Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

A Stochastic Control Strategy for Safely Driving a Powered Wheelchair [★]

Catalin Stefan Teodorescu[*] Bingqing Zhang[*] Tom Carlson [*]

[*] *Aspire Create, University College London, Royal National Orthopaedic Hospital, HA7 4LP, UK (e-mails: {s.teodorescu, bingqing.zhang.18, t.carlson}@ucl.ac.uk).*

Abstract: In this paper we deal with model-based control design in the presence of uncertainties. We use Stochastic Dynamic Programming to solve two problems, called longitudinal and lateral vehicle control. The goal is to allow safe driving (navigation) of a moving vehicle in an environment with static obstacles. We show how to define the optimization problems given the stochastic driver behavior and environment. The vehicle dynamics model is deterministic (obeys physical laws) and is explicitly integrated into the optimization problem. In terms of results, the numerically computed control policies provide best-on-average performance (according to the expected value operator). In simulation, it is shown that the vehicle effectively avoids obstacles, thus ensuring a safe drive experience.

Keywords: stochastic dynamic programming, inventory control, model-based control, velocity control, robot navigation, robot dynamics, electric vehicles, vehicle dynamics

1. INTRODUCTION

From a practical point of view, this work is about converting an off-the-shelf powered wheelchair into a semi-autonomous vehicle. The goal is to ensure a safe drive for users in an environment with obstacles. To address this safety concern, a *supervisory control* is designed that can assist driver decisions. This will correct potentially dangerous requests, by taking into account information coming from cheap ultrasonic sensors mounted to the sides of the vehicle. Environment awareness is the key to a safe drive.

A survey of task-oriented approaches to control dynamical systems like the powered wheelchairs – see Simpson (2005); Abbink et al. (2018), shows that the driving experience is not a deterministic process. For this reason, we think that integrating the uncertainty (stochastic behavior) might be the key to improved control. After careful selection of the control design, we decided to implement Stochastic Dynamic Programming (SDP) – see Bertsekas (2005). This seems to be the most suited to handle our inventory problem: it consists of an optimization problem that can be solved *offline*. The outcome is a lookup table that can easily be implemented *online*, on the actual vehicle.

Literature review in assistive technologies: Stochastic dynamic programming (SDP) has been widely used in robot navigation with uncertainties. In the area of assistive technologies, SDP has been applied to solve the navigation problem for shared controlled wheelchairs in a number of studies: Ghorbel et al. (2018); Demeester et al. (2008). They modeled the collaborative control as Partially Observable Markov Decision Process (POMDP) where the uncertainty (stochasticity) in user's intention and the environment are considered. Although some similarities can be found, the control problem is formulated and solved differently in our paper, in terms of following aspects:

(i) prediction of user's intention: studies mentioned above assume all users are goal driven. Consequently, they are matched against a set of pre-defined clustered goals. However, no such assumption is required in our approach where the user's intention is modeled as a stochastic process. In this work, we explicitly present a *blind driver* model and leave the other types of models for future work.
(ii) wheelchair dynamics model: an accurate system model is essential for providing useful information in solving such stochastic problem. Different from most of the works that neglect transitory dynamics (e.g. the relation between the driver's demanded velocity and the actual velocity), here we model them explicitly. Thus, we avoid having to rely on steady-state assumption.
(iii) *online* computation efficiency: the main drawback for POMDP methods are the high online computational cost. Although it has been largely improved in recent approximation methods such as hindsight and despot – see Ye et al. (2017), the best processing frequency is around 3 Hz. As for our method, the optimization problem is solved entirely offline, thus leveraging by at least a factor 10 the online decision making. Consequently, due to the way optimization is being handled, our method seems to be more suited for safe driving in static (not dynamic) uncertain environments.

Other limitations of SDP in general, as well as their consequences on our implementation, will be highlighted later on, in section 3.

Literature review in automotive field: This work was largely inspired by the SDP implementation of *longitudinal vehicle control* of Lin et al. (2004). This involved theory and simulations, and had not been evaluated on hardware according to

[★] 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)

Cook et al. (2006). Liu et al. (2018) mentioned that SDP had been proposed earlier by Kolmanovsky et al. (2002).

In parallel, we are aware of the works in Chalmers University Sweden, namely the articles by Rutquist (2002) (later on mentioned in his PhD thesis Rutquist (2017)) and Johannesson et al. (2007).

All aforementioned articles illustrate applications of SDP to specific types of vehicles (automotive field). In spite of small differences (e.g., use case on various vehicle types, modeling in terms of acceleration *versus* power, etc.), the methodology is fundamentally the same. It is reused here, in our work, for a substantially different type of vehicle and a more complex control problem.

Assistive control using SDP: In this work we consider the simultaneous control of the linear and angular velocities (v, ω) of a powered wheelchair. To achieve this, we need models for each of the four blocks depicted in Fig. 1.

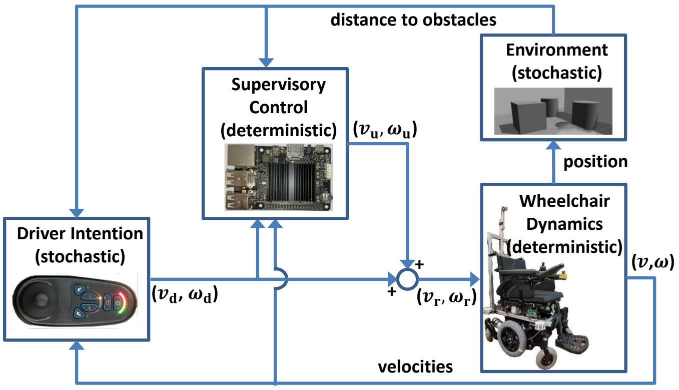

distance to obstacles
Supervisory Control (deterministic)
$(v_{\mathrm{u}}, \omega_{\mathrm{u}})$
Environment (stochastic)
position
Wheelchair Dynamics (deterministic)
(v,ω)
Driver Intention (stochastic)
$(v_{\mathrm{d}}, \omega_{\mathrm{d}})$
+
+
$(v_{\mathrm{r}}, \omega_{\mathrm{r}})$
velocities

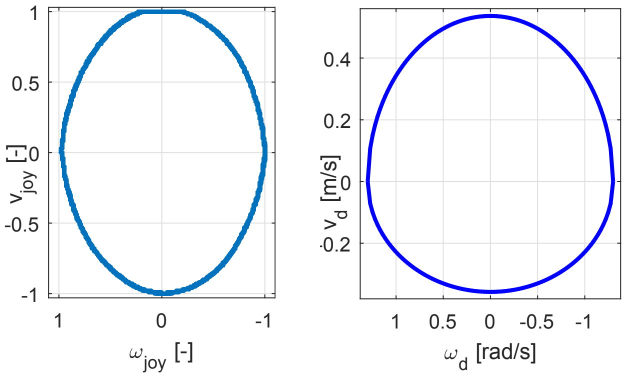
Fig. 1. Schematic overview of the stochastic framework.

This article is organized as follows. Section 2 is dedicated to modeling three blocks of Fig. 1, namely *Wheelchair dynamics*, *Environment* and *Driver Intention*. Then, we formulate two separate control problems in section 3 called the *longitudinal control*, where the aim is to control the variable v, and the *lateral control* where the ω variable is controlled. Eventually, they are joined together in order to allow safe driving. Finally, simulations are presented in section 4 and the paper ends with conclusions.

2. MODELING

In this work we consider the situation of a vehicle sitting and advancing on flat ground (no slope inclination).

Driver Intention: The driver expresses their intention by moving the joystick. The forward-backwards movement corresponds to a desired linear velocity v_{d}, whereas the left-right movement is interpreted as a desired angular velocity ω_{d}. Fig. 2 shows the relationship between joystick raw data $(v_{\mathrm{joy}}, \omega_{\mathrm{joy}})$ – an ellipse, and the associated desired velocities $(v_{\mathrm{d}}, \omega_{\mathrm{d}})$. All $(v_{\mathrm{d}}, \omega_{\mathrm{d}})$ pairs inside the blue shape of Fig. 2b are attainable (reachable) by the actual velocities (v, ω): one can see $(v_{\mathrm{d}}, \omega_{\mathrm{d}})$ pairs as user-demanded *references* which will, in turn, be attained after a while (i.e. after a short transient). The curve of Fig. 2b can be experimentally identified, or calculated according to the so-called *profiles* (pre-specified by the manufacturer or custom-modified using dedicated software and hardware), as we have done in Fig. 2. Note that both x-axes have been reversed in Fig. 2 so that positive angular velocities are found on the left-hand side of each figure. This is to overlap with driver gestures and thus facilitate their intuitive interpretation.



(a) Experimentally identified joystick input boundary (normalized)

(b) Associated maximum demanded velocities

Fig. 2. Boundary of the driver joystick raw data and associated maximum velocities.

Unlike wheelchair dynamics which obey physical laws (e.g. gravity, inertia, electric current flow, etc.) and can be modeled in a deterministic manner, the *driver intention* does not. Even in situations where the driver is given specific rules (e.g. a path to follow, avoid obstacles, etc.) there is still significant uncertainty involved. For this reason, we choose a *stochastic model* for the driver's intention at next sampling time k:

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

where $(w_{v_{\mathrm{d},k}}, w_{\omega_{\mathrm{d},k}})$ are two independent random variables within the boundaries of Fig. 2b. Without loss of generality for the SDP methodology, we assume here an uniform distribution for $(w_{v_{\mathrm{d},k}}, w_{\omega_{\mathrm{d},k}})$. We call this the *blind driver* model, because it acts irrespective of the presence of obstacles in the environment (there is equal chance of hitting or not obstacles).

In practice, we expect temporal as well as state variables dependency in the probability distribution. More complicated driver models could be integrated in a future study.

Wheelchair dynamics: Wheelchair dynamics obey physical laws which are well known. This allowed us to build a deterministic physical model using Euler-Lagrange method – see our previous work Teodorescu et al. (2019). We obtained:

$$\begin{pmatrix} v_{k+1} \\ \omega_{k+1} \end{pmatrix} = \mathrm{sat}_{\Omega_{v,\omega}} \begin{pmatrix} \sigma_1 v_k + \sigma_2 (v_{\mathrm{d},k} + v_{\mathrm{u},k}) \\ \sigma_3 \omega_k + \sigma_4 (\omega_{\mathrm{d},k} + \omega_{\mathrm{u},k}) \end{pmatrix} \tag{2}$$

where parameters σ_1 to σ_4 are given in Teodorescu et al. (2019).Although dynamics of variables (v_k, ω_k) are uncoupled (as can be seen from (2)), their maximum allowed values are dependent on one another. For this reason, we preferred to use vector notation in (2). The saturation ensures values (v_k, ω_k) stay within (inside) the $\Omega_{v,\omega}$-set, which corresponds to the blue shape of Fig. 2b. Note that maximum demanded velocities $(v_{\mathrm{d}}, \omega_{\mathrm{d}})$-set is exactly the same as the maximum attainable (reachable) velocities (v, ω)-set, called $\Omega_{v,\omega}$ in (2). In practice, there is a combination of a low-level velocity-feedback control and a Safety block that cuts off electric current flowing to motors that goes beyond certain values.

Coordinate frames and transformation. Notations. In this article we use two coordinate frames. On the one hand, the *base* *frame* is attached to the moving vehicle. Its two perpendicular axes are as follows: x_b-axis points forward (in front of the vehicle) and y_b-axis points to the left-hand side of the vehicle. Together, they form the base frame $o_\mathrm{b}x_\mathrm{b}y_\mathrm{b}$ and can be seen in Fig. 3. On the other hand, the *inertial frame* $o_0x_0y_0$ is fixed in the Cartesian space. By convention, we fix the inertial frame with respect to the initial position of the wheelchair (namely, when the vehicle is powered on): the origin o_0 is situated at the center of mass of the left drive wheel; x_0-axis and y_0-axis are parallel to x_b-axis and y_b-axis, respectively, the initial time t_0. This can be easily visualized later on, in section 4, Fig. 10.

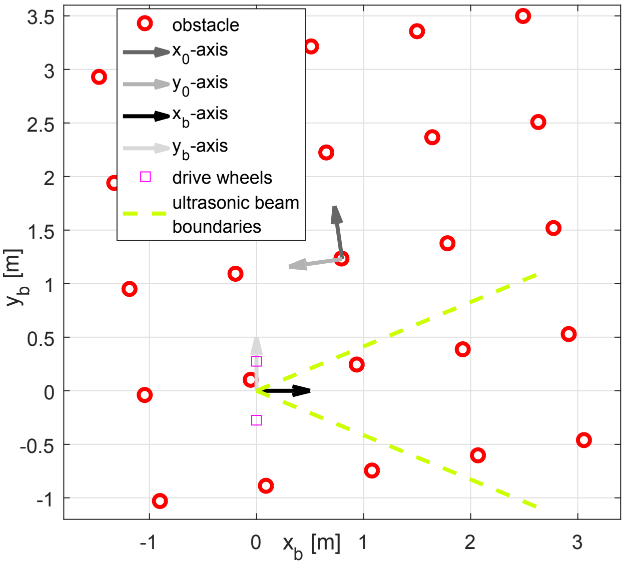


Fig. 3. View from above: vehicle advancing in an environment with uniformly-spaced obstacles.

In order to familiarize our reader with notations used in this paper (consistent with Spong et al. (2005)), we find it instructive to indicate the transformation of coordinates between the two aforementioned frames. Given a fixed obstacle in the inertial frame of coordinates $(x_\mathrm{o}^0, y_\mathrm{o}^0)$ it can be expressed in coordinates of the moving base frame $(x_{\mathrm{o},k}^\mathrm{b}, y_{\mathrm{o},k}^\mathrm{b})$ as:

$$\begin{pmatrix} x_\mathrm{o}^0 \\ y_\mathrm{o}^0 \end{pmatrix} = R_{\mathrm{o},k}^0 \begin{pmatrix} x_{\mathrm{o},k}^\mathrm{b} \\ y_{\mathrm{o},k}^\mathrm{b} \end{pmatrix} + o_{\mathrm{b},k}^0, \text{ with } R_{\mathrm{o},k}^0 = \begin{pmatrix} \cos\varphi_k & -\sin\varphi_k \\ \sin\varphi_k & \cos\varphi_k \end{pmatrix}$$

and $o_{\mathrm{b},k}^0 = \begin{pmatrix} x_{o_\mathrm{b},k}^0 \\ y_{o_\mathrm{b},k}^0 \end{pmatrix}$. Subscripts "o", "$o_\mathrm{b}$" stand for the obstacle and the origin of the base frame, respectively; superscripts "0", "b" indicate the inertial and the base frames, respectively. φ_k is the vehicle's angular displacement (its time-derivative is equal to the vehicle's angular velocity ω_k from (2)).

Environment: The vehicle is surrounded by obstacles, as can be seen in Fig 3. Below, we model the dynamics of one obstacle in coordinates of the base frame:

$$\begin{pmatrix} x_{\mathrm{o},k+1}^\mathrm{b} \\ y_{\mathrm{o},k+1}^\mathrm{b} \end{pmatrix} = \mathrm{sat}_{\Omega_\mathrm{b}} \begin{pmatrix} (x_{\mathrm{o},k}^\mathrm{b} - \Delta t\, v_k)\cos(\Delta t\,\omega_k) \\ +y_{\mathrm{o},k}^\mathrm{b}\sin(\Delta t\,\omega_k) \\ -(x_{\mathrm{o},k}^\mathrm{b} - \Delta t\, v_k)\sin(\Delta t\,\omega_k) \\ +y_{\mathrm{o},k}^\mathrm{b}\cos(\Delta t\,\omega_k) \end{pmatrix} \quad (3)$$

where Δt is given also in Teodorescu et al. (2019). The saturation involved in (3) is artificial, since the vehicle is not moving in any confined environment. However, it is necessary for SDP implementation, where all state variables need to stay on a finite grid. Consequently, the Cartesian space around the vehicle is bounded by a sufficiently large Ω_b-set which can have an arbitrary shape (a circle or square etc.). In particular, in this work we will set the boundaries of Ω_b to be consistent with the maximum custom range of the ultrasonic sensors (in the order of a few meters).

3. SUPERVISORY CONTROL

This section is entirely dedicated to the control block from Fig. 1. We start by recalling the Stochastic Dynamic Programming (in general), then formulate two optimization problems for safely driving the vehicle. In agreement with the terminology in automotive field, the *longitudinal control* problem involves optimally avoiding to crash into an obstacle situated in front of the vehicle; the *lateral control* problem consists in optimally steering away from an obstacle situated to one side of the vehicle. In robotics, these two problems are known as the translational and rotational control problems, respectively.

Stochastic Dynamic Programming: This method allows to solve an optimization problem (see Bertsekas (2005)):

$$\min_{u_0,..,u_{N-1}} \underset{w_0,..,w_{N-1}}{\mathrm{E}} \left\{ \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k) + g_N(x_N) \right\} \quad (4)$$

such that equality constraints $x_{k+1} = f(x_k, u_k, w_k)$ and inequality constraints are fulfilled. Standard notations have been used here: x_k is the vector of state variables; $u_0,..,u_{N-1}$ are each vectors of the control variables at different sampling time instants $k = 0,..,N-1$; the uncertainty variables are the vectors $w_0,..,w_{N-1}$; $f(\cdot)$ is a function governing the dynamics of the system; N is the time horizon.

In general, the solution of (4) cannot be calculated (pursued) analytically. Instead, it is computed numerically by first placing each variable on a finite grid:

$$\begin{aligned} &x_k \in \{x^1, x^2, .., x^{N_x}\}; \quad u_k \in \{u^1, u^2, .., u^{N_u}\} \\ &w_k \in \{w^1, w^2, .., w^{N_w}\} \end{aligned} \quad (5)$$

with N_x, N_u, N_w the total number of points on the state vectors grid, the control actions grid and the uncertainties grid, respectively. The superscript indicates the index of the state. Then, apply Bellman's principle of optimality – see Bertsekas (2005, §2): minimize the expected transition cost, which proceeds backwards in time $k = N-1,..,0$ from iteration $k+1$ to k:

$$J_k(x_k) = \min_{u_k \in \Omega_{u_k}} \underset{w_k}{\mathrm{E}} \{ g_k(x_k, u_k, w_k) + J_{k+1}(x_{k+1}(x_k, u_k, w_k)) \}$$

with initial condition $J_N(x_N) = g_N(x_N)$, and $g_k(\cdot)$ is the transition cost. This process is depicted on a simplified toy (dummy) example in Fig. 4. To the right-hand side of $k+1$ instant, one can notice the optimal paths associated to the cost-to-go $J_{k+1}(x_{k+1})$. Between the time instants $k+1$ and k the actual minimization takes place using three control law candidates u^1, u^2 and u^3, and averaging the effect of two uncertainties w^1 and w^2. We have crossed out (using a red x) the two candidate trajectories associated to u^1 and u^2 in order to symbolize the fact that they are suboptimal; u^3 is the optimum.

We end this general presentation of SDP by mentioning some practical aspects. Firstly, the computation (extraction process) of the optimal control laws takes significant amount of time (see later on in this section) making it well suited for *offline* computation. Second, it suffers from the well-known *curse of*

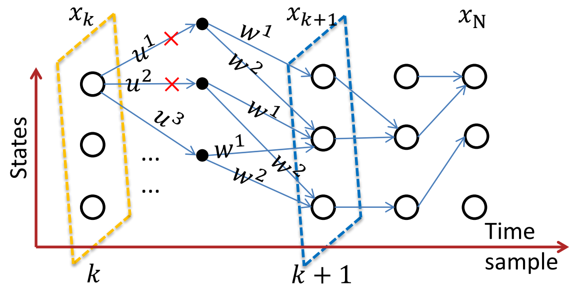


Fig. 4. SDP algorithm proceeding backwards in time – figure adapted from Bertsekas (2005, §2).

dimensionality (exponential increase of the required computation with each new state, control variable or uncertainty): compact-sized problems are preferred over large ones. Third, stationary control laws $u_0 = u_1 = .. = u_{N-1}$ are interesting, since they depend only on state variables vector (and thus are independent of time), making them usable for *online*, real-time implementation.

Longitudinal control: In this configuration we use state variables $x_k = (v_k, v_{\mathrm{d},k}, x^\mathrm{b}_{\mathrm{o},k})$, uncertainty $w_k = w_{v_{\mathrm{d},k}}$ and control action $u_k = v_{\mathrm{u},k}$. We make the assumptions that vehicle advances only forward ($v_k \geq 0$) and does not rotate ($\omega_k \equiv 0$ and $\omega_{\mathrm{d},k} \equiv 0$); consequently, $y^\mathrm{b}_{\mathrm{o},k}$ is constant. This gives a total of three state equations translating system dynamics (1)-(3). We formulate a *safe stop* strategy by weighting in the continuous-time transition cost *safety* (the first term) and supervisory corrective action (second term):

$$g(x(t), u(t), w(t)) = -\dot{x}^\mathrm{b}_\mathrm{o}(t) + \beta_1(x^\mathrm{b}_\mathrm{o}(t))\,|v_\mathrm{u}(t)| \qquad (6)$$

The meaning becomes clear once we think of the cumulative effect. The sum in (4) corresponds to an integrator in continuous time, and $\int_{t_0}^{t_\mathrm{f}} \dot{x}^\mathrm{b}_\mathrm{o}(\tau) = x^\mathrm{b}_\mathrm{o}(t_\mathrm{f}) - x^\mathrm{b}_\mathrm{o}(t_0)$ is the total traveled distance by the obstacle, between an initial instant t_0 and a final one t_f. When the obstacle approaches the vehicle, this quantity is negative, and when the obstacle moves away from the vehicle this quantity is positive. In *forward only* driving, this quantity can only be ≤ 0. This explains the minus sign in front of the first term in (6). The second term in (6) represents a penalty on control law deviations with respect to driver's intention. The weight function $\beta_1(\cdot)$ is chosen to be a polynomial of degree ≥ 2 in the variable $x^\mathrm{b}_{\mathrm{o},k}$ translating a desired dominance of the first term (safety concerns) when obstacles are very near. Using (3) with assumption $\omega_k \equiv 0$, it is not a surprise that discretizing the first term of (6) gives $\dot{x}^\mathrm{b}_{\mathrm{o},k} \approx (x^\mathrm{b}_{\mathrm{o},k+1} - x^\mathrm{b}_{\mathrm{o},k})/\Delta t = -v_k$ (see Fig. 5). This translates the fact that, in this configuration, the linear velocity of the obstacle is exactly the same as the wheelchair's linear velocity, but with opposite sign. the Consequently, the discrete-time transition cost is

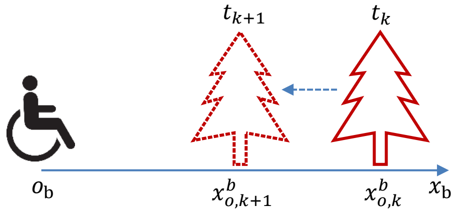
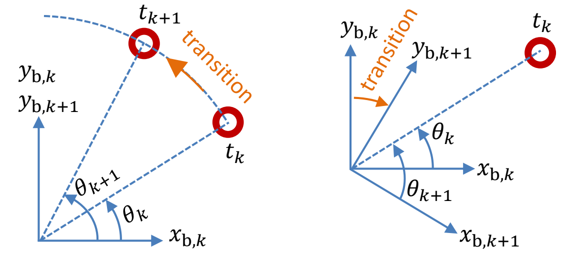


Fig. 5. Relative motion of the obstacle (the red tree) with respect to base frame.

$$g_k(v_k, v_{\mathrm{d},k}, x^\mathrm{b}_{\mathrm{o},k}, v_{\mathrm{u},k}) = \sigma_1 v_k + \sigma_2(v_{\mathrm{d},k} + v_{\mathrm{u},k}) + \beta_1(x^\mathrm{b}_{\mathrm{o},k})\,|v_{\mathrm{u},k}|$$

The inequality constraints of this optimization problem are as follows. Control law candidates $v_{\mathrm{u},k}$ should be chosen so that *predicted* velocity $v_{\mathrm{k+1}}$ as well as total *required* velocity $v_{\mathrm{u},k} + v_{\mathrm{d},k}$ should stay within the boundaries of $\Omega_{v\geq 0,\omega=0}$ from (2) and Fig. 2b. In particular, the latter condition will avoid the situation when driver demands to move forward and the *Supervisory control* responds by moving the wheelchair backwards, thus contradicting driver's expectation.

Lateral control: This problem is slightly more complex than the previous one. It involves one more state variable, making a total of four, $x_k = (\omega_k, \omega_{\mathrm{d},k}, x^\mathrm{b}_{\mathrm{o},k}, y^\mathrm{b}_{\mathrm{o},k})$ with associated state equations (1)-(3). The uncertainty is $w_k = w_{\omega_{\mathrm{d},k}}$ and the control variable is $u_k = \omega_{\mathrm{u},k}$. The vehicle is allowed to rotate only. To be more precise, we make the first assumption that the linear velocity of the vehicle is positive but very small $v_k \in (0, \varepsilon)$ with ε an arbitrarily small number. This makes dynamics of v_k negligible in (2), so we can disregard it, but also encourages us to formulate an optimal *safe turn* strategy by taking into account an obstacle of coordinates $(x^\mathrm{b}_{\mathrm{o},k}, y^\mathrm{b}_{\mathrm{o},k})$ situated towards the front of the vehicle. The second assumption is that this obstacle is situated towards the left-hand side of the vehicle: consequently, the vehicle should turn towards the right, heading away from the danger.



(a) Relative obstacle motion viewed with respect to two overlapped base frame instants

(b) Absolute obstacle position with respect to the fixed inertial frame

Fig. 6. View from above: perception of the nearby obstacle (in red) as the vehicle rotates.

As the vehicle turns in Cartesian space, the obstacle is perceived as rotating with an angle θ_k around the vehicle, in a circular motion. This is illustrated in Fig. 6. For safety reasons, we want at the next iteration, θ_{k+1} to be higher, thus heading away from the obstacle. By subtracting the two quantities, we end up with the optimization problem of minimizing $\theta_{k+1} - \theta_k = \Delta t\,\omega_{k+1}$. Hence, the transition cost is chosen to be

$$g_k(\omega_k, \omega_{\mathrm{d},k}, x^\mathrm{b}_{\mathrm{o},k}, y^\mathrm{b}_{\mathrm{o},k}, \omega_{\mathrm{u},k}) = \Delta t\left(\sigma_3\omega_k + \sigma_4(\omega_{\mathrm{d},k} + \omega_{\mathrm{u},k})\right) + \beta_1(d^\mathrm{b}_{\mathrm{o},k})\,|\omega_{\mathrm{u},k}|$$

where $d^\mathrm{b}_{\mathrm{o},k} = ||\left(x^\mathrm{b}_{\mathrm{o},k}\ y^\mathrm{b}_{\mathrm{o},k}\right)||_2$ is the distance to the obstacle. Note the same tuning function $\beta_1(\cdot)$ from section "Longitudinal control" was used here, ensuring trade-off between the two terms. The second term acts as penalty to deviations with respect to driver's demands.

The inequality constrains are also very similar to section "Longitudinal control". Candidates control values $\omega_{\mathrm{u},k}$ should be chosen so that *predicted* velocity $\omega_{\mathrm{k+1}}$ as well as total *required* velocity $\omega_{\mathrm{u},k} + \omega_{\mathrm{d},k}$ should stay within the boundaries of $\Omega_{v=0,\omega}$

from (2) and Fig. 2b. The first condition is imposed by feasibility concerns and the second is to avoid contradicting driver's expectation (when the driver demands to turn right, the *Supervisory Control* will not yield an action making the wheelchair rotate towards the left).

Joint longitudinal and lateral control: In order to achieve simultaneous control over the (v, ω) variables, we propose to combine together the two control laws. This is a straightforward option allowing safe navigation in an environment with obstacles. Although effective in practice, this solution does not take into account coupling effects between the two (v, ω) variables.

As future work, we would like to draft a few elements of the coupled (v, ω) problem. This new optimization problem would have to deal with a dynamic system with 6 state variables $x_k = (v_k, v_{\mathrm{d},k}, \omega_k, \omega_{\mathrm{d},k}, x^{\mathrm{b}}_{\mathrm{o},k}, y^{\mathrm{b}}_{\mathrm{o},k})$ and thus all 6 state equations (1)-(3), two uncertainties $w_k = (v_{\mathrm{d},k}, w_{\omega_{\mathrm{d},k}})$ and two control variables $u_k = (v_{\mathrm{u},k}, \omega_{\mathrm{u},k})$. This elegant multiple-input multiple-output formulation will suffer from one major drawback: the well-known *curse of dimensionality* mentioned in the Introduction, making it very expensive in terms of offline computation power.

Computation time: One of the most important aspects (a key point) when implementing the SDP algorithm is to use a grid (5) that is sufficiently fine so that transitory dynamics is properly captured by the optimization problem and consequently transitory effects can be observed in the resulting data. On a modern workstation (Dell T320 with Intel CPU Xeon E5-2407 running at 2.4 GHz), the time to compute the numerical solution of $v_{\mathrm{u},k}$ from section "Longitudinal control" (a 3-state-variable system), is in the order of a few days. To compare with, it takes 10 times more to compute $\omega_{\mathrm{u},k}$ from section "Lateral control" (a 4-state-variable system), namely in the order of a few weeks. A 6-state-variable system would require substantially more time of non-stop operation to compute the solution (possibly in the scale of years).

Sensors: To summarize the results of section 3, we now have two control laws

$$v_{\mathrm{u},k}(v_k, v_{\mathrm{d},k}, x^{\mathrm{b}}_{\mathrm{o},k}) \quad \text{and} \quad \omega_{\mathrm{u},k}(\omega_k, \omega_{\mathrm{d},k}, x^{\mathrm{b}}_{\mathrm{o},k}, y^{\mathrm{b}}_{\mathrm{o},k}) \qquad (7)$$

in the form of lookup tables (matrices). This include (v_k, ω_k) which are estimated from the custom-made drive wheel encoders, as well as $(v_{\mathrm{d},k}, \omega_{\mathrm{d},k})$ given by the standard joystick. On the other hand, there is no sensor capable of measuring $(x^{\mathrm{b}}_{\mathrm{o},k}, y^{\mathrm{b}}_{\mathrm{o},k})$. Instead, we have to rely on distance measurements from the ultrasonic sensors, placed in an array all around the vehicle. Each sensor has a theoretical beam width (coverage) of 45°and measures the distance $d^{\mathrm{b}}_{\mathrm{o},k}$ to the closest obstacle within range. However, the exact location of the obstacle is unknown: it can be anywhere on the the arc spanning from one side of the beam to the other end (the entire 45°range). Since this information is not accessible through sensor measurements, we propose to proceed to a reduction (folding) in the space of variables: for each $d^{\mathrm{b}}_{\mathrm{o},k}$ we take the average value on the arc mentioned above.

Finally, this allowed us to reexpress the control matrices from (7) as $\mathbf{v}_{\mathrm{u},k}(v_k, v_{\mathrm{d},k}, d^{\mathrm{b}}_{\mathrm{o},k})$ and $\boldsymbol{\omega}_{\mathrm{u},k}(\omega_k, \omega_{\mathrm{d},k}, d^{\mathrm{b}}_{\mathrm{o},k})$. Note that all dependent state variables are now accessible via sensor measurements. These control matrices will be tested in the next section.

4. SIMULATIONS

The stochastic framework depicted in Fig 1 provides not only the means to design control, but also to simulate the overall behavior. Given a known probability distributions of the driver intention, they can be used to simulate possible realizations. In Fig 8, the vehicle is fitted with 3 ultrasonic sensors mounted under the footplate on the front, each sensor covering a sector of 45°. The *longitudinal control* will use the distance measurement coming from the sensor covering region 1, while the *lateral control* will use the other two sensors covering regions 2 and 3. In this section we will use two driver models: one deterministic and the other one stochastic (the *blind driver* model). The vehicle can advance only forward (not backwards) and can rotate freely.

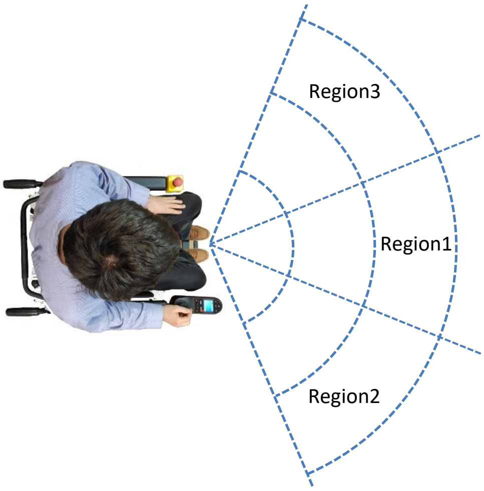


Fig. 7. View from above: wheelchair fitted with 3 ultrasonic sensors, each covering a separate region.

Now, we want to test the *blind driver* model. Fig 9 shows a random sequence of uniformly distributed points within $\Omega_{v_\mathrm{d},\omega_\mathrm{d}}$. We simulated the outcome of these driver demand points $(v_\mathrm{d}, \omega_\mathrm{d})$ on the vehicle in Fig 10a. This trajectory is entirely arbitrary and lasts 20 seconds. Notice a few obstacles were hit. As expected, once we enable *supervisory control* in Fig 10 no more obstacles were hit and the vehicle advances much more cautiously (the total traveled distance is reduced).

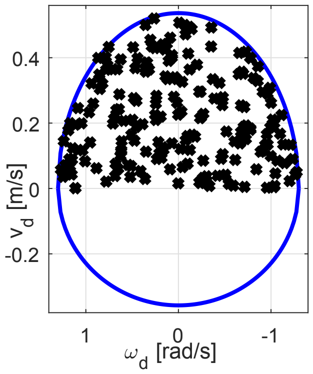
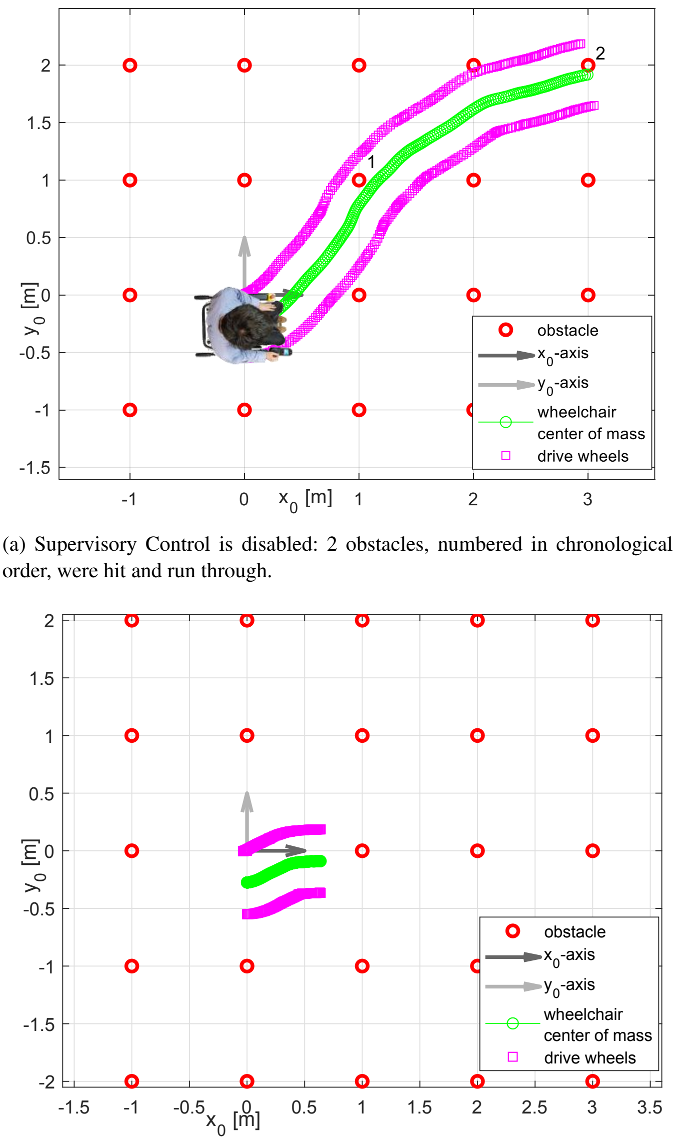


Fig. 8. Driver demand according to the *blind driver* model: a realization with random occurrence points (in black) translating forward vehicle motion and free (unconstrained) rotation.



(a) Supervisory Control is disabled: 2 obstacles, numbered in chronological order, were hit and run through.

(b) Supervisory Control is enabled

Fig. 9. View from above: Vehicle's trajectory issued by the *blind driver* model, in an environment with obstacles. Initial, starting position is where the driver is sitting (same for both figures).

5. CONCLUSION

This paper presented a stochastic framework for control design and simulation, based on Stochastic Dynamic Programming (SDP). The *supervisory control* assists the driver of a vehicle (a wheelchair) by correcting his actions, thus ensuring safe driving (navigation) in an environment with static obstacles. The feasibility of the concept is assessed by running simulations. They show the ability of the vehicle to safely navigate using a combination of longitudinal and lateral control.

This methodology is quite modular, allowing subsequent update of any of the four blocks of this stochastic framework: the driver intention model, wheelchair dynamics, the environment model, and the supervisory control. The major drawback of SDP in general, and of our implementation in particular, is the computation time. Therefore, as future work we intend to implement a *policy iteration* algorithm, due to its property of monotonic convergence towards the optimal solution.

REFERENCES

Abbink, D.A., Carlson, T., Mulder, M., de Winter, J.C., Aminravan, F., Gibo, T.L., and Boer, E.R. (2018). A topology of shared control systems – finding common ground in diversity. *IEEE Transactions on Human-Machine Systems*, 48(5), 509–525.

Bertsekas, D. (2005). *Dynamic programming and optimal control*, volume 1. Athena Scientific, Belmont, MA, third edition.

Cook, J.A., Sun, J., Buckland, J.H., Kolmanovsky, I.V., Peng, H., and Grizzle, J.W. (2006). Automotive powertrain control – a survey. *Asian Journal of Control*, 8(3), 237–260.

Demeester, E., Hüntemann, A., Vanhooydonck, D., Vanacker, G., Van Brussel, H., and Nuttin, M. (2008). User-adapted plan recognition and user-adapted shared control: A Bayesian approach to semi-autonomous wheelchair driving. *Autonomous Robots*, 24(2), 193–211.

Ghorbel, M., Pineau, J., Gourdeau, R., Javdani, S., and Srinivasa, S. (2018). A decision-theoretic approach for the collaborative control of a smart wheelchair. *Int. Journal of Social Robotics*, 10(1), 131–145.

Johannesson, L., Asbogard, M., and Egardt, B. (2007). Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming. *IEEE Trans. Intell. Transp. Syst*, 8(1), 71–83.

Kolmanovsky, I., Siverguina, I., and Lygoe, B. (2002). Optimization of powertrain operating policy for feasibility assessment and calibration: Stochastic dynamic programming approach. In *American Control Conf.*, 1425–1430. Anchorage, AK.

Lin, C.C., Peng, H., and Grizzle, J.W. (2004). A stochastic control strategy for hybrid electric vehicles. In *American Control Conf.*, 4710–4715. Boston, MA.

Liu, K., Wang, Y.Y., Haskara, I., Chang, C.F., Girard, A., and Kolmanovsky, I. (2018). Stochastic policies for online computation triggering in powertrain control. In *ASME 2018 Dynamic Systems and Control Conference*, V001T01A007–V001T01A007. Atlanta, Georgia, US.

Rutquist, P. (2002). Optimal control for the energy storage in a hybrid electric vehicle. In *Proc. 19th Int. Battery, Hybrid and Fuel Cell Electric Vehicle Symp. and Exhib. (EVS19)*, 1133–1139. Busan, South Korea.

Rutquist, P. (2017). *Methods for Stochastic Optimal Control under State Constraints*. Ph.D. thesis, Chalmers University of Technology, Göteborg, Sweden.

Simpson, R.C. (2005). Smart wheelchairs: A literature review. *Journal of Rehabilitation Research and Development*, 42(4), 423–436.

Spong, M.W., Hutchinson, S., and Vidyasagar, M. (2005). *Robot Modeling and Control*. John Wiley & Sons, Hoboken, NJ.

Teodorescu, C.S., Zhang, B., and Carlson, T. (2019). Probabilistic shared control for a smart wheelchair: A stochastic model-based framework. In *2019 IEEE International Conference on Systems, Man, and Cybernetics*. Bari, Italy.

Ye, N., Somani, A., Hsu, D., and Lee, W.S. (2017). DESPOT: Online POMDP planning with regularization. *Journal of Artificial Intelligence Research*, 58, 231–266.