Discrete-time Dynamic Modeling and Calibration of Differential-Drive Mobile Robots with Friction

Jong Jin Park[1], Seungwon Lee[2], and Benjamin Kuipers[3]

***Abstract*— Fast and high-fidelity dynamic model is very useful for planning, control, and estimation. Here, we present a fixed-time-step, discrete-time dynamic model of differential-drive vehicle with friction for reliable velocity prediction, which is fast, stable, and easy to calibrate.**

Unlike existing methods which are predominantly formulated in the continuous-time domain (very often ignoring dry friction) that require numerical solver for digital implementation, our model is formulated directly in a fixed-time-step discrete-time setting, which greatly simplifies the implementation and minimizes computational cost. We also explicitly take into account friction, using the stable formulation developed by Kikuuwe [1]. Friction model, while non-trivial to implement, is necessary for predicting wheel locks and velocity steady-states which occur in real physical systems.

In this paper, we present our dynamic model and evaluate it on a physical platform, a commercially-available electric powered wheelchair. We show that our model, which can run over 10^5 times faster than real-time on a typical laptop, can accurately predict linear and angular velocities without drift. The calibration of our model requires only a time-series of wheel speed measurements (via encoders) and command inputs, making it readily deployable to physical mobile robots.

I. INTRODUCTION

Fast and reliable vehicle models can be very useful for real-time motion planning and control. This is especially true for autonomous navigation, where the reliability and speed of the model are crucial for planning a feasible trajectory in real-time, where hundreds or thousand trajectories may have to be simulated and evaluated on-line each second to react to rapidly changing environment (e.g. [2]). For such applications, the model needs to be very fast (at least 10^5 times faster than real-time), accurate, and reliable. In particular, in needs to be able to predict non-linear phenomena such as wheel-locks and velocity steady-states induced by friction.

Kinematic models are simple and fast, but do not account for dynamics which is necessary for making accurate prediction. Also, while the kinematics of the wheeled mobile robots are well documented and understood, it is difficult to find a full dynamic model of a wheeled mobile robot with friction that satisfies our requirements. Existing dynamic models are predominantly formulated in the continuous-time domain, assume torque input, are often too slow [3], and also very often ignore dry friction. Seegmiller's recent thesis [4] provides an excellent literature review over the topic. (Also see [5], [6]).

Fig. 1. We calibrate and evaluate our dynamic model with a powered wheelchair, and show that it allows very fast (over 10^5 times faster than real-time) and reliable long-term prediction of vehicle velocities. The vehicle ($\sim$ 120 kg) has two drive wheels powered by two electric motors, four castor wheels, and a joystick input device.

Friction dynamics are very important for predicting velocity steady-states induced by friction under constant input and constant friction, and also for predicting wheel-locks when generated torque cannot overcome static friction. However, Coulumb friction and impact dynamics are fundamentally discontinuous phenomena [7] which are notoriously difficult to model, and it is well known that naïve continuous-time simulations with friction and impact can become numerically unstable. The best general approach (in 3D) to date in the continuous domain [5] employs numerical force-balance optimization, and achieves 10^3 times real-time in speed, with excellent agreement with real data.

Other classical approaches are usually formulated via Newton-Euler [8], [9], Lagrangian [10], [11], [12], [13], and more recently, the unified Newton-Euler-Lagrangian [6], but these approaches in general do not provide an explicit model for friction, nor a comparison with real data. There are 2D-dynamic models with wheel slip and tire friction [14], [15], [16], but they also do not have real-data evaluation and the equations quickly becomes cumbersome. Also, many of the physics-driven models require explicit measurements of mass, inertia, and friction parameters which can be difficult to obtain.

This paper presents a closed-form, fixed-time-step, discrete-time dynamic model for differential-drive mobile robots with friction, which is numerically stable and velocity estimate never diverge (i.e. no asymptotic drift). Our model assumes a 2D uniform surface and no-slip condition; and consists of an input device (e.g. joystick controller), two

*This work has taken place in the Intelligent Robotics lab in the Computer Science and Engineering Division of the University of Michigan. Research of the Intelligent Robotics lab is supported in part by grants from the National Science Foundation (CPS-0931474, IIS-1111494, IIS-1252987, and IIS-1421168).

[1] J. Park is with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA jongjinp@umich.edu

[2]S. Lee is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA leeswon@umich.edu

[3] B. Kuipers is with Faculty of Computer Science and Engineering, University of Michigan, Ann Arbor, MI 48109, USA kuipers@umich.edu

load-motor-wheel units with friction, and unicycle kinematics. The dynamics of the system is described as a grey-box model with jerk input. For handling friction, we incorporate a numerically stable closed-form solution proposed in [1], which is an excellent reference for different friction models for discrete-time systems but is not yet well known to vehicle modeling community.

We show that our model is fast and reliable, and is easy to calibrate and implement. Calibration of the model parameters only requires time-series of wheel speed measurements and commanded inputs. We evaluate our model against real data collected from a Quantum6000 electric powered wheelchair from Pride Mobility (Fig. 1), and show that it allows very fast (at least 10^5 times real-time on a typical laptop) and highly reliable long-term velocity predictions.

II. DYNAMIC MODEL OF THE DIFFERENTIAL-DRIVE VEHICLE

In this section, we describe our grey-box model (Fig. 2) for a differential-drive vehicle (an electric wheelchair). State vector for the robot is written as

$$q \equiv [s^R, \dot{s}^R, s^L, \dot{s}^L]^T \tag{1}$$

where s^R and s^L are the displacement of the right and the left wheel, respectively. Note that under the no-slip assumption there exists a bijective relation between the linear and angular velocities $[v\ \omega]^T$ and the wheel velocities $[\dot{s}_R, \dot{s}_L]^T$ (see (19)), thus $[s^R, \dot{s}^R, s^L, \dot{s}^L]^T \sim [v, \dot{v}, \omega, \dot{\omega}]^T$.

We will show that the dynamics of the system can be closely approximated by a difference equation (3) which is parametrized by 7 positive constants, c_1, c_2, α, β, γ, μ, l, where c_i are parameters for the input device, $\alpha, \beta, \gamma, \mu$ are motor and friction parameters, and l is a length of axle between the two wheels. Formally, we write

$$\begin{aligned} q_{k+1} &= f(q_k, u_k) \\ &= f(q_k, u_k; c_1, c_2, \alpha, \beta, \gamma, \mu, l) \end{aligned} \tag{2}$$

where $q_k \equiv q(t_k)$ is the state of the vehicle at time t_k, and $u_k \equiv u(t_k) \equiv [u_k^f, u_k^l]^T$ are forward and lateral commands to the system during the time interval $[t_k, t_{k+1})$.

A. Motor Dynamics with Friction

This subsection describes our model for the load-motor-wheel subsystem. This model assumes a DC motor connected to a wheel under a constant unknown load on a planar uniform surface. Dropping superscripts R (right) and L (left) for simplicity, we have the following state-space representation for each wheel in our discrete-time model for the load-motor-wheel subsystem,

$$\begin{bmatrix} \dot{s}_{k+1} \\ \ddot{s}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & h \\ -\beta h & 1-\gamma h \end{bmatrix} \begin{bmatrix} \dot{s}_k \\ \ddot{s}_k \end{bmatrix} + \begin{bmatrix} h \\ 0 \end{bmatrix} g_r(\dot{s}_k, \ddot{s}_k; \mu) + \begin{bmatrix} 0 \\ \alpha h \end{bmatrix} V_k \tag{3}$$

where s_k, $\dot{s}_k$, $\ddot{s}_k$ are displacement, speed, motor-generated acceleration of the wheel at time t_k; $g_r(\cdot)$ is acceleration due to external force, which we limit to mechanical friction in this work; and V_k is the (voltage) input to each motor unit. The state vector for each motor model is $[\dot{s}_k\ \ddot{s}_k]^T$, and $h \equiv t_{k+1} - t_k$ is the fixed time interval. The positive constants α, β and γ are coefficients for the input gain, velocity- and current-induced energy loss for the system, respectively. See Sect. II-B for derivation.

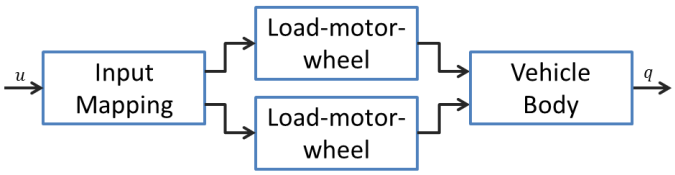

Fig. 2. Cartoon diagram of the model. Native control input to the system u maps to input signals for two load-motor-wheel subsystems with friction. For load-motor-wheel subsystem, we assume only the input command and wheel speeds are measurable. See text for details.

We implement a Coulomb friction model for acceleration due to load-induced mechanical friction $g_r(\cdot)$:

$$g_r(\dot{s}_k, \ddot{s}_k; \mu) = \begin{cases} \bar{\mu}(\cdot) & \text{if } |\bar{\mu}(\cdot)| \le \mu \\ \mathrm{sgn}(\bar{\mu}(\cdot))\,\mu & \text{otherwise} \end{cases} \tag{4}$$

which is adapted from Kikuuwe [1], where

$$\bar{\mu}(\dot{s}_k, \ddot{s}_k) \equiv -\frac{\dot{s}_k}{h} - \ddot{s}_k \tag{5}$$

and the positive constant μ is a parameter for the maximum Coulomb friction-induced acceleration. Note that the friction parameter μ is a function of load and surface condition.

The key feature of (4)-(5) is that the speed drops exactly to zero in a single step if $|\bar{\mu}(\cdot)| \le \mu$. This removes the need for special event detection and/or additional constraint-enforcing numerical optimizations near zero speed, which ensures numerical stability but can be expensive. See Kikuuwe [1] for an excellent reference, and extensions to other (more sophisticated) discrete-time friction models.

Note that this formulation is highly modular in the sense that any external acceleration (e.g. due to slope) can be added along with the friction term in $g_r(\cdot)$, thus extension to 3D (e.g. angled surfaces) should be straightforward.

B. Derivation of Discrete-time Motor Dynamics

Our model is formulated directly in discrete-time setting from first principles. For DC Motor, we have Newton's Law

$$J\ddot{\phi} = -b\dot{\phi} + \tau \tag{6}$$

where $\dot{\phi}$ is angular speed of the motor.

Motor torque τ is proportional to current,

$$\tau = Ki \tag{7}$$

which gives

$$i = \frac{J}{K}\ddot{\phi} + \frac{b}{K}\dot{\phi} \tag{8}$$

We also have Kirchhoff's Law,

$$V = L\frac{\mathrm{d}i}{\mathrm{d}t} + Ri + K\dot{\phi} \tag{9}$$

Substituting, we get

$$\begin{aligned}\frac{1}{L}V &= \frac{\mathrm{d}i}{\mathrm{d}t}+\frac{R}{L}i+\frac{K}{L}\dot{\phi} \\ &= \frac{J}{K}\frac{\mathrm{d}}{\mathrm{d}t}\ddot{\phi}+\frac{b}{K}\ddot{\phi}+\frac{R}{L}[\frac{J}{K}\ddot{\phi}+\frac{b}{K}\dot{\phi}]+\frac{K}{L}\dot{\phi} \\ &= \frac{J}{K}\frac{\mathrm{d}}{\mathrm{d}t}\ddot{\phi}+[\frac{b}{K}+\frac{RJ}{LK}]\ddot{\phi}+[\frac{Rb}{LK}+\frac{K}{L}]\dot{\phi} \qquad (10)\end{aligned}$$

Converting this to discrete-time format, with Euler approximation for differentiation,

$$\frac{\mathrm{d}}{\mathrm{d}t}\ddot{\phi} \simeq \frac{1}{h}[\ddot{\phi}_{k+1}-\ddot{\phi}_k] \qquad (11)$$

where subscript denote the k-th time-step, we can now write

$$\frac{1}{L}V_k = \frac{J}{K}\frac{1}{h}[\ddot{\phi}_{k+1}-\ddot{\phi}_k]+[\frac{b}{K}+\frac{RJ}{LK}]\ddot{\phi}_k+[\frac{Rb}{LK}+\frac{K}{L}]\dot{\phi}_k \qquad (12)$$

which becomes, after consolidation of coefficients and rearrangement,

$$\ddot{\phi}_{k+1} = \ddot{\phi}_k - \gamma\ddot{\phi}_k - \beta h\theta_k + \alpha h V_k \qquad (13)$$

where α, β, and γ are positive constants.

Finally, with Euler integration of velocity, we have

$$\dot{\phi}_{k+1} = \dot{\phi}_k + h\ddot{\phi}_k \qquad (14)$$

Combining (13)-(14) in matrix form, we have

$$\begin{bmatrix}\dot{\phi}_{k+1}\\ \ddot{\phi}_{k+1}\end{bmatrix} = \begin{bmatrix}1 & h\\ -\beta h & 1-\gamma h\end{bmatrix}\begin{bmatrix}\dot{\phi}_k\\ \ddot{\phi}_k\end{bmatrix}+\begin{bmatrix}0\\ \alpha h\end{bmatrix}V_k \qquad (15)$$

which converted to (3) via incorporation of the friction model (4) and conversion of the angular displacement ϕ to linear displacement s.

C. Steady-state Analysis

The load-motor-wheel subsystem (3) is a double integrator, since Voltage input adds to acceleration, and acceleration adds to velocity. It is well known that dynamic models, which are integrators, can drift. We provide straightforward steady-state analysis of our model and show that our model predicts velocity steady-states where motor torque is canceled by friction and wheel locks where motor torque cannot overcome friction.

As we will see in Sect. III-B, this matches well with the observed behavior of the vehicle. Our velocity estimate does not drift, due to the friction and energy loss terms.

Suppose the load-motor-wheel subsystem (3) is in steady state so that $\dot{s}_k = \dot{s}_\infty$ and $\ddot{s}_k = \ddot{s}_\infty$ for $^\forall k$ under some constant input V_∞. To begin, assume the vehicle is moving forward at constant speed, i.e. $\dot{s}, \ddot{s} > 0$ and $|\bar{\mu}(\cdot)| > \mu$, then we have, from (3)-(4):

$$\begin{bmatrix}\dot{s}_\infty\\ \ddot{s}_\infty\end{bmatrix} = \begin{bmatrix}1 & h\\ -\beta h & 1-\gamma h\end{bmatrix}\begin{bmatrix}\dot{s}_\infty\\ \ddot{s}_\infty\end{bmatrix}+\begin{bmatrix}-\mu h\\ 0\end{bmatrix}+\begin{bmatrix}0\\ \alpha h\end{bmatrix}V_\infty \qquad (16)$$

so that

$$\begin{cases}\dot{s}_\infty &= \frac{\alpha}{\beta}V_\infty - \frac{\mu\gamma}{\beta} \quad (V_\infty > \frac{\mu\gamma}{\alpha}) \\ \ddot{s}_\infty &= \mu\end{cases} \qquad (17)$$

which means the steady-state speed is a function of constant input, and when in steady state the constant motor acceleration is canceled out by constant friction, as expected.

Performing this analysis for all cases, we can write the steady-state velocity as a function of steady-state input as

$$\dot{s}_\infty = \begin{cases}\frac{\alpha}{\beta}V_\infty - \frac{\mu\gamma}{\beta} & \text{if } V_\infty > \frac{\mu\gamma}{\alpha} \\ \frac{\alpha}{\beta}V_\infty + \frac{\mu\gamma}{\beta} & \text{if } V_\infty < -\frac{\mu\gamma}{\alpha} \\ 0 & \text{otherwise}\end{cases} \qquad (18)$$

where $-\frac{\mu\gamma}{\alpha} \le V_\infty \le \frac{\mu\gamma}{\alpha}$ is the friction-induced deadband, i.e. wheel locks due to friction.[1]

Simple dynamic/kinematic models without dry friction, which are usually double integrators, simply cannot predict this deadband. This has important implications for predicting vehicle motion, especially when it operates at low speed including start and stop. This is extremely important for safety.

D. Differentially Driven Vehicle Body

For the motion of differential-drive vehicles on 2D plane under no-slip condition, we can write

$$\begin{bmatrix}v\\ \omega\end{bmatrix} = \begin{bmatrix}\frac{1}{2} & \frac{1}{2}\\ \frac{1}{l} & -\frac{1}{l}\end{bmatrix}\begin{bmatrix}\dot{s}^R\\ \dot{s}^L\end{bmatrix} \qquad (19)$$

where v and ω are linear and angular velocity of the rigid body, l is the length of the axle between the two drive wheels which is assumed to be known, and $\dot{s}^R$ and $\dot{s}^L$ are the speed of the right and the left wheel.

E. Input Mapping

The native command input to our system (Fig. 1) is forward-backward (u_k^f) and left-right (u_k^l) joystick positions which is updated at the start of every time interval $[t_k, t_{k+1})$. Empirically, we have found the following non-linear model for joystick-input-to-motor-input mapping work well for our system

$$\begin{bmatrix}V_k^R\\ V_k^L\end{bmatrix} = \begin{bmatrix}1 & g_u(u_k^f)\\ 1 & -g_u(u_k^f)\end{bmatrix}\begin{bmatrix}u_k^f\\ u_k^l\end{bmatrix} \qquad (20)$$

where V_k^R and V_k^L are the voltage inputs to the right and the left motors. Since voltage input to DC-motors controls the derivative of motor torque, Voltage can be described as a jerk input to the system.

$$g_u(u_k^f) = c_1(1 - c_2|u_k^f|) \qquad (21)$$

is a non-linear scaling factor for the lateral joystick position which models interaction between forward-backward and left-right joystick commands, where c_1 and c_2 are non-zero positive constants that parameterize this mapping. Eq. (21) models that the magnitude of lateral input (rotational command) is scaled down linearly as the magnitude of the forward input increases, which is often the case in commercially available platforms.

[1] Although outside the scope of this paper, Eq. (18) immediately suggests a model-based feedforward control as it maps control input to desired steady-state velocity. This mapping can easily be inverted after collapsing nullspace ($V_\infty = 0$ if $\dot{s}_\infty = 0$), and can also be combined with weak feedback for faster convergence. See [2] for an example.

III. MODEL CALIBRATION AND EVALUATION

A. Data Collection and Model Calibration

We calibrate six model parameters $\zeta \equiv [c_1, c_2, \alpha, \beta, \gamma, \mu]$ (the length of the axle l is measured directly) via solving an error-minimization problem over an ensemble of M trajectories (time-series) of measurements, where each trajectory contains N data points of speed measurements $[\tilde{v}, \tilde{\omega}]^T$ and recorded control inputs $u = [u^f, u^l]^T$. Specifically, we have

$$\underset{\zeta}{\text{minimize}} \quad L = \sum_{j=1}^{M} \Big(\sum_{i=k_j+1}^{k_j+N} \big(|\hat{v}_i - \tilde{v}_i| + c_\omega |\hat{\omega}_i - \tilde{\omega}_i| \big) \Big) \tag{22}$$

$$\text{subject to} \quad \hat{q}_{i+1} = f(\hat{q}_i, u_i; \zeta, l) \tag{23}$$

$$\hat{q}_{k_j} = [\tilde{v}_{k_j}, \tilde{\omega}_{k_j}, \frac{\mathrm{d}}{\mathrm{d}t}\tilde{v}_{k_j}, \frac{\mathrm{d}}{\mathrm{d}t}\tilde{\omega}_{k_j}]^T \tag{24}$$

where hat ($\hat{\cdot}$) represents an estimate, tilde ($\tilde{\cdot}$) represents a measurement, c_ω is a weight (we have $c_\omega = 1$), $f(\cdot,\cdot)$ is the dynamics of the system, and q_{k_j} is the state at start time t_{k_j} of each series of measurements. Namely, from M selected data points, we simulate the vehicle state for N steps in the future using each initial estimate $\hat{q}_{k_j}, (j = 1...M)$, and compute absolute error in velocity space using total of $M \times N$ samples. We perform numerical differentiation of each measurement using 5-point differencing to compute the derivatives.

We have collected wheel speed measurements (via encoders) and native command inputs (joystick commands) from our wheelchair robot (Fig. 1). Our training data (Fig. 3-5, Fig. 11-12), consists of 37.5 minutes (2250 seconds) of driving data from various test runs under standard step, sinusoid, and random command signals of varying magnitude. It contains total of 37500 samples with sampling time of 0.06 sec. From the collected measurements, we have extracted uniformly spaced $M = 3800$ (overlapping) trajectory segments of length $N = 50$ (3 seconds) for calibration.

For easy calibration, we have allowed the data to contain partial corruption due to unmodeled effects of castor wheels, bumps on the ground, varying ground conditions and load, which has motivated our choice of L1 norm in the cost function (Eq. (22)), as the impact of the large deviations due to those unmodeled effects are less pronounced compared to usual L2 norm.

Our calibration problem (22)-(24) was solved with a numerical optimizer, *fmincon*, in MATLAB with an interior-point algorithm. With the training set, it converged to $\zeta_* = [0.4265, 0.006491, 0.2315, 6.548, 4.073, 0.1676]^T$ after 160 iterations, with each iteration taking about 13 sec (total 35 min) on a 2.8 GHz i7 processor.

We want to emphasize that having the ensemble of short simulations of N steps into the future was very important for the convergence of the numerical optimizer. In fact, with too short a simulation ($N = 1$, single-step prediction) or too long a simulation ($N > 1000$) the optimizer was unable to find a good solution. In general, the length of prediction N should be long enough to capture the steady-state of the system, but not too long to make error (which depends on the prediction length) too sensitive to parameter change.

TABLE I
VELOCITY PREDICTION ERRORS WITH ROBOT-DRIVEN DATA

	abs. err.	std.
v (m/s)	0.045	0.059
ω (rad/s)	0.075	0.11

TABLE II
VELOCITY PREDICTION ERRORS WITH HUMAN-DRIVEN DATA

	abs. err.	std.
v (m/s)	0.029	0.033
ω (rad/s)	0.091	0.091

Note that calibration of our model requires only the time-series of wheel speed measurements and joystick inputs. This is important since for many practical applications where wheel speeds and the native command inputs are often the only available variables for measurements.

Fig. 3-5 shows selected subsets of the training data. (See Appendix for the full training set.) Although this is not part of our quantitative evaluation, we also show a long-term simulation result for the full 2250 sec using the fitted model parameters, the recorded control sequences, and *only* the initial state estimate at $t = 0$. Overall, the model is able to capture the slightly underdamped transient responses, the steady-states, and the wheel-locks due to friction very well, and does not drift. Considering the calibration was done using the collection of 3 second trajectories, we believe this is an excellent result. See captions for details.

B. Model Evaluation

For quantitative evaluation, we compare the long-term simulation from the model and the measured robot speed from (1) 500 seconds of robot-controlled [2] driving (Fig. 6-8) and (2) 350 seconds of human-controlled driving (Fig. 9-10). In both cases, the initial state at $t = 0$ is recursively propagated forward using the recorded command signal and the model calibrated using the training set. The predictions do not drift, and match the measurements very well.

The computational load for this forward simulation is minimal, due to the simplicity of the model. The simulation can run at well over 10^3 times real-time in MATLAB, and over 10^5 times real-time in C++. Furthermore, implementation is extremely simple as it does not require numerical solver.

IV. DISCUSSION AND CONCLUSION

In this paper, we present a discrete-time dynamic model for differential-drive mobile robots, where friction is a key component which allows accurate estimation of velocity steady-states and wheel-locks observed in physical systems.

By combining fundamental electromechanics and a stable discrete-time friction model [1], we have constructed a highly expressive dynamic model of electrically powered vehicles with minimal number of parameters (Sect. II). Our model can be calibrated with only the speed measurements and native control inputs via straightforward numerical optimization (Sect. III.A.), which can predict robot speed with

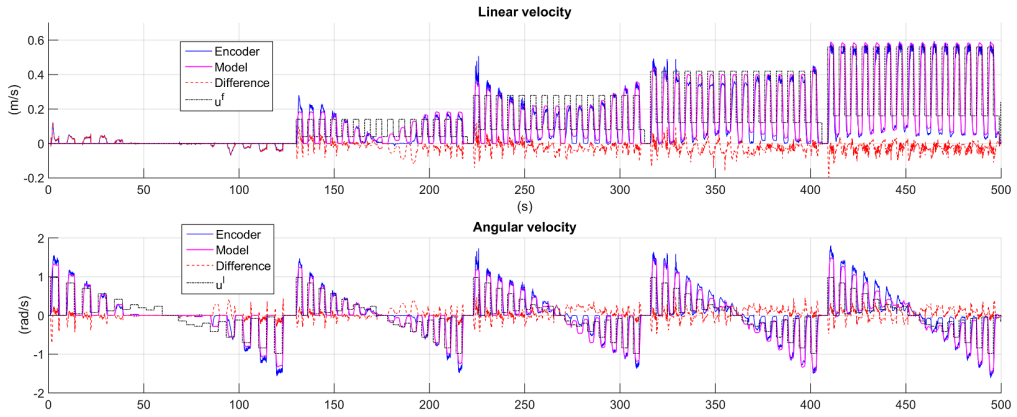

Fig. 3. (Best viewed in color) A series of step inputs in the first 500 seconds of the training set. Measured speed (blue), model prediction from $t = 0$ (magenta, bold) using the calibrated parameters, and scaled command inputs (dotted black) are error (dotted red) are shown. The system seems to be slightly underdamped and the model is able to reproduce the behavior and the steady-states of the system.

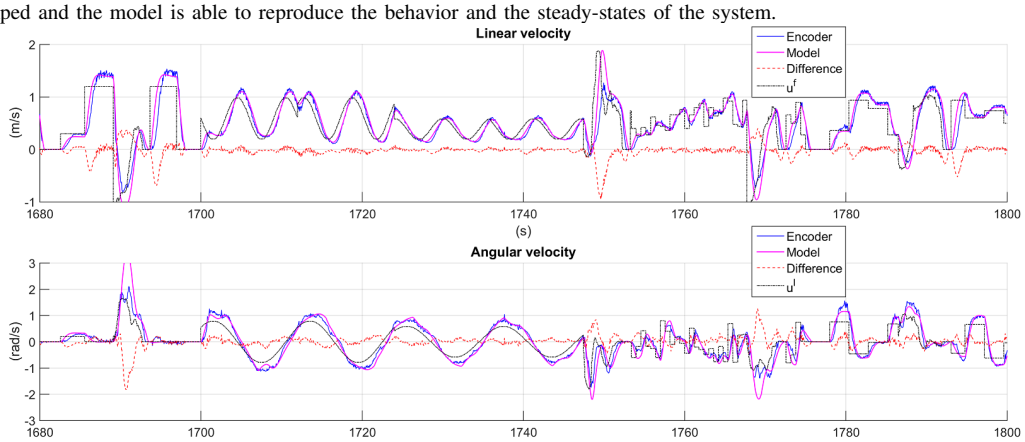

Fig. 4. (Best viewed in color) Sinusoidal and random step inputs in the two-minute time interval [1680, 1800] (s). Again, model response is simulated from $t = 0$ without further state measurements. The predicted speed using the time-varying commands matches well with data.

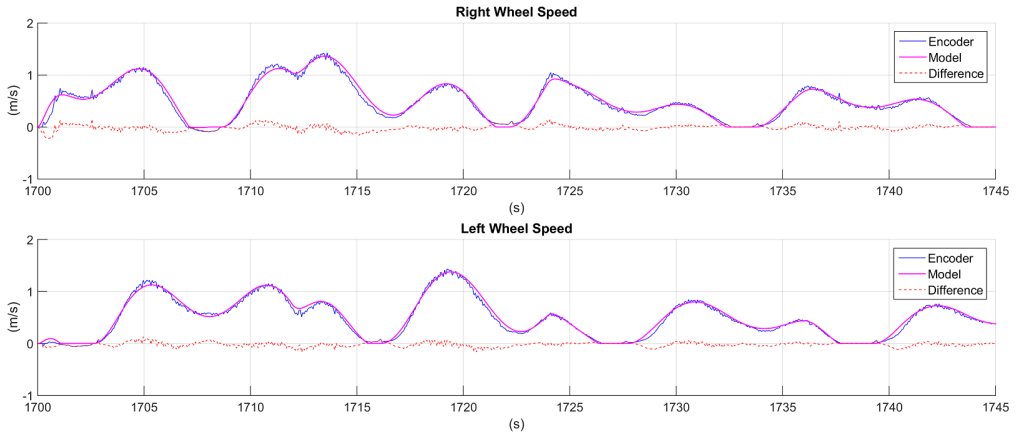

Fig. 5. (Best viewed in color) Closer look at the right and left wheel velocities in the time interval [1700 1745] of the training set. The model can predict wheel locks ($t = 1707, 1721, 1733$, top, $t = 1716, 1727, 1738$, bottom) due to friction while the robot maneuvers under sinusoidal command input, which results in interesting velocity curves in Fig.4 in [1700 1745]

high accuracy even in very long-term simulations. This is due to the encoded steady-states in the model (Sect III.B) which prevents the model from drifting asymptotically. We have shown that our model, with complexity and speed comparable to a simple kinematic model, can predict both transient and steady-state velocity outputs of the dynamical system with high accuracy (Sect. IV).

The model, however, is built on a limiting assumptions (e.g. uniform planar surface, no-slip condition), and ignores important source of disturbances such as castor wheels and the passenger. We would like to extend this work for more general case to handle those external variables in the near future. In particular, we are interested in using convolutional neural nets for modeling general dynamic systems.

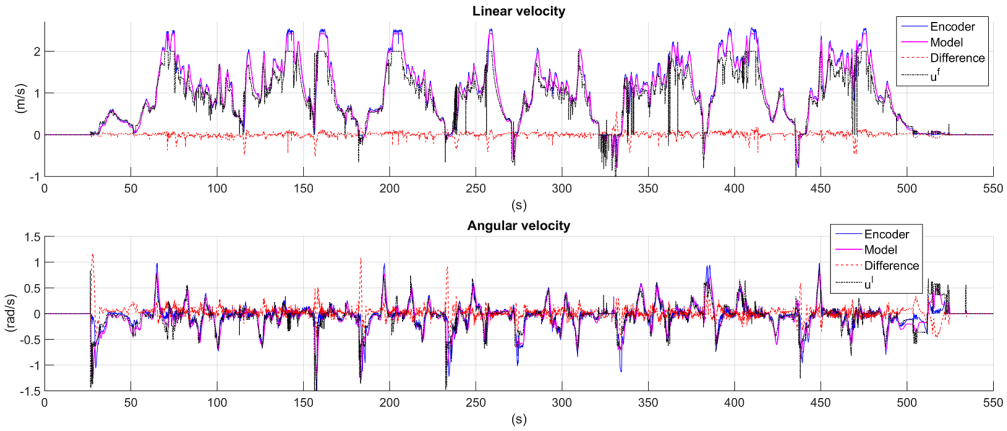

Fig. 6. (Best viewed in color) 500 seconds of robot-controlled driving, which is our first test set. Measured speed (blue), model prediction from $t = 0$ (magenta, bold), scaled command inputs (dotted black), and error (dotted red) are shown. The model is able to predict the robot speed for the entire duration without asymptotic drift without using state measurements. See Fig. 7-8 for zoomed-in views, and Table I for statistics.

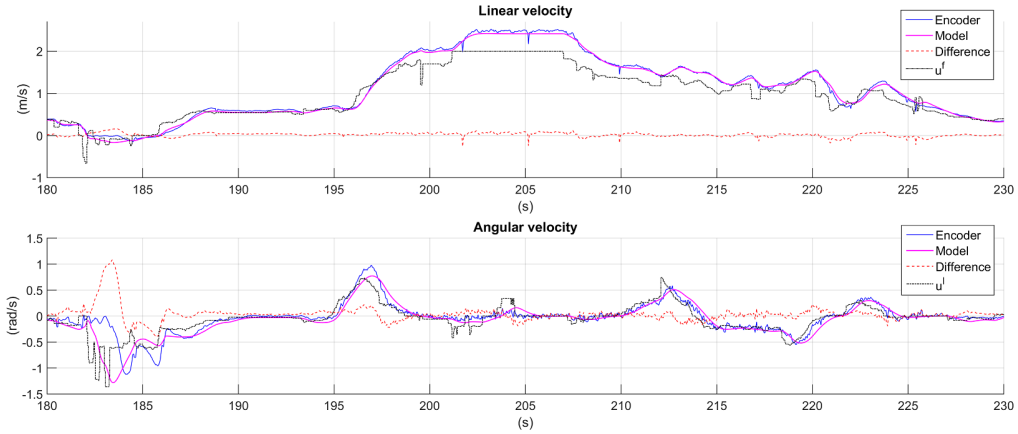

Fig. 7. (Best viewed in color) Zoomed-in view of the robot-controlled driving in the time interval [180, 230]. Note that the model prediction quickly recovers from large deviation near $t = 183$, where unaligned castor wheels caused near 2 second delay in vehicle response. Castors and unmodeled effects can cause large disturbances, but due to the steady-state guarantee our model can quickly recover to the steady-states encoded in the fitted model parameters.

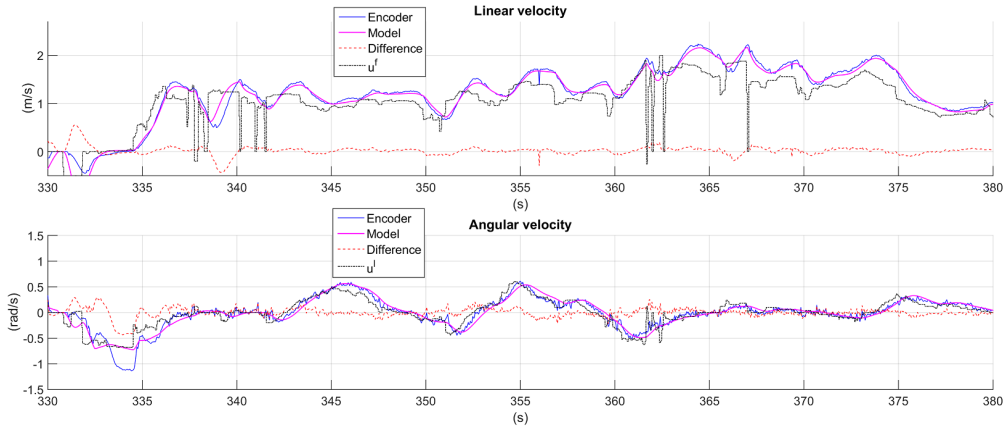

Fig. 8. (Best viewed in color) Zoomed-in view of the robot-controlled driving in the time interval [330, 380].

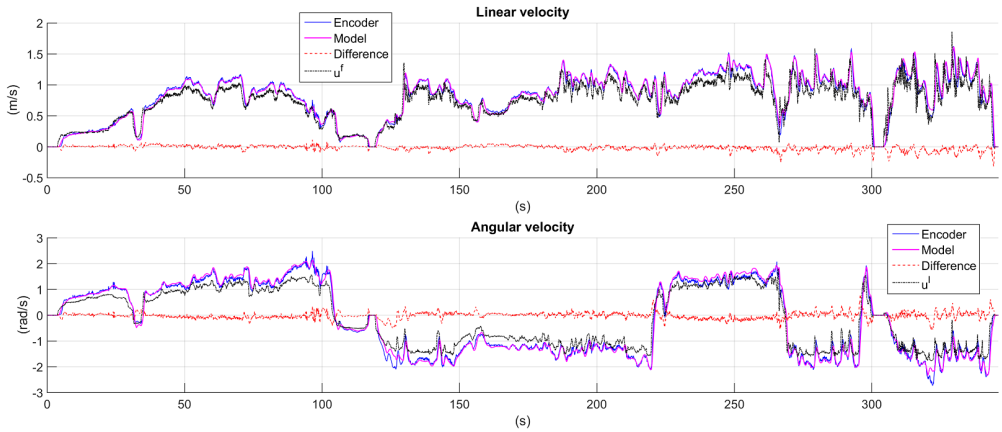

Fig. 9. (Best viewed in color) 350 seconds of human-controlled driving, which is our second test set. Measured speed (blue), model prediction from $t = 0$ (magenta, bold), scaled command inputs (dotted black), and error (dotted red) are shown. The model is able to predict the robot speed for the entire duration without asymptotic drift using only the initial state and the command input, although the command tends to be more noisy than the inputs in the training set. See Table II for statistics.

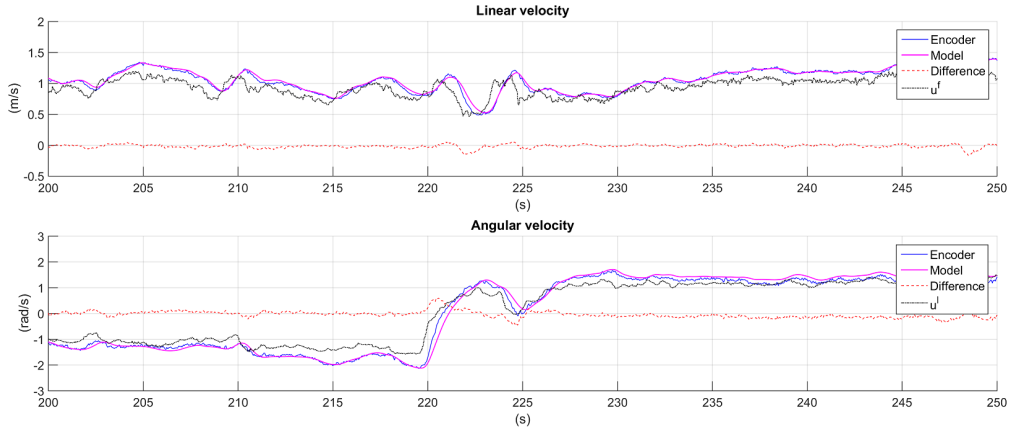

Fig. 10. (Best viewed in color) Zoomed-in view of the human-controlled driving in the time interval [200, 250].

APPENDIX

A. Full Training Set

Fig. 11-12 shows the entire data used for training the model, along with the 2250-sec long simulation from our model using the initial state at ($t = 0$) and the command sequences u. See Sect. III-B for detailed analysis.

REFERENCES

[1] R. Kikuuwe, N. Takesue, A. Sano, H. Mochiyama, and H. Fujimoto, "Fixed-step friction simulation: from classical Coulomb model to modern continuous models," in *2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Aug 2005, pp. 1009–1016.

[2] J. Park, "Graceful navigation for mobile robots in dynamic and uncertain environments," Ph.D. dissertation, University of Michigan, Apr 2016.

[3] T. Howard, "Adaptive model-predictive motion planning for navigation in complex environments," Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-09-32, 2009.

[4] N. A. Seegmiller, "Dynamic model formulation and calibration for wheeled mobile robots," Ph.D. dissertation, Carnegie Mellon University, Oct 2014.

[5] N. Seegmiller and A. Kelly, "High-fidelity yet fast dynamic models of wheeled mobile robots," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 614–625, Jun 2016.

[6] R. Dhaouadi and A. A. Hatab, "Dynamic modelling of differential-drive mobile robots using Lagrange and Newton-Euler methodologies: A unified framework," *Advances in Robotics and Automation*, vol. 2, no. 2, 2013.

[7] D. E. Stewart, "Rigid-body dynamics with friction and impact," *SIAM Review*, vol. 42, no. 1, pp. 3–39, 2000.

[8] L. Huttenhuis, C. van Heteren, and T. J. A. de Vries, "Modelling and control of a fast moving, highly maneuverable wheelchair," in *Proceedings of the International Biomechatronics workshop*, Apr 1999, pp. 110–115.

[9] W. H., B. Salatin, G. G. Grindle, D. Ding, and R. A. Cooper, "Real-time model based electrical powered wheelchair control," *Medical Engineering and Physics*, vol. 31, no. 10, pp. 1244–1254, Dec 2009.

[10] T. Fukao, H. Nakagawa, and N. Adachi, "Adaptive tracking control of a nonholonomic mobile robot," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 609–615, Oct 2000.

[11] A. Albagul and Wahyudi, "Dynamic modelling and adaptive traction control for mobile robots," *International Journal of Advanced Robotic Systems*, vol. 1, no. 3, pp. 149–154, 2004.

[12] S. K. Saha and J. Angeles, "Kinematics and dynamics of a three-wheeled 2-DOF AGV," in *1989 IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 1989, pp. 1572–1577 vol.3.

[13] B. d'Andrea Novel, G. Bastin, and G. Campion, "Modelling and control of non-holonomic wheeled mobile robots," in *1991 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Apr 1991, pp. 1130–1135 vol.2.

[14] N. Sidek and N. Sarkar, "Dynamic modeling and control of nonholonomic mobile robot with lateral slip," in *2008 Third Int. Conf. on*

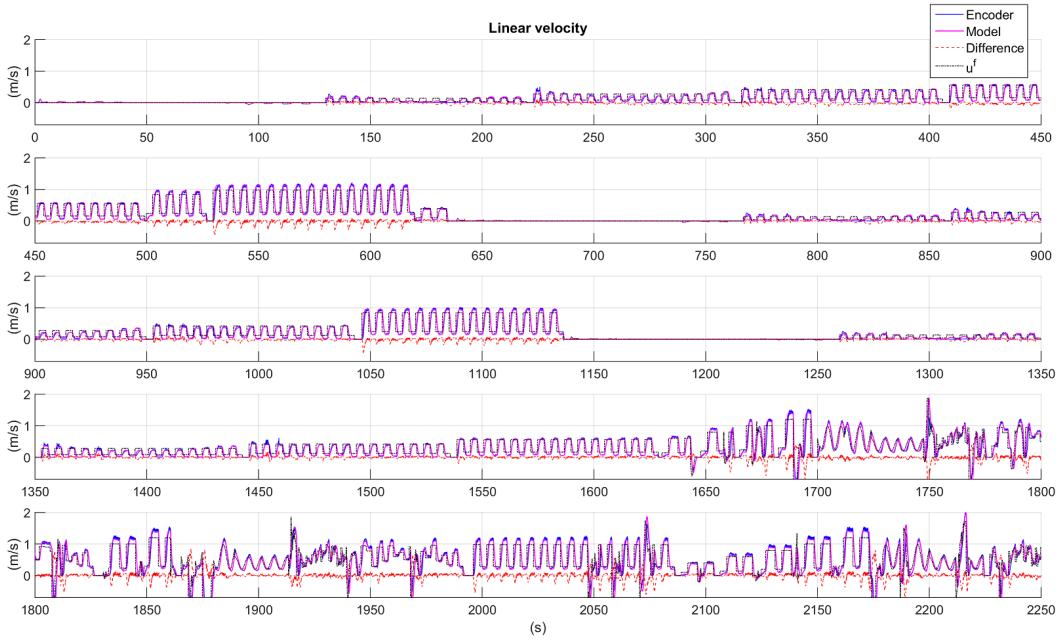

Fig. 11. (Best viewed in color) Linear velocity measurements (blue) and joystick commands (dotted black) in the training set. We also show Long-term simulation result (2250 sec) propagated forward from the initial state (magenta, bold).

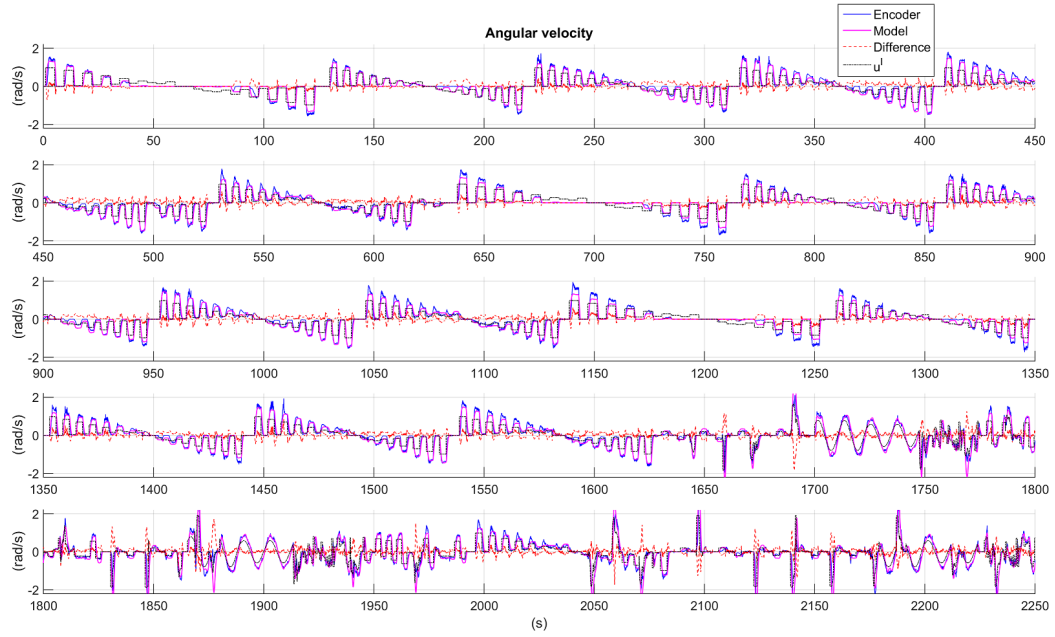

Fig. 12. (Best viewed in color) Angular velocity measurements (blue) and joystick commands (dotted black) in the training set. We also show Long-term simulation result (2250 sec) propagated forward from the initial state (magenta, bold).

Systems (ICONS), Apr 2008, pp. 35–40.

[15] Y. Tian, N. Sidek, and N. Sarkar, "Modeling and control of a nonholonomic wheeled mobile robot with wheel slip dynamics," in *2009 IEEE Symposium on Computational Intelligence in Control and Automation*, Mar 2009, pp. 7–14.

[16] S. Nandy, S. N. Shome, R. Somani, T. Tanmay, G. Chakraborty, and C. S. Kumar, "Detailed slip dynamics for nonholonomic mobile robotic system," in *2011 IEEE Int. Conf. on Mechatronics and Automation*, Aug 2011, pp. 519–524.

[17] C. M. Wang, "Location estimation and uncertainty analysis for mobile robots," in *1988 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Apr 1988, pp. 1231–1235.

[18] B. W. Johnson and J. H. Aylor, "Dynamic modeling of an electric wheelchair," *IEEE Transactions on Industry Applications*, vol. IA-21, no. 5, pp. 1284–1293, Sep 1985.

[19] D. K. Hanna and A. Joukhadar, "A novel control-navigation system-based adaptive optimal controller & EKF localization of DDMR," *International Journal of Advanced Research in Artificial Intelligence*, vol. 4, no. 5, pp. 21–29, May 2015.

[20] K. Thanjavur and R. Rajagopalan, "Ease of dynamic modelling of wheeled mobile robots (WMRs) using Kane's approach," in *Robotics and Automation*, Apr 1997, pp. 2926–2931.