Probabilistic Shared Control for a Smart Wheelchair: A Stochastic Dynamic Programming Approach*

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

Abstract— **This article presents progress made towards implementing a shared control framework for a smart wheelchair based upon stochastic dynamic programming. First, we describe the mechanical, electrical and software design process of our instrumented wheelchair platform. Then, we detail a deterministic control-oriented model of the wheelchair motion dynamics using Euler-Lagrange equations. This is followed by the development of a stochastic model of the human driver's intention using Markov chain. Finally, we end our contribution with a discussion of the future implementation and evaluation of our stochastic dynamic programming.**

I. INTRODUCTION

The area of shared control has a growing body of research. It is interesting to note that much of this research employed a technique called *linear blending* which combines linearly the user's demand and the actions of a high-level algorithm. Essentially, the final command is a weighted sum between the two, and various methods have been proposed to optimize these weights [1]–[4]. Trautman demonstrated theoretically that a probabilistic approach is superior to linear blending in terms of safety: linear blending does not guarantee safety mathematically [5]. To investigate this issue, Ezeh *et al.* implemented a probabilistic shared control (PSC) approach where the user's trajectory and the path planner's trajectory are modelled as a joint probability distribution [6]. Although experiments with PSC showed some improvement over linear blending in specific circumstances, they were not as substantial as we would have expected.

Two possible reasons may account for this. Firstly, although the kinematic limitations (namely maximum velocity and acceleration) of the robot have been taken into account according to the dynamic window approach [7] algorithm, the wheelchair's dynamics were not modeled explicitly. Secondly, the user intention model may have been overly simplistic or too general (broad), unable to capture the user's complex intentions. In particular, a memoryless implementation was used in [6]. In order to solve these issues, we propose here a stochastic dynamic programming approach where a wheelchair dynamic model is formulated and validated in simulation. In addition, the user's intention is estimated using Markov chain modeling, where the present information is used in order to predict the future, at the next sample.

A. A smart wheelchair

Assistive technologies are key elements intended to meet the needs of people with chronic disabilities. They facilitate social inclusion, increase self-esteem and autonomy to carry out everyday tasks.

This article presents progress made in building a new research platform: an instrumented electric-powered wheelchair. We used a commercial Sunrise Medical Salsa M2 model and added off-the-shelf sensors to it (ultrasonic sensors, encoders, electric current sensor). The result can be seen in Fig. 1.

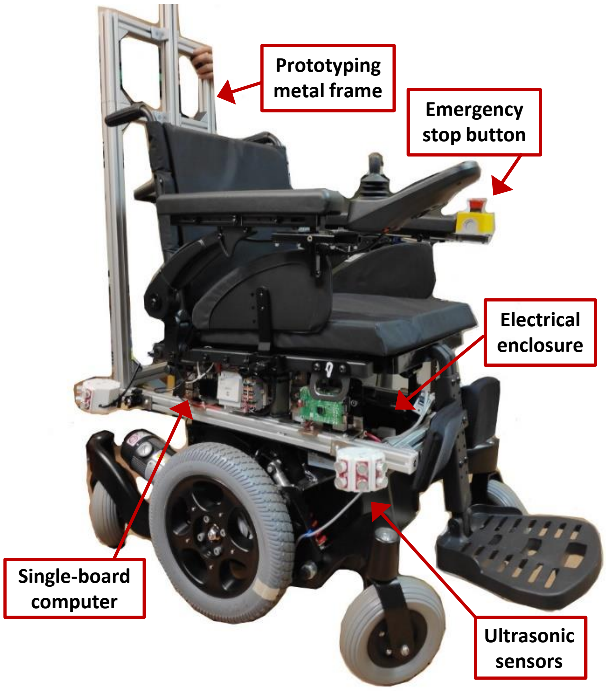


Fig. 1: Instrumented wheelchair

The final goal is to convert the instrumented wheelchair into a semi-autonomous vehicle capable of ensuring safe ride to the human driver using a PSC.

B. Stochastic Dynamic Programming

A careful selection among candidate algorithms capable of tackling our complex control problem led us to Stochastic Dynamic Programming (SDP) [8]. We believe SDP is the right candidate for the following general reasons. A stochastic optimization problem with nonlinear constraints is solved offline. The outcome is an explicit analytical solution

*This work was supported by Interreg FCE project ADAPT number 116

[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

Manuscript 253 submitted to 2019 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Received March 30, 2019.

$u(x)$ (a lookup table) that can be implemented in real-time on embedded electronics. Actually, the resulting control law is deterministic and can be used for extensive offline analysis (e.g. check reliability, safety etc.), prior to online implementation. The methodology is industry-ready [9], [10].

In particular, our work was largely inspired by the SDP methodology used in [9]. Let us highlight a list of common features between [9] and our current work:

- in terms of problem formulation, the end goal (target) is not a fixed point in the Cartesian space. This is a typical path planning requirement, which is a different problem.
- the path to follow by drivers is not predefined upfront. Instead, it is a random ride in a circuit with obstacles (drivers may choose by themselves where to go).
- plant dynamics (mechanical, electrical) is properly modeled and hence not neglected (by making use of the assumption that transients are fast enough so that steady-state is reached between successive time samples).

On the other hand, in this work we have to make the effort to translate and formulate a new optimal control problem. Below is a list of significant differences:

- the framework in [9] is specific to the automotive sector, while here we target assistive technologies. In particular the vehicle in [9] is intended to be autonomous, while here we are interested in semi-autonomous vehicles.
- the optimization problem is different: authors in [9] are interested in the power balance between multiple power sources (a battery versus internal combustion engine). The cost function accounts for fuel consumption and pollutants. In this work, we are interested in obstacle avoidance.
- the mathematical formulation in [9] takes into account one stochastic variable (namely the linear velocity of the vehicle), while here we have two stochastic variables (the linear velocity of the vehicle, as well as its angular velocity). In other words the motion in [9] is 1-dimensional (1D), along a line, while here we are interested in a 2-dimensional (2D) planar motion.
- there is no awareness of the environment in [9], while here we use ultrasonic sensors.
- plant model dynamics are defined by a second order ordinary differential equation (ODE) system in [9], while here we have a nineth order ODE. Contrary to [9], no quasi-static modeling is used here.

The block diagram in Fig. 2 is central to this work. It summarizes the requirements for SDP: control design as well as simulation. In the following, v and ω stand for the wheelchair's linear velocity and angular velocity, respectively. Fig. 2 illustrates a 2-layer control strategy. The driver expresses his intention by moving the joystick. There is a one-to-one relation (that can be experimentally identified) between positions on the joystick's plane and the demanded linear velocity v_{dem} and demanded angular velocity ω_{dem}. In this article, we prefer to work with velocities instead of joystick positions. They will be sent to the supervisory control (terminology according to [9]), which in turn will calculate an optimal decision based on environment awareness (distance to obstacles) and send it as reference linear velocity v_{ref} and reference angular velocity ω_{ref} to the wheelchair dynamics block. To summarize, another way to see Fig. 2 is that supervisory control filters (in a broad sense) signals coming from the driver by incorporating environment awareness (obstacles).

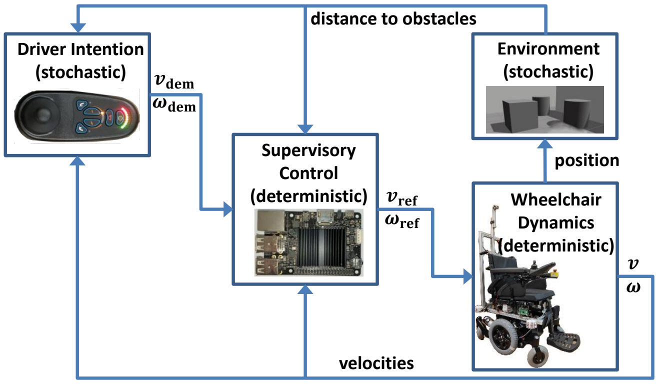


Fig. 2: Stochastic model overview.

In this paper we will focus on two blocks of Fig. 2 (namely *Wheelchair Dynamics* and *Driver Intention*) and leave the other two for future work.

This article is organized as follows. In section II we present the architecture of the instrumented wheelchair. Then, section III presents the deterministic wheelchair dynamic model using Euler-Lagrange method which is validated in simulation. Next, analytic modeling of the human driver intention is performed in section IV. A discussion follows in section V highlighting the link towards future work. Finally, the paper ends with conclusions.

II. ARCHITECTURE

In order to build a stable, reliable and reusable research platform (an instrumented wheelchair) it is important to address properly all the three design criteria: mechanical, electrical and software design. They will be presented next.

A. Mechanical design

For prototyping, we used an aluminium frame directly attached to the base of the wheelchair (see Fig. 1). It offers the ability to easily slide and mount other components, like cable trunking, DIN rail, sensors etc. Each rod ends with a 3D printed rounded end cap used to cover the sharp edge and thus avoid injuries.

Encoders: We opted for a minimally invasive compact design for mounting encoders, type Kubler 8.KIS40.1362.0500. In Fig. 3 the reader can identify their position, in the narrow space available on the inside of the main wheel. Both the housing as well as the pulley attached to the main wheel (110 teeth, pitch 2.5 mm) were 3D printed using PLA material.

B. Electrical design

Fig. 1 shows the selected position for a single-board computer (Odroid C2 or Raspberry Pi 3b+), namely to one side of the wheelchair. This position allows efficient usage of

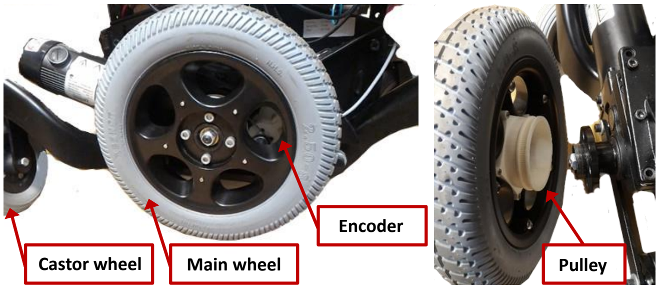


Fig. 3: Mechanical design: encoder mount.

the build-in wifi. Most of the other electronics were placed inside a modular electrical enclosure (see Fig. 4) which sits under the wheelchair seat, above the battery in normal operation. It acts as a Faraday cage, thus protecting the sensitive electronic components situated inside from electromagnetic interference, mainly coming from the motors.

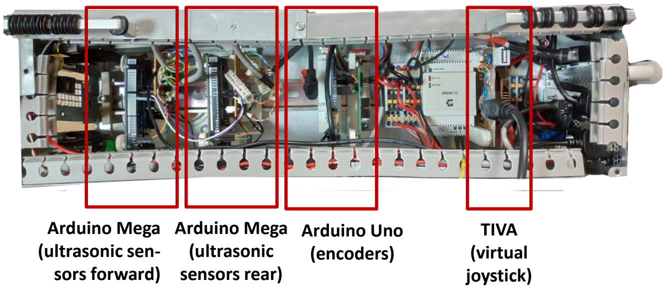

Arduino Mega (ultrasonic sensors forward)
Arduino Mega (ultrasonic sensors rear)
Arduino Uno (encoders)
TIVA (virtual joystick)

Fig. 4: Electrical design: close-up of the electrical enclosure

1) Ultrasonic sensors: In Fig. 1, two out of four ultrasonic sensor clusters are visible. Each cluster is equipped with three SRF08 sensors. The two forward clusters are connected to one Arduino Mega and the rear clusters are connected to another Arduino Mega. Both Arduinos sit inside the electrical enclosure, see Fig. 4. These ultrasonic sensors communicate via I2C (i.e. TWI) bus with the Arduinos. Below are some important practical considerations.

Dimensioning: One I2C network consists of one Arduino Mega and 6 ultrasonic sensors. The lumped capacitance was experimentally measured to be 260 pF, which is well below the I2C bus limit specification of 400 pF. Our experience shows that beyond it, frequent errors occur. Extra 2.2kΩ pull-up resistors needed to be added in parallel to the built-in 10kΩ present inside the SDA and SCL lines of Arduino Mega.

2) Electrical enclosure: Apart from the two Arduino Mega synchronized using serial communication and responsible for firing ultrasonic sensors, the electrical enclosure also contains:

- an Arduino Uno plus a hardware counter receiving signals from the two wheel encoders;
- a TIVA board acting as *virtual joystick* and allowing taking control of the wheelchair from a ROS node. The ROS node reads a ROS topic with user defined velocities and converts it into a low-level R-Net bus message, sent to the power module. The latter will interpret it as a signal coming from a joystick. This prototype was developed by the team of INSA Rennes [4].

3) Electric current sensor: Two inexpensive CZH-LABS D-1085 sensors, capable of measuring up to 50 Ampere, were mounted on an inner DIN rail, under the wheelchair seat, above the battery: see Fig. 5. They measure current flowing from the power module to each motor. Later on, we will use them for indirect torque estimation.

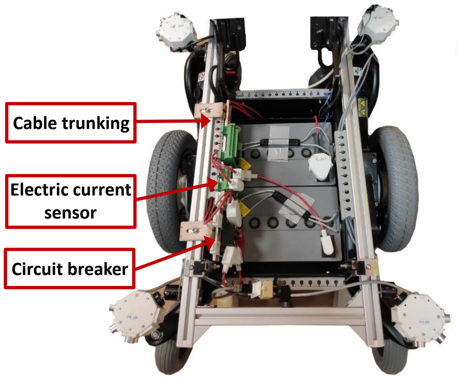


Fig. 5: Electrical design: components attached to the inner side of the prototyping metal frame

C. Software

We opted for Robot Operating System (ROS) as middleware, since it allows to flexibly interconnect sensors and collect data. Fig. 6 illustrates a schematic diagram of the instrumented wheelchair architecture. In the middle sits the single-board computer acting as ROS Master.

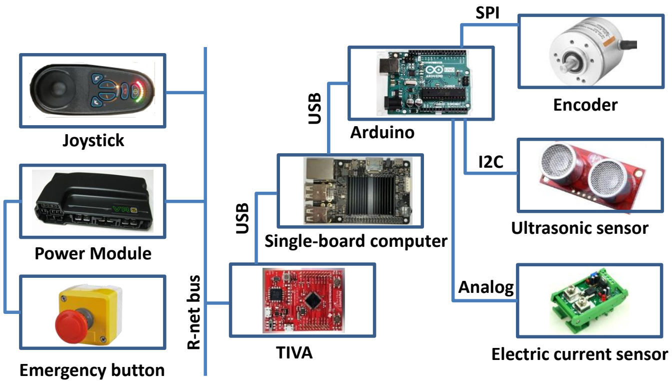
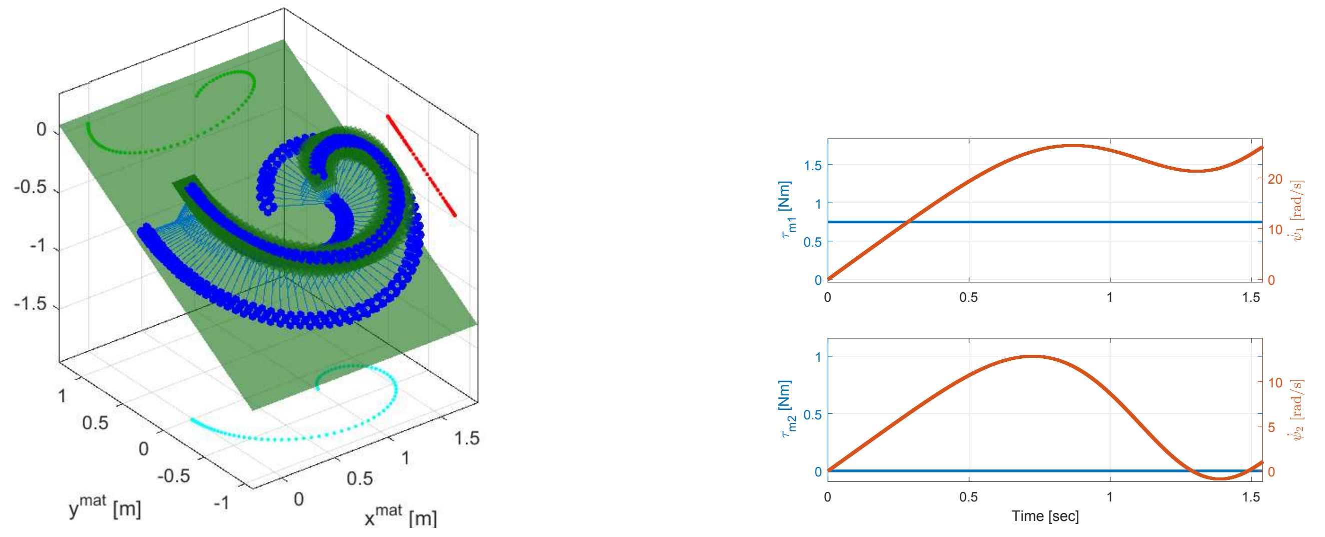


Fig. 6: Architecture of the instrumented wheelchair.

III. Modeling Wheelchair Dynamics

In this work, the wheelchair is sitting on an inclined plane (a ramp) with angle $\alpha(t)$. Notice this angle does not need to be fixed, instead, the orientation of the slope with respect to the inertial frame needs to be fixed.

In order to build a dynamic model, we separate rotating parts from the static ones. We modeled the entire system



(a) in blue: trajectories of the 3 centers of mass: the two main wheels (of mass m_{w}) and the wheelchair body including the driver (mass m_3)

(b) details of applied torque and wheels velocity profiles

Fig. 7: Wheelchair motion down a slope with $\alpha = 30°$; actuators run in open loop control $(\tau_{\mathrm{m1}}, \tau_{\mathrm{m2}}) = (0.75, 0)$ Nm.

wheelchair plus driver as three distinct objects, rigidly connected between them:

- the two main wheels as vertical rolling disks [11, §1.4]. We neglected the other 4 castor wheels.
- the wheelchair body including the driver as an uniform rectangular box.[1]

We made the assumptions regarding each of the three aforementioned parts:

- each mass is concentrated at their individual center of mass.
- links connecting parts are rigid (not flexible).
- wheels are rolling without slipping.

This allowed us to build a lumped mass system as follows.

A. Euler-Lagrange model

The Lagrangian is taken to be the total kinetic energy of the system minus the potential energy (gravity effect) [11]–[13]. On the one hand we have the effect of the two main wheels (see the first line of the following formula) and on the other hand, the wheelchair body including the driver (see the second line of the following formula):

$$L = \sum_{i=1}^{2}\left\{\frac{1}{2}m_{\mathrm{w}}\begin{pmatrix}\dot x_i & \dot y_i & \dot z_i\end{pmatrix}'\begin{pmatrix}\dot x_i\\ \dot y_i\\ \dot z_i\end{pmatrix} + \frac{1}{2}I_{\mathrm{w,zz}}\dot\psi_i^2 - m_{\mathrm{w}}g_{\mathrm{a}}z_i\right\}$$
$$+\frac{1}{2}m_3\begin{pmatrix}\dot x_3 & \dot y_3 & \dot z_3\end{pmatrix}'\begin{pmatrix}\dot x_3\\ \dot y_3\\ \dot z_3\end{pmatrix} + \frac{1}{2}I_{\mathrm{zz}}\frac{r^2}{l_{\mathrm{w}}^2}(\dot\psi_1-\dot\psi_2)^2 - m_3 g_{\mathrm{a}} z_3$$

where (x_i, y_i, z_i) are Cartesian coordinates of each center of mass with respect to the inertial frame, $i = 1, 2, 3$. Their velocities are calculated as follows:

$$\dot x_i = v_i\cos\alpha\,\cos\left(\frac{r}{l_{\mathrm{w}}}(\psi_1-\psi_2)\right) \tag{1a}$$

$$\dot y_i = v_i\sin\left(\frac{r}{l_{\mathrm{w}}}(\psi_1-\psi_2)\right) \tag{1b}$$

$$\dot z_i = v_i\sin\alpha\,\cos\left(\frac{r}{l_{\mathrm{w}}}(\psi_1-\psi_2)\right) \tag{1c}$$

where

$$v_1 = \dot\psi_1 r;\quad v_2 = \dot\psi_2 r;\quad v_3 = (\dot\psi_1+\dot\psi_2)\,r/2$$

and ψ_1, ψ_2 are angular displacements of the right and left main wheel, respectively. All parameters are indicated in Table I, together with their nominal values (most of them taken from datasheets).

Finally, derivations of the Euler-Lagrange equation lead to the second order system:

$$\begin{bmatrix} a & b\\ b & a\end{bmatrix}\begin{bmatrix}\ddot\psi_1\\ \ddot\psi_2\end{bmatrix} + c\begin{bmatrix}\dot\psi_1\\ \dot\psi_2\end{bmatrix} - \begin{bmatrix} d\\ d\end{bmatrix}\cos\left(\tfrac{r}{l_{\mathrm{w}}}(\psi_1-\psi_2)\right)$$
$$= e\begin{bmatrix}\tau_{\mathrm{m1}}\\ \tau_{\mathrm{m2}}\end{bmatrix} \tag{2}$$

where τ_{m1} and τ_{m2} are torques applied to the motors, and the parameters:

$$\begin{aligned} a &= m_{\mathrm{w}}r^2 + \tfrac{1}{4}m_3 r^2 + I_{\mathrm{w,zz}} + I_{\mathrm{zz}}\tfrac{r^2}{l_{\mathrm{w}}^2} + \eta_{\mathrm{G}}J_{\mathrm{m}}r_{\mathrm{G}}^2\\ b &= \tfrac{1}{4}m_3 r^2 - I_{\mathrm{zz}}\tfrac{r^2}{l_{\mathrm{w}}^2}\\ c &= \eta_{\mathrm{G}}B_{\mathrm{m}}\,r_{\mathrm{G}}^2\\ d &= \left(m_{\mathrm{w}} + \tfrac{m_3}{2}\right)g_{\mathrm{a}}\,r\sin\alpha\\ e &= \eta_{\mathrm{G}}r_{\mathrm{G}}\end{aligned} \tag{3}$$

[1] This might seem to our reader an overly simplistic premise, whereas a better model would follow the natural shape of the person sitting on the wheelchair. As we will see later on, only two parameters are necessary for this dynamic model: the mass (which is independent of the shape) and the moment of inertia. The latter can be either extracted from a realistic CAD design or experimentally identified as proposed here.

TABLE I: Parameters of the dynamic model

Symbol	Meaning	Nominal Value
m_{w}	mass of a main wheel	2.6 kg
r	radius of a main wheel	0.1651 m
$I_{\mathrm{w,zz}}$	moment of inertia of a main wheel	$\frac{1}{2}m_{\mathrm{w}}r^2$
l_{w}	length between the two main wheels	0.55 m
m_3	mass of wheelchair body including the driver	200 kg
I_{zz}	moment of inertia of wheelchair body including the driver	4.42 $\mathrm{kg\,m^2}$
r_{G}	motor gearbox ratio	26
η_{G}	motor gearbox efficiency	0.97
J_{m}	lumped moment of inertia of motor rotor and rotating parts of gearbox	$9.6\,10^{-5}$ $\mathrm{kg\,m^2}$
B_{m}	lumped rotational viscous damping	$1.85\,10^{-4}$ Nm s/rad
g_{a}	gravitational acceleration	9.81 $\mathrm{m/s^2}$

It is interesting to note in (2) the coupling effect between the dynamics of each degree-of-freedom ψ_i (with $i = 1,2$): note the place of the parameters a and b, as well as the argument of the cosine.

B. Experimental identification of parameters

Some parameters in system (2)-(3) can easily be obtained through direct measurements (e.g. mass, radius of wheels, lengths) or are supplied by the manufacturer (e.g. motor gearbox ratio). Other parameters are more difficult to infer, especially if datasheets are unavailable (e.g. moments of inertia). Luckily, they can be experimentally identified using the linearity in the parameters property [12, §7.5.3], [14]:

$$Y(\psi, \dot{\psi}, \ddot{\psi})\theta = e\,\tau_{\mathrm{m}}$$

where

- ψ, $\dot{\psi}$, $\ddot{\psi}$ are vectors representing displacement, velocity and acceleration, respectively; $\psi = \left(\psi_1, \psi_2\right)$;
- θ is the parameter vector consisting of products and/or sums of physical parameters (3);
- $\tau_{\mathrm{m}} = \left(\tau_{\mathrm{m}1}, \tau_{\mathrm{m}2}\right)$ is the vector of motor torques. In practice, a monotonic increasing function describes the relation between electric current flowing through the windings and motor torques. Often, it is approximated by a constant gain [12, §6.1].
- $Y(\psi, \dot{\psi}, \ddot{\psi})$ is the regressor matrix.

The above system is control-oriented. It is simple enough to be used in a real-time application, yet captures a great deal of physical phenomena. In particular, it can be used for:

- open-loop control by model inversion: specify desired trajectory profiles $\psi(t)$ and the model will indicate which are the associated motor torques $\tau_{\mathrm{m}}(t)$.
- indirect measurement of motor torques. making it a *software sensor*: measure in real-time ψ and its higher order derivatives $\dot{\psi}$ and $\ddot{\psi}$, pass them to the model and obtain torques τ_{m}.
- simulating correct overall behavior, as we shall detail in the next section.

C. Validation in simulation

In order to gain physical insight of system (2)-(3), we simulated the following scenario: a wheelchair is positioned heading down a ramp and starts advancing due to the joint action of gravity and constant torque applied to the right motor only ($\tau_{\mathrm{m}1} > 0, \tau_{\mathrm{m}2} = 0$). The trajectories of the three center of mass can be visualized in Fig. 7a. Notice the circular shape described by the wheelchair which is what we expected. Details of these trajectories are given in Fig. 7b. In particular, it is interesting to notice in the bottom plot of Fig. 7b, towards the end of the simulation, that the left wheel will decelerate to the point of starting to rotate backwards ($\dot{\psi}_2 < 0$). This happens as the wheelchair rotates and advances up the slope, and is exactly what one would notice in real world. This confirms the model follows correctly the physical laws.

D. Block model output

In this work, we consider the output of system (2)-(3) to be the linear velocity v and the angular velocity ω. They can easily be calculated using the linear relation:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = A \begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \end{bmatrix}, \text{with } A = r \begin{bmatrix} 0.5 & 0.5 \\ 1/l_{\mathrm{w}} & -1/l_{\mathrm{w}} \end{bmatrix}. \qquad (4)$$

Note that matrix A is invertible.

E. Block model input. Low-level control

In this section we are interested to achieve tracking of desired reference trajectories $(v_{\mathrm{ref}}, \omega_{\mathrm{ref}})$, which are specified by the supervisory control in Fig. 2. Actually, our decision to work with velocities as inputs is motivated and imposed by the input design requirement of the *virtual joystick* described in section II-B.2.

A simple typical choice for this type of robotics problem is the independent joint control [12, §6]. It relies on the assumption that coupling between each axis of the wheelchair is weak. By analyzing the parameter values of system (2)-(3) we confirm that this is indeed the case. Consequently, we use a Proportional-Integral (PI) controller to ensure tracking of the desired reference on each axis, as depicted in Fig. 8. The block named T stands for the transform (4) and the block named T^{-1} represents its unique inverse.

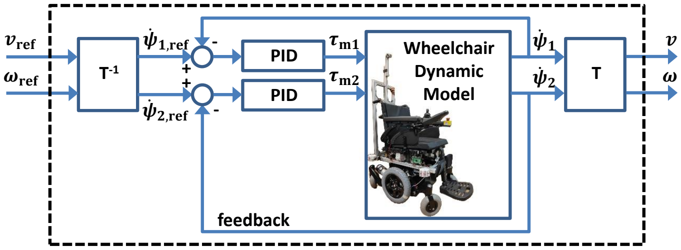


Fig. 8: Wheelchair model with low-level PI control.

Each PI is governed by the dynamics:

$$\dot{\tau}_{\mathrm{m}i} = K_{\mathrm{P}}\left(\ddot{\psi}_{i\,\mathrm{ref}} - \ddot{\psi}_i\right) + K_{\mathrm{I}}\left(\dot{\psi}_{i\,\mathrm{ref}} - \dot{\psi}_i\right) \tag{5}$$

where $i = 1,2$, and K_{P}, K_{I} are tuning parameters.

F. Augmented input-output system

In this section we highlight the structure of the black block depicted using dashed line in Fig. 8. It is the same one from Fig. 2 and is necessary later on, for control design and simulation. To summarize, it has:

- inputs: velocities $(v_{\mathrm{ref}}, \omega_{\mathrm{ref}})$;
- outputs: velocities (v, ω) from (4); position (x_3, y_3, z_3) from (1a)-(1c);
- dynamics written as a first order ordinary differential system accounting for the following ODEs: (2)-(3) with state variables $(\psi_1, \dot{\psi}_1, \psi_2, \dot{\psi}_2)$; (5) with state variables $(\tau_{\mathrm{m}1}, \tau_{\mathrm{m}2})$; (1a)-(1c) only for $i = 3$, with state variables (x_3, y_3, z_3). To summarize, this gives a total of 9 state variables, stored in the state vector X.

This model will be used for SDP. Therefore it needs to be discretized in time: $X_{k+1} = f(X_k, v_{\mathrm{ref},k}, \omega_{\mathrm{ref},k})$ where k is time sample, and $f(\cdot)$ is calculated accordingly. Furthermore, it will be augmented once more using the stochastic dynamics of driver intention.

IV. Modeling Driver Intention

Unlike wheelchair motion which obeys physical laws, that are well known and can be modeled in a deterministic way (see section III), it is very difficult to know upfront the decisions $(v_{\mathrm{dem}}, \omega_{\mathrm{dem}})$ of any driver, given a circuit with predefined obstacles. The best one can hope is to derive a statistical model of driver intention:

$$v_{\mathrm{dem},k+1} = w_{1,k} \qquad \omega_{\mathrm{dem},k+1} = w_{2,k}$$

where $w_{1,k}$ and $w_{2,k}$ are random variables.[2] In this section we are interested in the analytic probability distribution functions for two proposed driver models. In future work, they will be used to:

- form a stationary Markov chain.
- generate random driving cycles.

A. The control design philosophy

In control problems with uncertain variables, the probability distribution of the uncertainty is the means to inform about plant dynamics which are likely to occur. Therefore, it is important to highlight the possibility of hitting obstacles so that this situation is taken into account when designing control. Two extreme driver models are the *expert driver*, who will know to keep distance from obstacles and easily find its way through a circuit, and the driver which makes wrong moves and hits the obstacles. The latter is of particular interest to us, as we intend to achieve control capable of correcting worst-case scenarios. Two variants will be analyzed in the following sections.

B. Blind driver model

This driver explores the environment ignoring obstacles, is indifferent to their location and has an equal chance to hit or not to hit the obstacles. Therefore, this driver can be modeled using an uniform distribution: see Fig. 9(a).

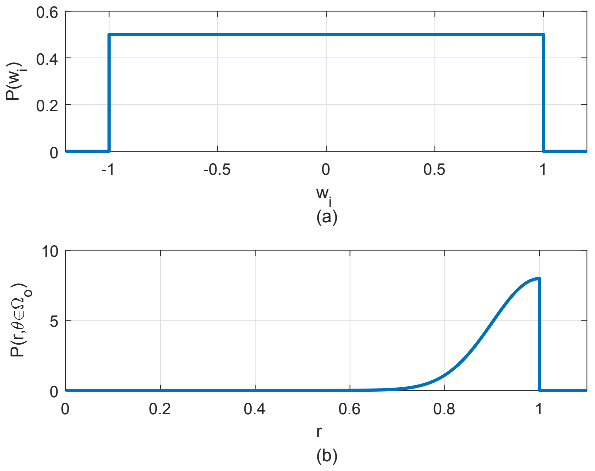


Fig. 9: Standard probability distributions: (a) uniform, $i = 1,2$; (b) Gaussian, $R = 1$. Normalized x-axis.

C. ”Naughty child” driver model

This driver will tend to request maximum velocity and hit the closest obstacle on purpose, situated at a given R radius around the wheelchair. Otherwise, the driver will just move maximum forward velocity. The value of R corresponds to the range of our calibrated ultrasonic sensors: this choice allows us to match the driver's sight with how far the instrumented wheelchair can reliably sense obstacles in the environment.

Largely inspired by the dynamic window approach [7], in Fig. 10 we have overlapped the velocity space $(v_{\mathrm{dem}}, \omega_{\mathrm{dem}})$ and wheelchair's moving frame within the Cartesian space (the origin represents the receding wheelchair position). As the wheelchair advances, obstacles are to be found in any of the 8 regions Ω_i, with $i = 1,..,8$; each region is covered by one ultrasonic sensor with 45° beam spread. Let Ω_{o} be that specific region where the obstacle is closest to the wheelchair, and the angle $\theta \in [0, 360^\circ)$ used to sweep the regions. Then, the probability distribution of the *naughty child* model is:

$$P(r,\theta) = \begin{cases} \sqrt{\dfrac{2}{\pi\sigma^2}}\exp\left\{-\dfrac{(r-R)^2}{2\sigma^2}\right\} & ,\text{if } \theta \in \Omega_{\mathrm{o}} \\ 0 & ,\text{otherwise} \end{cases} \tag{6}$$

where σ is the standard deviation of this normal distribution; $r \in [0, R)$ is the distance from the origin along a ray. This distribution is illustrated in Fig. 9(b). By using the one-to-one change of coordinates:

$$\begin{aligned} v_{\mathrm{dem}} &= r\sin\theta & \theta &= \operatorname{atan2}(v_{\mathrm{dem}}, \omega_{\mathrm{dem}}) \\ \omega_{\mathrm{dem}} &= r\cos\theta & r &= v_{\mathrm{dem}}/\sin\theta \end{aligned}$$

one can express (6) as $P(v_{\mathrm{dem}}, \omega_{\mathrm{dem}})$.

[2] Please note the difference in notation between the Latin alphabet letter w used to represent uncertainty, and the Greek alphabet ω which stands for angular velocity.

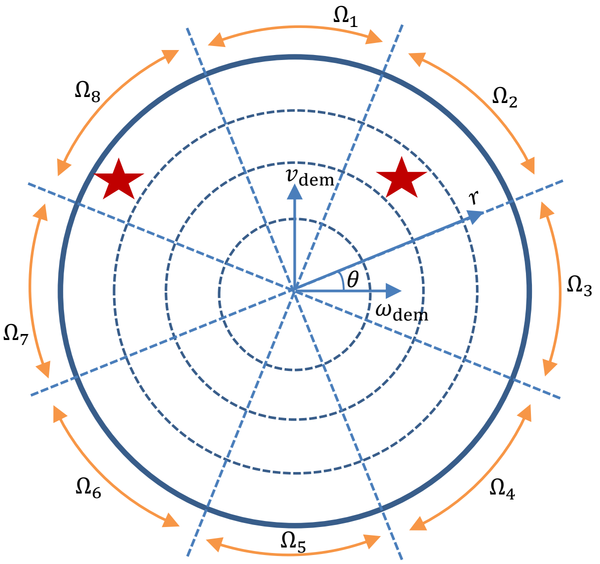


Fig. 10: The velocity space subdivided into regions Ω_i, $i = 1,..,8$. Red stars indicate the presence of obstacles in those directions. The blue solid circle is the domain boundary, in accordance to the physical joystick outputs.

V. Discussion

Contrary to other control design techniques (e.g. deterministic dynamic programming [8]) which necessitate a given deterministic driving cycle, stochastic dynamic programming (SDP) is appealing for wheelchair obstacle avoidance since the *supervisory control* is optimized over a family of random driving cycles (i.e. driver intentions) in an average sense. We expect the control law issued by the *naughty child* model to be quite reactive, repulsive to the nearest obstacle. This happens due to the high probability that the driver is expected to deliberately hit the obstacle. On the other hand, the control law issued by the *blind driver* will be less cautious in terms of obstacle avoidance, since there is an equal probability that the driver might choose to do something else than hitting the obstacle. Surprisingly, we expect an *expert driver* model to be the least appropriate for *safe* control design. For this reason we decided not to model the expert driver here. The probability of hitting obstacles would be quite low, so SDP would not properly take into account this (presumably rare) occurrence.

Now, we propose to look at our obstacle avoidance control problem from another point of view. SDP is optimal with respect to the chosen driver intention model. Consequently, we expect that a real *expert driver* participant might not feel comfortable at all when testing the instrumented wheelchair running a control policy computed using the *naughty child* or *blind driver* model. Therefore, in the future, we intend to compute multiple SDP control policies associated to various driver models, and sort them in ascending order in terms of safety (it is common language in wheelchair field to think of Profile 1,2,etc.) Then, the real participant driver might have to play with them before finding the one that best suits his expectations.

VI. Conclusions

This article covered two design requirements of stochastic dynamic programming for a future smart wheelchair (assistive technologies). First, a dynamic input-output model of wheelchair motion was elaborated using Euler-Lagrange equations. Inputs as well as outputs are defined in velocity space. Second, two models of driver intention were presented as means to cope with a safe ride requirement and thus avoid obstacles.

References

[1] T. Carlson and Y. Demiris, “Collaborative control for a robotic wheelchair: evaluation of performance, attention, and workload,” *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 876–888, 2012.

[2] Q. Li, W. Chen, and J. Wang, “Dynamic shared control for human-wheelchair cooperation,” in *2011 IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, May 2011, pp. 4278–4283.

[3] A. Goil, M. Derry, and B. D. Argall, “Dynamic shared control for human-wheelchair cooperation,” in *2013 IEEE 13th Int. Conf. on Rehabilitation Robotics (ICORR)*, Seattle, USA, June 2013, pp. 1–6.

[4] 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.

[5] P. Trautman, “Assistive planning in complex, dynamic environments: a probabilistic approach,” in *2015 IEEE Int. Conf. on Systems, Man, and Cybernetics*, Hong Kong, October 2015, pp. 3072–3078.

[6] C. Ezeh, P. Trautman, L. Devigne, V. Bureau, M. Babel, and T. Carlson, “Probabilistic vs linear blending approaches to shared control for wheelchair driving,” in *2017 Int. Conf. on Rehabilitation Robotics (ICORR)*, Toronto, Canada, July 2017, pp. 835–840.

[7] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

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

[9] 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.

[10] I. Kolmanovsky, I. Siverguina, and B. Lygoe, “Optimization of powertrain operating policy for feasibility assessment and calibration: Stochastic dynamic programming approach,” in *American Control Conf.*, Anchorage, AK, May 2002, pp. 1425–1430.

[11] A. M. Bloch, *Nonholonomic Mechanics and Control*. Springer, 2003.

[12] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Hoboken, NJ: John Wiley & Sons, 2005.

[13] Á. Odry, I. Harmati, Z. Király, and P. Odry, “Design, realization and modeling of a two-wheeled mobile pendulum system,” in *Proc. of the 14th Int. Conf. on Instrumentation, Measurement, Circuits and Systems*, Salerno, Italy, June 2015, pp. 75–79.

[14] C. S. Teodorescu, S. Vandenplas, B. Depraetere, J. Anthonis, A. Steinhauser, and J. Swevers, “A fast pick-and-place prototype robot: design and control,” in *IEEE Multi-Conference on Systems and Control*, Buenos Aires, Argentina, September 2016.