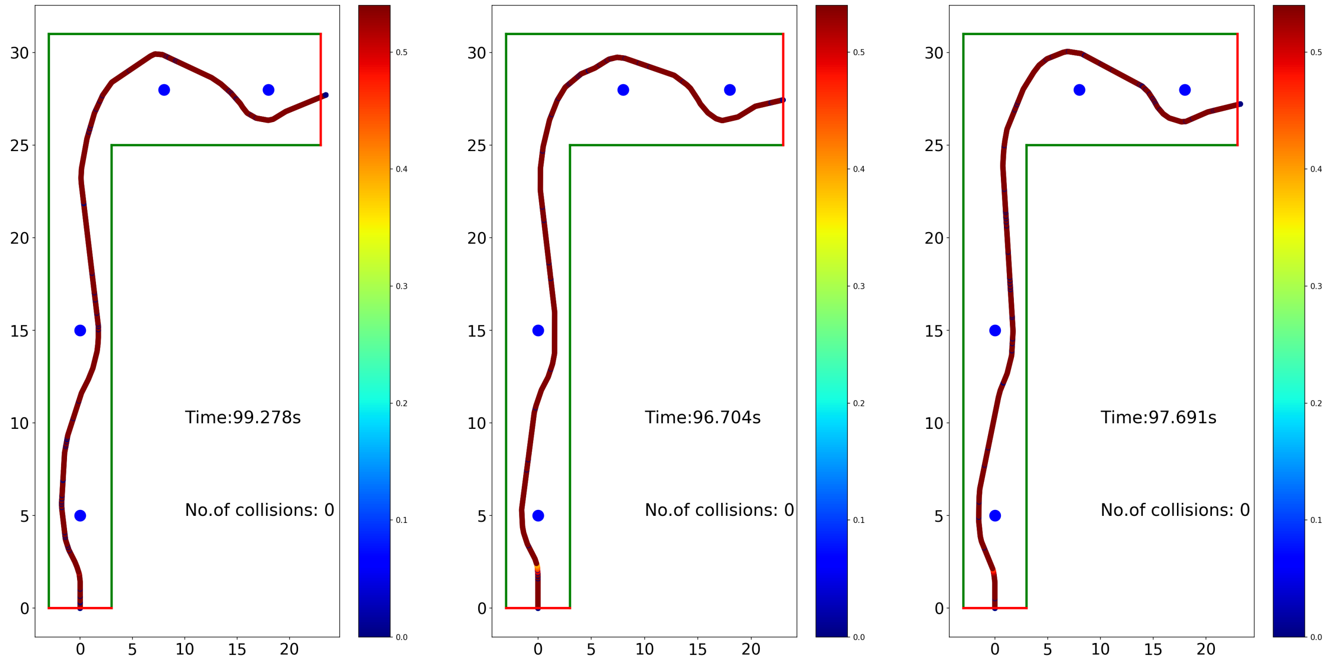


Fig. 9: View from above: Trajectory of the vehicle's center of mass, in Cartesian space (x-axis and y-axis measured in meters), for 3 trials of the manoeuvrability test. The color at each point in space along the trajectory represents vehicle's linear velocity $v \in [0, v^{\max}]$, with $v^{\max} = 0.54$ m/s (see colorbar). Green lines indicate the walls; the four static obstacles that need to be avoided are illustrated with blue dots.

In terms of the maneuverability test, three trials have been performed and visualized as shown in Fig. 9. On average, the completion time is 97.8 seconds and no collisions occurred.

Our preliminary simulation results indicate that the control algorithm does indeed exhibit the level of safety required, whilst still allowing sufficient maneuverability. In the future, we aim to test it in more complicated scenarios and then eventually on the prototype wheelchair that we have built. This will involve experiments with human participants, which could be healthy, able-bodied or potential end-users or existing wheelchair users.

VI. Conclusions

This paper presented two velocity-based obstacle avoidance control algorithms, namely the longitudinal and the lateral control, respectively. Simulations in Unity3D on a wheelchair model show feasibility, whilst we intend to test it on the actual vehicle in the future.

References

[1] D. Bertsekas, *Dynamic programming and optimal control*, 3rd ed. Belmont, MA: Athena Scientific, 2005, vol. 1.

[2] C. S. Teodorescu, B. Zhang, and T. Carlson, "Probabilistic shared control for a smart wheelchair: A stochastic model-based framework," in *2019 IEEE International Conference on Systems, Man, and Cybernetics*, Bari, Italy, October 2019.

[3] ——, "A stochastic control strategy for safely driving a powered wheelchair," in *2020 IFAC World Congress*, Berlin, Germany, July 2020, (under review).

[4] L. Devigne, "Low-cost robotic solutions for safe assisted power wheelchair navigation : towards a contribution to neurological rehabilitation," Ph.D. dissertation, INSA, Rennes, France, December 2018.

[5] R. C. Simpson, "Smart wheelchairs: A literature review," *Journal of Rehabilitation Research and Development*, vol. 42, no. 4, pp. 423–436, 2005.

[6] C.-C. Lin, H. Peng, and J. W. Grizzle, "A stochastic control strategy for hybrid electric vehicles," in *American Control Conf.*, Boston, MA, June-July 2004, pp. 4710–4715.

[7] M. R. Petry, A. P. Moreira, R. A. Braga, and L. P. Reis, "Shared control for obstacle avoidance in intelligent wheelchairs," in *2010 IEEE Conference on Robotics, Automation and Mechatronics*, Singapore, June 2010, pp. 182–187.

[8] H. Seki, Y. Kamiya, and M. Hikizu, "Real-time obstacle avoidance using potential field for a nonholonomic vehicle," in *Factory Automation*. IntechOpen, 2010.

[9] M. Babel, F. Pasteau, S. Guégan, P. Gallien, B. Nicolas, B. Fraudet, S. Achille-Fauveau, and D. Guillard, "HandiViz project: clinical validation of a driving assistance for electrical wheelchair," in *2015 IEEE Int. Workshop on Advanced Robotics and its Social Impacts (ARSO)*, Lyon, France, July 2015, pp. 1–6.

[10] L. Devigne, V. K. Narayanan, F. Pasteau, and M. Babel, "Low complex sensor-based shared control for power wheelchair navigation," in *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea, October 2016, pp. 5434–5439.