

Model-based sensor fusion and filtering for localization of a semi-autonomous robotic vehicle*

Catalin Stefan Teodorescu¹, Irving Caplan¹, Harry Eberle¹ and Tom Carlson¹

Abstract—This paper refines a physically-inspired model governing the dynamic motion of a vehicle. We present a method used to perform experimental parameter calibration, and then use this model to build an observer (an extended Kalman filter). Experimental results with a robotic vehicle fitted with a prototype kit focus on recovering the truthful real-world information in the context of systematic errors (a faulty wheel encoder sensor), randomly occurring errors (a faulty ultrasonic sensor) and simplifying model assumptions (e.g. usage of two identical motors). We show that our model-based approach is able to perform reasonably well even under these extreme circumstances.

I. INTRODUCTION

In our previous works [1], [2], we have developed a model-based control algorithm for safely driving a semi-autonomous robotic vehicle (a commercial electric powered wheelchair converted into a research platform). The proposed control design was based on stochastic dynamic programming. As expected with this type of approach, the quality of the control law is highly dependent on how representative the *plant model* is with respect to the *real world* phenomena. The more reliable the plant model, the better the control law performs in the real world. The same mechanical dynamics model of a nonholonomic unicycle-type robot moving on a horizontal or inclined plane can be used not only for control design, but also for defining an observer as we will show in this paper.

The contributions of this article are as follows. In continuation of our previous work [1], here we further elaborate upon the *plant model* by performing experimental parameter identification. Up until now, we had only been working with *nominal* (theoretical) parameters and simulations, whereas in this article we work with real world experimental data. This improves the reliability of the model, which will then be used for state estimation (sensor data filtering). Our contribution is in the internal model used for the chosen model-based observer. We chose to use an *extended Kalman filter* (EKF) [3], [4], although many other options are open, e.g. the simplified kinematic model [5]; *particle filter* [5], [6, §6.7],

[3]; Bayesian filter [3]; the *explicit complementary filter* [6, §3.4].

In our work, the filtered sensor data is then used as input to the *assist-as-needed* (terminology according to [7]) or *shared control* (terminology according to [8]) algorithm. More generally, it is a semi-autonomous control solution well adapted to the needs of the impaired people. This is used for designing mobility assistance systems for navigation in environments with static obstacles, which is still an active field of research [9].

This paper is organized as follows. Section II is theoretical, presenting the mathematics of an internal model. Section III is dedicated to the instrumentation of the vehicle. Results from both previous sections are used in the implementation of an EKF in section IV, and the paper ends with Conclusions in section V.

II. MODELING

A. Velocities: Consistent with notations in the literature [10], we use v for linear velocity and ω for angular velocity. At any given instant t , the vehicle is either: (i) standing still ($v = 0, \omega = 0$), or (ii) moving in a straight line ($v \neq 0, \omega = 0$), or (iii) turning in a circle with radius $R_M = v/\omega$ (any $v, \omega \neq 0$) around an imaginary axis k perpendicular to the ground. In Fig. 1, the depicted instantaneous angular velocity vector is $k\omega$ and the linear velocity vector is jv , where the free unit vectors $k = (0 \ 0 \ 1)$ and $j = (0 \ 1 \ 0)$ are expressed with respect to the inertial frame $o_0x_0y_0z_0$; $\omega = \dot{\varphi}$, with φ the angular displacement. In this work, we use the base frame $o_Mx_My_Mz_M$ rigidly attached to the moving vehicle. The base frame's origin can be expressed in (with respect to) the inertial frame as

$$o_M^0 = v \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix} \quad (1)$$

where the first and second components of o_M^0 indicate time-dependency along the x_0 -axis, and y_0 -axis, respectively (see Fig. 1). The superscript “0” indicates that coordinates are expressed in the inertial frame.

B. Plant models: The references indicated in the Introduction, show that rather simple plant models are being used for model-based observers. As we aim to work with faulty sensors, relying on more accurate model dynamics is essential to compensate for these errors. In our previous work [1] we have derived the dynamic equations governing the motion of the vehicle on an inclined plane. This model made use of *nominal parameters* (given in [1]) and we showed in simulation that the overall behavior was consistent with

*This work is carried out as part of the INTERREG VA FMA ADAPT project “Assistive Devices for empowering disAbled People through robotic Technologies” <http://adapt-project.com/index.php>. The Interreg FCE Programme is a European Territorial Cooperation programme that aims to fund high quality cooperation projects in the Channel border region between France and England. The Programme is funded by the European Regional Development Fund (ERDF)

¹Catalin Stefan Teodorescu, Irving Caplan, Harry Eberle and Tom Carlson are with Aspire Create, University College London, Royal National Orthopaedic Hospital, HA7 4LP, UK {s.teodorescu, i.caplan, h.eberle, t.carlson}@ucl.ac.uk

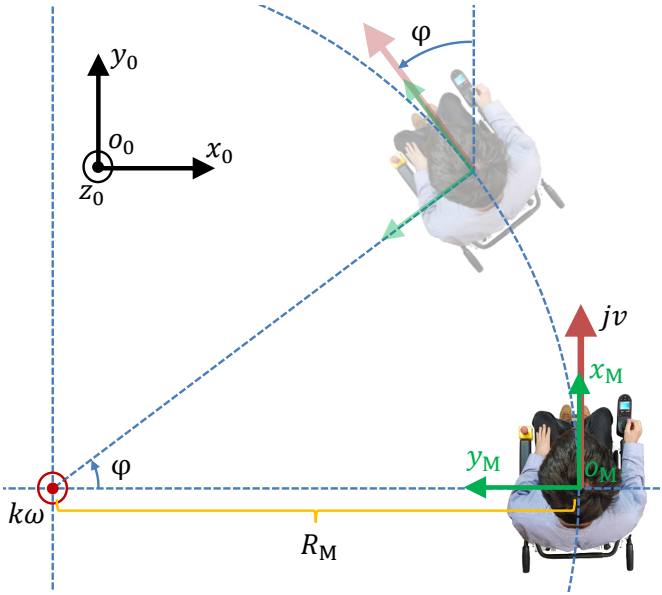


Fig. 1: View from above: circular trajectory of the moving vehicle when setting the linear velocity $v > 0$ and the angular velocity $\omega > 0$

our expectations. In the continuation of that work, here we rewrite the system of equations in a way that it: (i) allows for experimental *linear* parameter identification; (ii) can be used for model-based state estimation. Two distinct but equivalent system forms will be used for that purpose, both explained in the following two sections.

C. The (Y, θ) form: It is a well-known property that mechanical systems can be written in such a way to express linear dependency [10] on some physical lumped parameters θ . We carried out all calculations which allowed us to rewrite the overall dynamics in a compact form:

$$Y(v, \dot{v}, \omega, \dot{\omega})\theta = \mathbf{l}_m \quad (2)$$

where $\mathbf{l}_m = (l_{m1}, l_{m2})$ is the vector of motors' electric current and, for our vehicle, the 2×5 elements of the regressor matrix $Y(v, \dot{v}, \omega, \dot{\omega})$ are:

$$\begin{aligned} y_{i1} &= \frac{r}{r_G} (m_w(\dot{v} + (-1)^{i+1} l_w \dot{\omega}/2) + m_b \dot{v}/2) \\ y_{i2} &= \frac{r}{r_G} (\dot{v} + (-1)^{i+1} l_w \dot{\omega}/2); \quad y_{i3} = \frac{1}{r r_G} (\dot{v} + (-1)^{i+1} l_w \dot{\omega}/2) \\ y_{i4} &= (-1)^{i+1} \frac{r}{r_G} \frac{\dot{\omega}}{l_w}; \quad y_{i5} = \frac{r_G}{r} (v + (-1)^{i+1} l_w \omega/2) \end{aligned}$$

with row index $i \in \{1, 2\}$; the lumped parameters vector

$$\theta = \left(\frac{1}{\eta_G k_m}, \frac{J_m}{\eta_G k_m}, \frac{I_{w,zz}}{\eta_G k_m}, \frac{I_{zz}}{\eta_G k_m}, \frac{B_m}{k_m} \right)' \quad (3)$$

We refer the reader to [1], for the meaning of all these physical parameters. Special attention is paid to the *motor torque constant* k_m . In our model, we made the assumption that both motors are identical, which allowed us to use the same k_m for both motors. In reality, motors are slightly different and this motivated us to experimentally identify

separately each *motor torque constant* k_{mi} with $i = 1, 2$. One method was to use the input-output power balance: the output mechanical power (torque τ_{mi} multiplied by shaft's angular velocity ω_{mi}) equals the product between the input electrical power (motor voltage U multiplied by motor electric current i_{mi}) and the motor efficiency η_i . Using the first-order approximation $\tau_{mi} = k_{mi} i_{mi}$, one can extract the ratio $k_{mi}/\eta_i = U/\omega_{mi}$. Since we cannot measure directly U on the vehicle (a PWM signal at high frequency 20 kHz), we will use instead the available low-pass resistor-capacitor filtered signal V_{RC} and limit our analysis to quasi-steady-state behaviour, when V_{RC} is an average value of U . Thus,

$$k_{mi}/\eta_i \approx V_{RC}/\omega_{mi} \quad (4)$$

Results are shown, later on, in Fig. 6d. A second method to calculating k_m is to use the non-invasive *in situ* approach [11]. Here, the key to successful identification of k_m is that the robot should move in a way that is persistently exciting [6, §9.6].

Using experimental data $(v, \omega, \mathbf{l}_m)$ and the relation (2), the computed *Least Squares* [10] result:

$$\hat{\theta} = (\mathbf{Y}'\mathbf{Y})^{-1}\mathbf{Y}'\mathbf{l}_m \quad (5)$$

will be biased to the actual, physical θ from (3). This is the price to pay for applying this straightforward Least Squares method, where we lose the physical insight of θ (a white box model) and, instead, have to work afterwards with a mathematical object $\hat{\theta}$ (a grey box model). In (5), \mathbf{Y} and \mathbf{l}_m are formed by stacking, on top of each other, associated experimental data measurements (samples) taken at various time instants.

In order to effectively compute $\hat{\theta}$ numerically in (5), we needed to: (i) apply zero-phase digital filtering on the acceleration signal; (ii) arbitrarily increase the numerical precision¹ beyond *double* such as to avoid singularity of the matrix $\mathbf{Y}'\mathbf{Y}$.

D. The ODE form: We have converted the system dynamics into the following ODE form:

$$\begin{pmatrix} \dot{v} \\ \dot{\omega} \end{pmatrix} = \begin{pmatrix} a_3 & 0 \\ 0 & a_4 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} + \begin{pmatrix} a_1 & a_1 \\ a_2 & -a_2 \end{pmatrix} \mathbf{l}_m \quad (6)$$

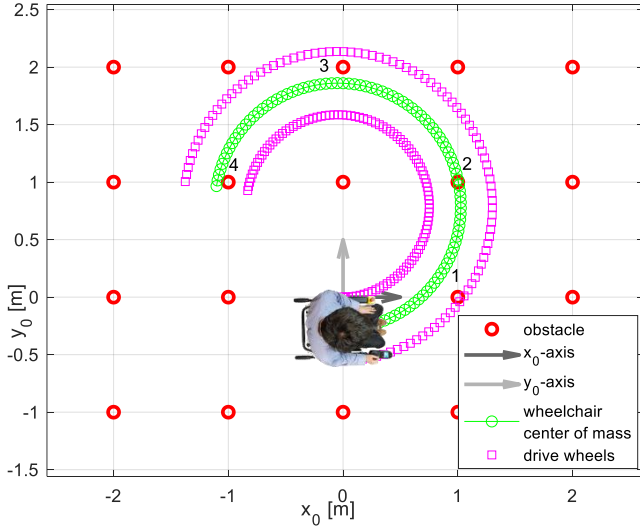
where positive parameters a_1 and a_2 , as well as negative parameters a_3 and a_4 explicitly depend on θ from (3): see their explicit formulas in Appendix VI. Actually, one can use the experimentally identified $\hat{\theta}$ from (5) when computing a_1 to a_4 . In (6), it is interesting to notice that dynamics of v and ω are uncoupled from one another.

E. A static obstacle: A straightforward calculation of the equation of motion for a static obstacle as perceived by a moving frame is:

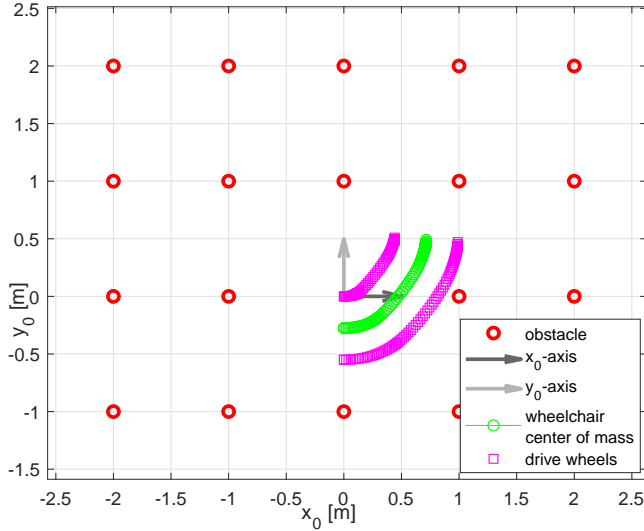
$$\dot{p}_o^M = -S(k\omega)p_o^M - \begin{pmatrix} 1 \\ 0 \end{pmatrix} v, \quad \text{with } S(k\omega) = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix} \quad (7)$$

The vector $k\omega$ was depicted in Fig. 1. This formula allows for a physical interpretation where the first term in (7),

¹Implemented in Matlab using Symbolic Math Toolbox's functions `digits()` and `vpa()`



(a) Supervisory Control is disabled during 10 seconds: 4 obstacles, numbered in chronological order, were hit and run through



(b) Supervisory Control is enabled during 10 seconds: no obstacle was hit

Fig. 2: View from above: Vehicle's trajectory with driver demanding constant velocities (i.e. fixed joystick position), in an environment with obstacles. Initial, starting position is where the driver is sitting (same for both figures).

namely $S(k\omega)p_o^M = k\omega \times p_o^M$ [10] expresses the tangential linear velocity component caused by the rotating vehicle (this is not to be confused with vector jv). The minus sign in front of the first term indicates that if the vehicle rotates in one direction (e.g. clockwise), then the static vehicle is perceived in the base frame as moving in the opposite direction (e.g. anti-clockwise). The second term in (7) expresses the contribution of the linear velocity jv from Fig. 1 which, according to its definition [1] always acts along the x_M -axis. The minus sign in front of the second term in (7) follows the physical intuition: a moving vehicle advancing towards a static obstacle is perceived from the base frame as a dynamic (moving) obstacle coming towards a static vehicle.

Alternatively, an equivalent form to system (7) is [10, §14.7.3]:

$$d_o^M = -v \cos \theta_o^M \quad \text{and} \quad \dot{\theta}_o^M = -\omega + v \frac{\sin \theta_o^M}{d_o^M} \quad (8)$$

where d_o^M is the distance to the obstacle, and θ_o^M is the angle from x_M -axis to the obstacle (see Fig. 3).

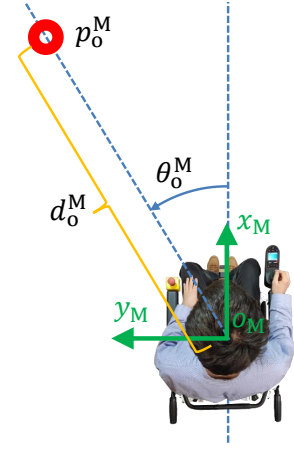


Fig. 3: View from above: Static obstacle in red situated at location p_o^M (expressed in base frame coordinates $o_M x_M y_M z_M$), distance d_o^M away from the vehicle, and angle θ_o^M

To verify that (7) and (8) are equivalent, one can use the change of variables:

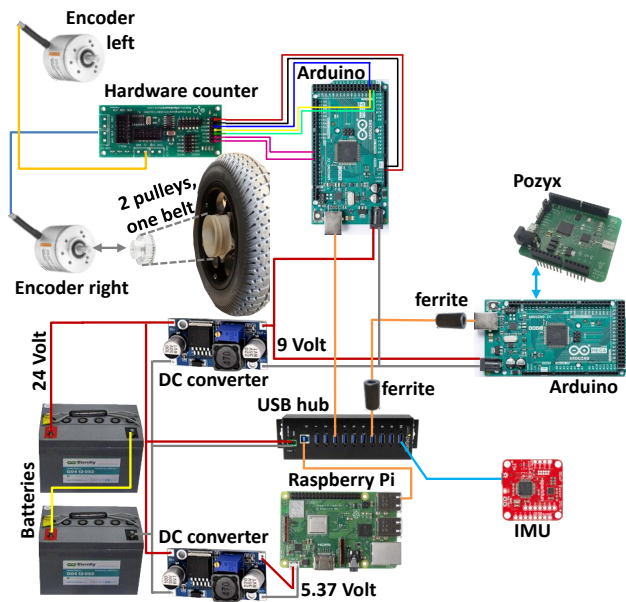
$$p_o^M = \begin{pmatrix} \cos \theta_o^M \\ \sin \theta_o^M \end{pmatrix} d_o^M \quad \text{and} \quad \begin{aligned} d_o^M &= \|p_o^M\|_2 \\ \theta_o^M &= \arctan(p_o^M(2)/p_o^M(1)) + n\pi \end{aligned}$$

where $n \in \mathbb{Z}$.

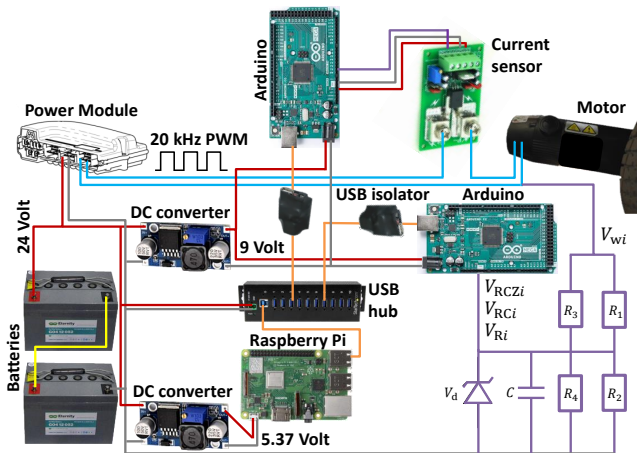
F. Control: In this section, we created in simulation, an artificial environment with static obstacles situated about 1 m apart from each other. An assist-as-needed *Supervisory Control* [2] is used in order to illustrate the concept of safe navigation.

Fig 2a shows the position of the wheelchair's drive wheels and center of mass as it makes a turn. The *Supervisory Control* is disabled and the driver keeps the joystick in a fixed position corresponding to $v = 0.52$ [m/s] and $\omega = 0.27$ [rad/s], for 10 seconds: this will cause the vehicle to turn in circle, towards the left. Notice a few obstacles were hit and run through (they are present in the space between the two drive wheels). Next, in Fig 2b the same conditions as earlier mentioned are kept, except that the *Supervisory Control* is now enabled. Notice that no obstacles were hit under this condition. The total traveled distance is much less compared to the previous case, since the vehicle reduces its velocities gradually, according to the distance to obstacles: the closer the obstacle, the lower the velocities. Once the obstacle gets out of the way (e.g. the driver might acknowledge the danger and steer the vehicle in another direction), the allowed velocities would increase to standard operation.

Now that we have ended presenting the theoretical results, we move on to describing the instrumented vehicle which



(a) Dead reckoning (wheel encoders, IMU) and absolute position (Pozyx)



(b) Motors' electric power monitoring (motor voltage and electric current)

Fig. 4: Wiring electrical components

will then allow us to measure some of the quantities introduced in this section. Others will be reconstructed using state estimation, as described later on, in section IV.

III. THE ELECTRICAL CIRCUIT. SENSORS

In order to provide environment awareness capabilities to the vehicle, we needed to make a few adjustments. A schematic electrical circuit diagram of the additional custom-made components is given in Figs. 4a and 4b.

A. Dead reckoning using encoders: We mounted industrial wheel encoders on our test platform that are able to provide raw odometry measurements. The turning angle (i.e. angular displacement) of the right drive wheel ψ_1 , measured in radians, is calculated from the right encoder counts (ticks) ς_1 as:

$$\psi_1 = \varsigma_1 \frac{2\pi N_{ep}}{N_{ec} N_{mp}} \quad (9)$$

where each parameter is explained in Table I. A similar formula is used to calculate the angular displacement of the left drive wheel ψ_2 .

TABLE I: Parameters for odometry

Symbol	Meaning	Value
N_{ep}	number of teeth of a small aluminium pulley, physically connected to the encoder's shaft, see Fig. 4a	30
N_{mp}	number of teeth of a larger 3D-printed pulley, physically attached to the main drive wheel, see Fig. 4a	110
N_{ec}	number of counts issued by the encoder for one full 360° rotation of its shaft	2000

By combining together ψ_1 and ψ_2 , we are able to compute φ as depicted in Fig. 1. Moreover, using numeric differentiation, it is possible to estimate (v, ω) from (6), which can also be visualized in Fig. 1. For exact formulas, we guide the reader to our previous work [1]. The absolute position of the wheelchair can now be expressed in the inertial frame using (1).

Validation: We performed indoor experiments looking at two situations: (i) vehicle advancing forward along a linear spatial trajectory for 10 m, and (ii) vehicle rotating 180° . In the first case, total traveled linear distance errors were around 4%, whereas the second case showed rotation angle errors on φ around 10%. This significant decrease in performance encouraged us to seek alternatives to improve the rotation estimation.

B. Dead reckoning using an IMU's gyroscope: We used the gyroscope's angular velocity measurement ω around the k -axis illustrated in Fig. 1. Experimental measurements showed better results compared to wheel encoders, as will be shown later on, in Fig. 6b.

C. Absolute position measure using Pozyx: Pozyx Labs developed a commercial indoor position tracking solution [12] using an ultra-wide band chip designed and manufactured by Decawave. Specifications indicate an accuracy within 10 cm confirmed experimentally by [5], who evaluated it for sporting activities. Our kit consisted of 4 anchors (fixed antennas) and one mobile tag (mobile antenna), which can be used as a visual guide or a rough representation of the *ground truth*. In particular, the *mobile tag* was fixed on a 20 cm wooden extension of the metal frame, high up so that: (i) line-of-sight is kept between the anchors and the tag, similar to the wheelchair used in [13]; (ii) antenna's efficiency is not mitigated by the nearby presence of metal parts [12].

D. Monitoring motors' electric power: For online (real-time) monitoring of electric power, we needed to measure both electric current as well as voltage. First, for measuring t_m from (6), we have mounted two identical off-the-shelf electric current sensors: see Fig. 4b and details in Appendix VII, Table IV. Second, the voltage is measured using a custom-made voltage sensor, detailed below.

Motor voltage measurement: The circuit in Fig. 4b consists of a voltage divider (resistors R_1 to R_4), one capacitor C for low-pass filtering the voltage and one optional Zener diode Z_d with nominal Zener voltage (i.e. knee) V_d . The values for all these parameters can be found in

Appendix VII, Table III. They were dimensioned (chosen) so that: (i) the overall RC low-pass filter has a cutoff frequency of 100 Hz, signal that can be recorded in real-time using an Arduino²; (ii) the maximum power dissipation on each component is well within specifications. The Zener diode is used for protecting the Arduino (which can handle an analog input of up to 5.5 Volt according to specifications) in case an overvoltage might occur (e.g. due to back EMF when the vehicle runs down a slope at high speed: the Zener diode starts operating, acting as a saturation on the measured voltage $V_{RCZi}(t) = \text{sat}(V_{RCi}(t))$); here we used the subscript to indicate the individual components used to build the electrical circuit). The Zener diode's main drawback is related to the *knee* effects (the significant increase in electric current close to the knee). Thus, the electric current starts to flow ahead of reaching V_d , making the theoretical voltage-divider formula:

$$V_{Ri} = R_{2\parallel 4} V_{wi} / (R_{1\parallel 3} + R_{2\parallel 4}) \quad (10)$$

erroneous; V_{wi} is the voltage on one of the two wires forming the *motor lead* connecting the power module to the motor (with $i = 1, 2$ standing for each wire index from here onwards); the selected resistance values are given in Appendix VII. The effect of adding the Zener diode alongside the voltage divider circuit is assessed quantitatively in Table II. There we run five steady-state measurements using a Fluke 175 multimeter with 3 digits precision after the decimal point. The second column is the theoretical voltage-divider output (10). The third column shows the experimentally measured voltage $V_{RCi, \text{exp}}$ across the circuit made up of the resistance voltage divider plus the capacitor. The fourth column indicates the voltage $V_{RCZ_{di}, \text{exp}}$ across the circuit formed by the resistance voltage divider plus capacitor plus Zener diode. Note the effect of the capacitor C is (theoretically) negligible for steady-state measurements. Finally, the last column of Table II shows the relative experimental error when adding the Zener diode:

$$\varepsilon_i = 100(V_{RCi, \text{exp}} - V_{RCZ_{di}, \text{exp}}) / V_{RCZ_{di}, \text{exp}}$$

(measured in percentage) which increases as the input voltage V_{wi} increases.

TABLE II: Steady-state performance of the motor voltage circuit

V_{wi} [V]	V_{Ri} [V]	$V_{RCi, \text{exp}}$ [V]	$V_{RCZ_{di}, \text{exp}}$ [V]	ε_i [%]
29.2	5.119	5.101	4.684	8.2
25.6	4.488	4.473	4.331	3.2
22.0	3.857	3.840	3.806	0.9
18.4	3.225	3.210	3.204	0.2
14.8	2.594	2.579	2.578	0.0004

Dynamic model: In response to an increase of PWM duty cycle and/or the pulses' amplitude, a motor would start to spin and consequently the vehicle advances. The PWM

²Note the wheelchair's Power Module issues a 20 kHz power signal to drive the motors, which is well beyond the recording capabilities using analog input of an Arduino (in the order of 1 kHz).

pulses are low-pass filtered by the electric current motor sensor, and the Arduino would sense (measure) an average voltage $V_{RCi}(t)$ according to:

$$R_{\parallel} C \dot{V}_{RCi}(t) = -V_{RCi}(t) + R_{2\parallel 4} V_{wi}(t) / (R_{1\parallel 3} + R_{2\parallel 4}) \quad (11)$$

Note that in steady-state $V_{RCi}(t) = V_{Ri}$ from (10). Here we assumed normal working operation where overvoltage protection is not necessary (the voltage across the Zener diode is well below its *knee* voltage). All parameters from (11) are detailed in Appendix VII.

By combining together voltages measured on both wires ($i = 1, 2$ in (11)) that make up the *motor lead* associated to one motor, we get:

$$R_{\parallel} C \dot{V}_{RC}(t) = -V_{RC}(t) + R_{2\parallel 4} U(t) / (R_{1\parallel 3} + R_{2\parallel 4})$$

with $U = V_{w1} - V_{w2}$ being the difference in potential across the two wires; $V_{RC} = V_{RC1} - V_{RC2}$ is measured online using the Arduinos.

Finally, the motor circuit dynamics are given by:

$$L_m \, dI_{mi} / dt = -R_m I_{mi} - k_b \omega_{mi} + U$$

where L_m and R_m are the motor inductance and resistance, respectively; the back EMF constant k_b has the same numeric value as k_m provided standard units [10].

E. Ground loops issue: Although individual sensors' wiring did not pose any problem, making them all work together (integration) proved to be a challenging task. The *USB isolators* in Fig 4 were the key to effectively break the loop formed by connecting the battery's negative terminal to the non-isolated DC converter, then to the Arduino's ground, next via the USB to the RPi's ground, and finally via the non-isolated DC converter back to the battery's negative terminal.

The next section is dedicated to combining benefits of two sensors described above (the encoders and the gyroscope) in order to improve the quality of measurements. For this we shall make use of the internal model from section II.

IV. AN EXPERIMENT WITH AN OBSERVER

We begin this section by presenting the chosen Extended Kalman Filter (EKF) implementation and the reasons behind it. Next, rather than focusing on the situations where the EKF performs well (observe asymptotic convergence towards the *real plant*, as expected in theory [3], [4]), we create a more challenging situation where inputs to EKF are faulty sensor *real world* data.

Among the many successful implementations of EKF readily available in the literature [14], we opted for the continuous-time extended Kalman filter with asynchronous discrete-time measurements [4, §5.3.3]. The reasons were related to its future implementation in Robot Operating System (ROS):

- the ability to work with input asynchronous measurements. ROS is *not* a real-time system and packets coming from sensors are published on topics as they arrive, sometimes dropped out or lost. This should not break the EKF from functioning. Another effective implementation is [15].

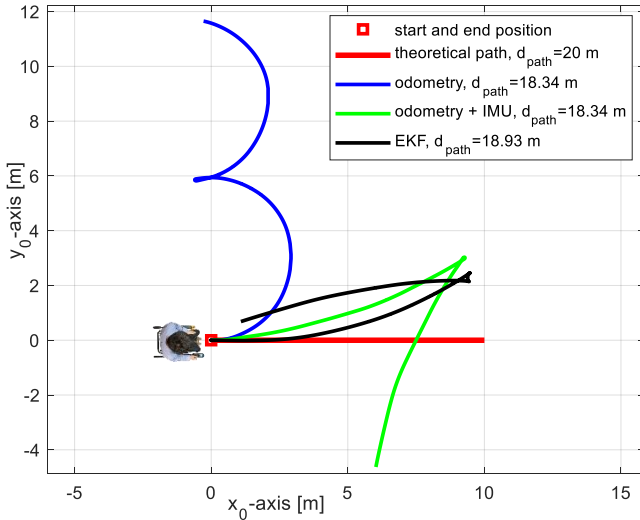


Fig. 5: View from above: trajectories in Cartesian space of the moving vehicle according to various sensors

- its potential to upscale the publishing frequency on output ROS topics. This relies on the prediction model during two successive measurements, running at higher frequency than the observations.

The *internal model* consists of 4 state variables: the velocities (v, ω) with their dynamics (6); the (x_M, y_M) -coordinates of one static obstacle with its motion p_o^M in (7). Equivalently, the *internal model* may be used instead of (x_M, y_M) , the dynamics of the obstacle expressed in terms of distance and angle (d_o^M, θ_o^M) according to (8). An input to this state estimator is the readily available motors' electric current vector \mathbf{i}_m : see, e.g. Fig. 6c. The observations $z = (v, \omega, d_o^M)$ come, respectively, from the odometry, IMU's gyroscope and ultrasonic sensors. For initializing the EKF, we used the datasheet containing the ultrasonic sensor's characteristics. This allowed us to define the initial condition for the estimated states (we've used zero for the velocities, corresponding to a standstill situation, and the most probable location for the obstacle), and the initial value of the *reconstruction error* covariance matrix. For tuning this EKF we've used diagonal constant *noise* covariance matrices.

A. The Experiment: We defined a scenario consisting of a succession of three tasks: advance forward 10 m, then turn 180° and finally go back to the initial position. These 3 tasks are clearly marked in Fig. 6. The path followed by the advancing vehicle is illustrated by the straight red line in Fig. 5, with the start and end point indicated using a red square.

We decided to analyze in this paper a rather extreme situation by using poorly calibrated odometry. Specifically, we artificially modified the wheel encoder pulleys ratio N_{ep}/N_{mp} from (9). This created a systematic error where the location of the vehicle is interpreted as deviating more towards the left compared to reality: this is visible on the blue curve in Fig. 5 and Fig. 6b. Moreover, the linear velocity value calculated by the odometry has tendency to be lower

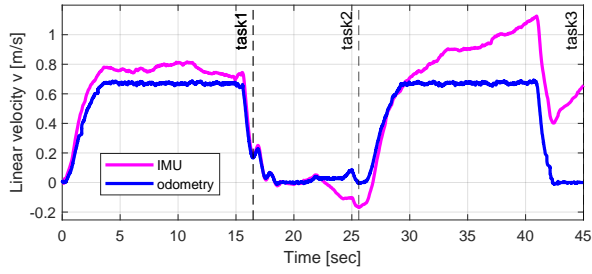
than reality: compare the two curves in Fig. 6a during the first half of the recording (later on, the IMU data seems to diverge).

B. Traveled trajectory: Two metrics were used for assessing the behavior of various sensor fusion combinations: (i) the visualization, in Cartesian space, of the total traveled trajectory, and (ii) the total distance along the traveled path d_{path} . In Fig. 5 the green curve corresponds to raw experimental data measurements coming from two separate sensors: we used the linear velocity v from the odometry and the angular velocity ω from the IMU's gyroscope around the vertical axis (theoretically perpendicular to the ground, but in reality close to it) of the IMU. This shows significant improvement compared to the blue curve corresponding to odometry-only. Finally, the black EKF curve outperforms the others: the end (terminal) position is quite close to the begin (start). An interesting exercise is to try to understand why the EKF is not able to compensate for the erroneous systematic deviation towards the left. A possible reason is that the EKF tries to balance information coming from the sensors with that of the *internal model*. One assumption used to derive the equations of the internal model is that the vehicle uses two identical motors (their characteristics should be identical), which is not true in reality. In particular, while the vehicle advances in straight line, we see in Fig. 6c that the right motor's electric current is higher than the left motor's: this might be due to increased friction in the gearbox, whereas the *internal model* will interpret this as turning towards the left. Consequently, the EKF does not have any means to correct for the systematic deviation towards the left.

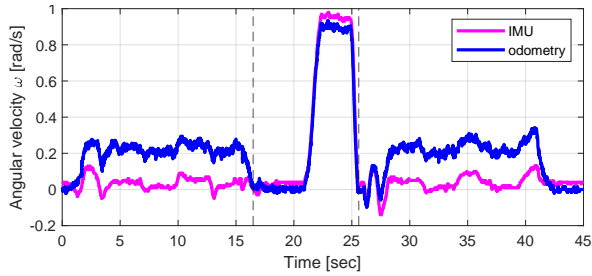
We shall end this section by mentioning a limitation of our study: the lack of a *ground truth* in Fig. 5, which we hope to overcome in future, by making use of the Pozyx system [5] indicated in Fig. 4a.

C. Reconstructed obstacle location: In spite of the systematic error presented above, it would be interesting to know if this EKF is capable of reconstructing the exact location of the static obstacle. Recall that we measure only the distance to the obstacle d_o^M and would like to estimate its (x_M, y_M) coordinates, or equivalently, reconstruct the angle θ_o^M to the obstacle. In other words, this defines a *virtual sensor* for (x_M, y_M) , or θ_o^M , respectively. Simulations show that it is quite challenging to achieve this, especially for our scenario where we don't combine simultaneous linear with angular motion. Consequently, these two state variables are not sufficiently simultaneously excited. To overcome this limitation, we use the standard deviation of the reconstruction error to assess the confidence that the EKF has on its estimates: they form the semi-major and semi-minor axes of the cyan ellipse in Fig. 7. Notice this ellipse is much wider along the y_M -axis compared to the x_M -axis, expressing the fact that the estimator has less confidence in the first estimate. For the particular time instant when the screenshot was taken in Fig. 7, the estimated obstacle location (in green) is significantly far away from the real obstacle (in red).

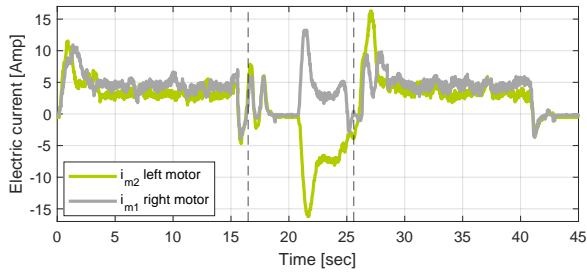
The EKF computes at each time instant the second moments $\sigma^2 = E\{\varepsilon^2\}$ of the reconstructed error $\varepsilon = d_o^M - \hat{d}_o^M$.



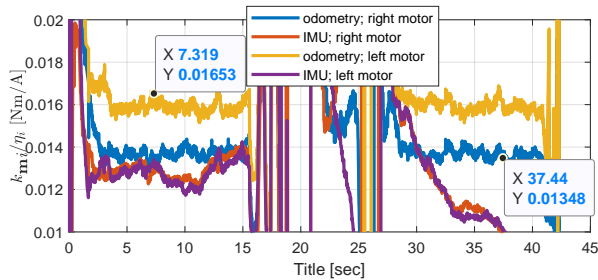
(a) Comparison between two sensors: curves should have coincided



(b) Comparison between two sensors: curves should have coincided



(c) Motors' sensors



(d) Parameter identification of the ratio between motor constants k_{mi} and motor efficiencies η_i using (4), during the quasi-steady-state motion of the vehicle ($i = 1$ corresponds to the right motor; $i = 2$ is the left motor). The legend indicates the sensor used to estimate the motor angular velocity ω_{mi} from (4), as well as which motor.

Fig. 6: Experimental data recorded with the instrumented vehicle running (executing) the 3-tasks scenario

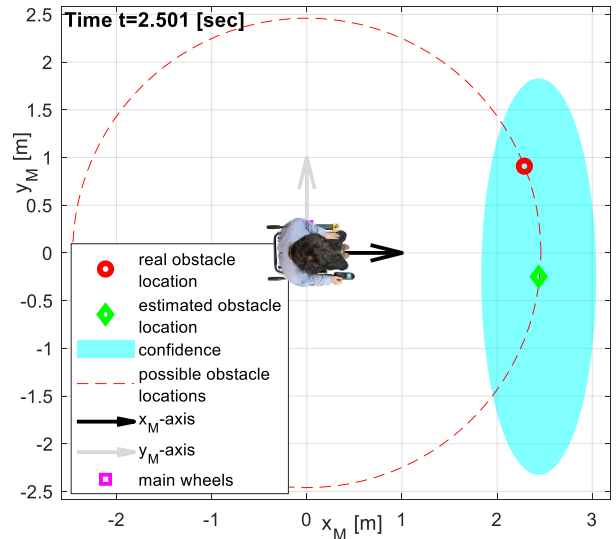


Fig. 7: View from above: Vehicle aware of a static obstacle: its location was estimated with low confidence (we deliberately created a systematic error (a fault) on the left drive wheel encoder's signal)

In Fig. 8, we draw $\hat{d}_o^M \pm \sigma$, thus obtaining an envelope around the estimated signal. Note that from time to time, the ultrasonic sensor fails to measure accurately the distance d_o^M , situation where it outputs zero. As expected, the estimation \hat{d}_o^M diverges locally, then recovers as soon as the measured signal d_o^M improves. Translated back into our assist-as-needed control navigation problem, the vehicle reacts with moderation to the presumed nearby presence of obstacles and the driver is still in control. This result is quite positive, compared to the more penalizing situation of using raw data, which would make the vehicle stop ($d_o^M = 0$ means an obstacle was just hit).

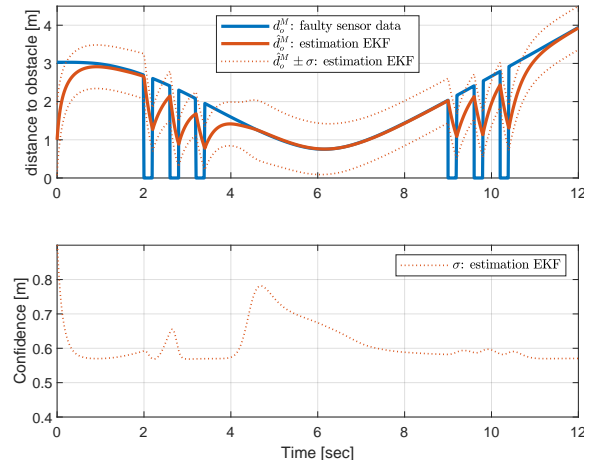


Fig. 8: Reconstructed distance to obstacle (red line) and associated confidence (dotted red line) using ultrasonic sensor data (blue) that fails arbitrarily (3 times between seconds 2 and 4, and again 3 times between seconds 9 and 11)

TABLE IV: Selected electrical and mechanical components on the sensors' schematic diagram in Fig. 4

Component	Description
DC Converter	Unisolated DC-DC Step Down Converter LM2596, max 3A
USB isolator	Olimex USB Full Speed USB 2.0 Compliant Port Isolator (Farnell Order Code: 1795095)
USB hub	Startech 10-Port Industrial USB 3.0 Hub with ESD & 350W Surge Protection
Current sensor	CZH-LABS D-1085 50A electric current sensor (based on ACS758)
Arduino	Arduino Mega 2560
Raspberry Pi	Raspberry Pi 3B+
Hardware counter	2x 32 bit quadrature encoder counter [17]
Encoder left, right	Kubler Incremental Encoder 500 ppr 6 x 12.5 mm Shaft 5 V dc (RS part: 918-9082)
UWB Tags	Ultra-wide band <i>Developer tag</i> [12]
2 pulleys, one belt	Contitech Synchroflex, Timing Belt, 260 Teeth, 650mm Length X 6mm Width (RS part: 474-5218); Aluminium Timing Belt Pulley, 6mm Belt Width x 2.5mm Pitch, 30 Tooth (RS part: 286-4581)
IMU	SparkFun 9DoF Razor IMU M0 – SEN-14001

V. CONCLUSIONS

A model-based holistic approach was pursued in this paper, showing both advantages but also limitations. Calibrating a physically-inspired mechanical model against experimental data, then integrating it into an observer improved the quality of signal estimations. Although errors were not completely eliminated, using these estimations as input to our assist-needed algorithm provided a smooth driving experience and a more effective balance between the driver's intention and the corrective action. A future application where we aim to test the same calibrated model presented in this paper, is as *indirect* measurement for motors' torque, thus integrating it into a cheap *software* torque sensor.

APPENDICES

VI. WHEELCHAIR DYNAMICS

This section provides details concerning the parameters used in (6):

$$\begin{aligned}
 a_1 &= 0.5 \left(\frac{r}{r_G} \theta_1 \left(m_w + \frac{m_b}{2} \right) + \frac{r_G}{r} \theta_2 + \frac{\theta_3}{r r_G} \right)^{-1} \\
 a_2 &= 0.5 \left(\frac{r}{r_G} \left(\theta_1 m_w \frac{l_w}{2} + \frac{\theta_4}{l_w} \right) + \frac{r_G}{r} \frac{l_w}{2} \theta_2 + \frac{\theta_3}{r r_G} \frac{l_w}{2} \right)^{-1} \\
 a_3 &= - \left(\left(\frac{r}{r_G} \right)^2 \frac{\theta_1}{\theta_5} \left(m_w + \frac{m_b}{2} \right) + \frac{\theta_2}{\theta_5} + \frac{\theta_3}{r_G^2 \theta_5} \right)^{-1} \\
 a_4 &= - \left(\left(\frac{r}{r_G} \right)^2 \frac{1}{\theta_5} \left(m_w \theta_1 + \frac{2\theta_4}{l_w^2} \right) + \frac{1}{\theta_5} \left(\theta_2 + \frac{\theta_3}{r_G^2} \right) \right)^{-1}
 \end{aligned}$$

All physical parameters, including their meaning and nominal values, are indicated in [1].

VII. DIMENSIONING ELECTRICAL COMPONENTS

The electrical circuit used to measure motor voltage from Fig. 4b uses the electrical components from Table III. Before building a physical prototype, we checked the overall performance using the free simulator [16].

TABLE III: Electric components used for motor voltage measurement

Symbol	Nominal value (datasheet, color code, etc.)	Experimentally measured value	Part number
R_1	30 k Ω	30220 Ω	RS Stock No.: 148-843
R_2	6.98 k Ω	6980 Ω	RS Stock No.: 683-3972
R_3	3.9 k Ω	3835 Ω	-
R_4	820 Ω	807 Ω	-
C	280 nF	-	-
V_d	5.1 V	-	RS Stock No.: 136-4856

$$R_{2\parallel 4} = R_2 R_4 / (R_2 + R_4) = 723.36 \Omega$$

$$R_{1\parallel 3} = R_1 R_3 / (R_1 + R_3) = 3403.13 \Omega$$

$$R_{\parallel} = R_{1\parallel 3} R_{2\parallel 4} / (R_{1\parallel 3} + R_{2\parallel 4}) = 596.56 \Omega$$

REFERENCES

- [1] 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, pp. 3136–3141.
- [2] —, "A rule-based assistive control algorithm for safe navigation for a powered wheelchair," in *2020 European Control Conference*, Saint Petersburg, Russia, May 2020, pp. 1204–1209.
- [3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [4] F. L. Lewis, L. Xie, and D. Popa, *Optimal and Robust Estimation. With an Introduction to Stochastic Control Theory*, 2nd ed. CRC Press, 2008.
- [5] M. Ridolfi, S. Vandermeeren, J. Defraye, H. Steendam, J. Gerlo, D. De Clercq, J. Hoebeke, and E. De Poorter, "Experimental evaluation of UWB indoor positioning for sport postures," *Sensors*, vol. 18(1), no. 168, 2018.
- [6] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, 2nd ed. Springer, 2017, vol. 118.
- [7] L. Devigne, M. Aggravi, M. Bivaud, N. Balix, C. S. Teodorescu, T. Carlson, T. Spreters, C. Pacchierotti, and M. Babel, "Power wheelchair navigation assistance using wearable vibrotactile haptics," *IEEE Trans. Haptics*, vol. 13, no. 1, pp. 52–58, January–March 2020.
- [8] D. A. Abbink, T. Carlson, M. Mulder, J. C. de Winter, F. Aminravan, T. L. Gibo, and E. R. Boer, "A topology of shared control systems – finding common ground in diversity," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 509–525, 2018.
- [9] V. Narayanan, A. Spalanzani, and M. Babel, "A semi-autonomous framework for human-aware and user intention driven wheelchair mobility assistance," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea, October 2016, pp. 4700–4707.
- [10] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, 2nd ed. Hoboken, NJ: John Wiley & Sons, 2020.
- [11] P. Corke, "In situ measurement of motor electrical constants," *Robotica*, vol. 14, no. 4, pp. 433–436, 1996.
- [12] Pozyx Labs, accessed on May 10, 2020. [Online]. Available: <https://www.pozyx.io/documentation/creator/anchor-setup>
- [13] B. Perrat, M. J. Smith, B. S. Mason, J. M. Rhodes, and V. L. Goosey-Tolfrey, "Quality assessment of an ultra-wide band positioning system for indoor wheelchair court sports," *SAGE J.*, vol. 229, no. 2, pp. 81–91, April 2015.
- [14] K. Bader, B. Lussier, and W. Sch on, "A fault tolerant architecture for data fusion: A real application of Kalman filters for mobile robot localization," *Robotics and Autonomous Systems*, vol. 88, pp. 11–23, 2017.
- [15] C. Tessier, C. Cariou, C. Debain, F. Chausse, R. Chapuis, and C. Rousset, "A real-time, multi-sensor architecture for fusion of delayed observations: Application to vehicle localization," in *2006 IEEE Intelligent Transportation Systems Conference*, Toronto, Canada, September 2006, pp. 1316–1321.
- [16] SystemVision, accessed on June 12, 2020. [Online]. Available: www.systemvision.com
- [17] Hardware counter, accessed on May 10, 2020. [Online]. Available: <https://www.roboticshardwarestore.com/product/2x-32-bit-quadrature-encoder-counter-r2-spi-interface/>