2014 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS 2014)
September 14-18, 2014, Chicago, IL, USA

Experimental Analysis of Models for Trajectory Generation on Tracked Vehicles

Jonathan R. Fink and Ethan A. Stump

Abstract— **We begin to bridge the gap between high-level motion planning and execution by adopting models to abstract the complicated skid-steer vehicle dynamics and evaluating their suitability as motion predictors for a feed-forward control framework. We consider three kinematic motion models and a drivetrain model in experiments on two surface types with a small tracked vehicle. We perform statistical analysis of the predictive accuracy of these models when used to create optimal open-loop plans for a set of canonical maneuvers and discuss the applicability of these models for a closed-loop control framework.**

I. INTRODUCTION

Though it is perhaps the simplest drive design to implement, the humble skid-steer ground vehicle platform actually represents an incredibly complex dynamic system that can only by fully understood by delving into the murky realms of contact mechanics and shear deformations. Though we often like to analyze the system as if it touched the earth with two point ground contacts, the fact that it does not means that any turn can only be accomplished by some surface sliding across another. Even for the canonical case of sliding on homogenous surfaces, we must deal with models involving empirical parameters and discontinuous effects; if we consider sliding over granular media such as dirt or gravel, we must begin to think about continuum mechanics.

In reality, there is a large gap between the abstract treatment of vehicles in a kinematic planning framework (e.g. [1]) and the actual execution of planned trajectories. These issues are commonly avoided by just moving very slowly so that the vehicle acts more like a purely kinematic system, but we want to be able to plan and act with the full speed of our vehicles, motivated by any number of requirements on operation tempo or scale.

One argument would be to rely on feedback to avert the issue of unmodeled dynamics and accurately track our required trajectories. We contend that this is not the complete answer for three reasons:

1) There is no obvious error signal available at the rate and scale of a single, rapid maneuver: GPS errors are too large to be helpful on the scale of meters; visual odometry is slow and dependent on favorable, feature-rich environments; accurate inertial navigation is too expensive and bulky; and odometry is subject to the same dynamic errors that we are trying to use feedback to counteract!
2) Feedback assumes that we approximately understand the relationship between control signals and error dynamics and that this relationship can be linearized. In reality, vehicle platforms have motor saturation, time delays, and velocity profiling on top of the nonlinear effects of terrain interaction.
3) Trajectory tracking for nonlinear systems works best when incorporated into a two-degree-of-freedom controller [2] that requires feed-forward planning in addition to feedback.

Having a good feed-forward model also lets us broaden our considerations to include optimality and use techniques such as receding horizon control [3].

In this paper, we begin to bridge the gap between high-level trajectory generation and execution by adopting models to abstract the complicated skid-steer vehicle dynamics and evaluating their suitability as motion predictors for a feed-forward control framework. After discussing the related work, we present the modeling methodology and then apply it to developing predictive kinematic motion models for a treaded skid-steer vehicle on two different experimental surfaces. By comparing to ground truth, we perform statistical analysis of the predictive accuracy of these models when used to create optimal open-loop plans for a set of canonical maneuvers. Finally we conclude with a discussion about the applicability of these models for a closed-loop control framework.

II. RELATED WORK

Though fully understanding the motion of a skid-steer vehicle requires modeling the dynamics of terrain interaction, some authors have presented useful kinematic approximations. Understanding and estimating the kinematic slip is performed for a four-wheeled vehicle in [4] and for a tracked vehicle in [5]. Central to the model is the concept that turning motions with slip have instantaneous centers of rotation that are located forward or backward from the centerline of the platform, in contrast to the ideal differential drive model. Other models treat the slip as simple ratios between wheel/tread speed and effective travel speed [6] or velocity-dependent disturbances with longitudinal, lateral, and angular components [7].

Simplified dynamics are studied in [8] after discussion of kinematic approximations. A more thorough treatment of the vehicle-terrain interaction is given in [9] for the case of treaded vehicles driving on loose dirt/sand and in [10] for wheeled vehicles in a variety of soil types. [11] summarizes decades of study in terramechanics for off-road

United States Army Research Laboratory, 2800 Powder Mill Rd, Adelphi, MD 20783 `(jonathan.r.fink3.civ, ethan.a.stump.civ)@mail.mil`

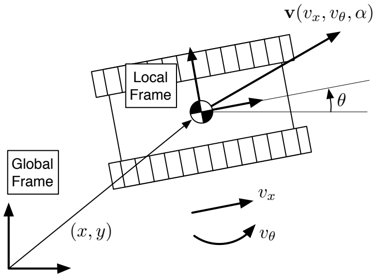


Fig. 1. Frames and coordinates of our planar treaded robot. The ideal differential drive velocity is v_x forward and v_θ turning, but we correct these values with a kinematic slip model to find the more accurate velocity $\mathbf{v}(v_x, v_\theta, \alpha)$.

wheeled and tracked vehicles and provides what is perhaps the only complete dynamic model for a turning tracked vehicle, though it is for quasi-static motion and may not be applicable to small platforms.

It is important to note that some of these works explicitly seek to perform online estimation of the slip models by using an IMU [4], differential-GPS [7], or visual odometry [12].

The work of [13] takes the approach of using a discontinuous Coulomb friction model to study the stick-slip behavior of a four-wheeled skid-steer vehicle and finding transition relations that enable the resulting hybrid system to match periodic position measurements. Because the entire tread is slipping during motion, it is not directly applicable, but suggests that it is not impossible to build discontinuous dynamic models in a real system.

III. METHODOLOGY

Our system model must be composed of two separate pieces: the interaction between tread and ground to produce motion, and the platform dynamics at work when turning commanded velocities into actual tread velocities. We choose to focus here on developing kinematic approximations to vehicle slip rather than more expressive dynamic models to see how well they perform before increasing the model complexity.

A. Kinematic Motion Models

We treat the effects of unmodeled vehicle dynamics, including tread-terrain interactions, by incorporating corrections to the effective velocity of the platform. These corrections are established by experimentally fitting relevant models from the literature [8], [12] to a series of maneuvers performed over a range of speeds and turning rates. By comparing the difference between the predicted and actual final position at the end of each motion, we show the value of using these corrections to help remove the systematic bias from the trajectory integration.

The frames and coordinates are summarized in Fig. 1. The position of the robot is found by integrating the planar kinematics to find position:

TABLE I

VELOCITY MODELS USED IN KINEMATIC EQUATIONS

Model	*Equation*	$\lvert\alpha\rvert$
Ideal Differential Drive	$\mathbf{v} = \begin{bmatrix} v_x \\ 0 \\ v_\theta \end{bmatrix}$	0
Effective Wheel-Base [8]	$\mathbf{v} = \begin{bmatrix} v_x \\ 0 \\ \alpha \cdot v_\theta \end{bmatrix}$	1
General Kinematic Slip [12]	$\mathbf{v} = \begin{bmatrix} v_x \\ 0 \\ v_\theta \end{bmatrix} + C(v_x, v_\theta) \cdot \alpha$	9

$$\dot{\rho} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{A(\theta)} \mathbf{v}(v_x, v_\theta, \alpha) \tag{1}$$

Here α is our set of kinematic correction parameters and v_x and v_θ are the forward and turning velocities we would expect to achieve under an ideal differential drive model given our tread velocities. For a platform tread width of B, these can be found from the left and right tread velocities (v_L and v_R) as:

$$v_x = \frac{v_L + v_R}{2} \qquad v_\theta = \frac{v_R - v_L}{B}$$

The three kinematic models considered are summarized in Table I. The ideal differential drive model treats the treads as point contacts with the ground and assumes they do not slip. In the *Effective Wheel-Base* model [8], we consider that rolling resistance from the treads skidding over the ground during turning can reduce the effective turning velocity and scale it accordingly. The authors of [8] point out that this is equivalent to considering an ideal differential drive model with an effectively larger wheel base. Lastly, in the *General Kinematic Slip* model (our modification of the model presented in [12]), we consider that slip conditions may lead to velocity changes in both the longitudinal and lateral directions. Following the authors of [12], we allow the correction to be velocity-dependent by defining:

$$C(v_x, v_\theta) = \begin{bmatrix} \mathbf{c}_x & & \\ & \mathbf{c}_y & \\ & & \mathbf{c}_\theta \end{bmatrix}_{3\times 9} \tag{2}$$

$$\mathbf{c}_x = \begin{bmatrix} v_x & \lvert v_\theta\rvert & v_x\lvert v_\theta\rvert \end{bmatrix}$$

$$\mathbf{c}_y = \begin{bmatrix} v_x & v_\theta & v_x v_\theta \end{bmatrix}$$

$$\mathbf{c}_\theta = \begin{bmatrix} v_x & v_\theta & v_x v_\theta \end{bmatrix}$$

To fit values of α for the models, we would ideally need to establish motions where we can compare the expected velocity and actual velocity under a wide range of conditions. However, since we lack the sensor capabilities to accurately measure platform velocities in the field, we settle instead for studying the perturbative effect of the parameters on the integrated trajectory in the spirit of [14]. As presented in

[12], this could be done by looking at a larger motion where we can get accurate relative pose information over extended time-slices (such as by visual odometry or a tactical-grade INS) and then fitting parameters to minimize the error between expected final pose and actual final pose. For this effort, we chose to use a motion capture system for the sake of establishing an accurate baseline to compare these more field-appropriate techniques to in the future.

We seek to choose α to minimize the squared-distance between our measured trajectory, $\rho_m(t)$, and our predicted trajectory, $\rho(t;\alpha)$, found by solving Eq. 1 from start time t_0 to end time t_1 with parameter values α. The objective function is:

$$f(\alpha) = \frac{1}{2}\int_{t_0}^{t_1} \|\rho(t;\alpha) - \rho_m(t)\|^2 dt \tag{3}$$

We can solve this more quickly by utilizing the results of [14] to find the sensitivity of the integrated trajectory to the parameters. The crux of the technique lies in recognizing that the derivative of Eq. 1 with respect to the parameters:

$$\begin{gathered}\dot{\rho}(t) = A(\theta)\mathbf{v}(t)\\ \frac{\partial\dot{\rho}}{\partial\alpha} = \frac{d}{dt}\frac{\partial\rho}{\partial\alpha} = \frac{\partial(A\mathbf{v})}{\partial\rho}\frac{\partial\rho}{\partial\alpha} + A\frac{\partial\mathbf{v}}{\partial\alpha}\end{gathered}$$

has special structure that allows us to explicitly evaluate the linear transition matrix, $\Phi(t,t_0)$, and write the derivative of the pose as a function of time:

$$\frac{\partial\rho(t)}{\partial\alpha} = \Phi(t,t_0)\frac{\partial\rho(t_0)}{\partial\alpha} + \int_{t_0}^{t}\Phi(t,\tau)A(\theta(\tau))\frac{\partial\mathbf{v}(\tau)}{\partial\alpha}d\tau \tag{4}$$

With this relationship, we can use the Jacobian of Eq. 3 (assuming that all poses are relative to the trajectory start so $\rho = \rho_m = 0$),

$$\frac{\partial f}{\partial\alpha} = \int_{t_0}^{t_1}\left[(\rho-\rho_m)^T\int_{t_0}^{t}\Phi(t,\tau)A(\theta(\tau))\frac{\partial\mathbf{v}(\tau)}{\partial\alpha}d\tau\right] \tag{5}$$

in a nonlinear least-squares technique (i.e. Levenberg-Marquardt) to quickly arrive at a fit. Evaluation of Eq. 5 is done efficiently by recognizing that at each time step of the evaluation of the outside integral, we can reuse the inside integral from the last time step and push it through Eq. 4 with the addition of the next piece of the inside integral.

B. Drive-System Modeling

The kinematic motion models presented above assume known tread velocities v_L, v_R as measured based on encoders situated on the motors themselves. This is suitable when the goal is to accurately estimate the platform motion – the so-called odometry problem. We are instead interested in *predicting* the platform motion given time-varying control inputs to the treads. While instantaneous tread-speed models that rely on the underlying closed-loop system to match control inputs may be suitable for certain low-speed regimes, we are focusing on operating in high-speed regimes that approach the capability limits of the platform. Thus, we must consider dynamic models for the input-output response of the full powertrain.

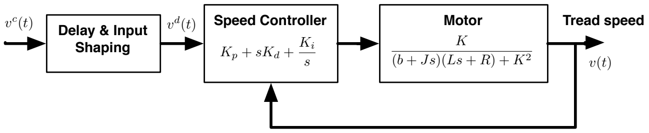


Fig. 2. Block diagram of the input-output system for the left and right treads

We can model a tracked or skid-steered vehicle with two independent closed-loop motor systems that have commanded tread velocity $v_L^c(t), v_R^c(t)$ as input and measured velocity $v_L(t), v_R(t)$ as output. In general, we consider a drive-system consisting of three components as depicted in Fig. 2: (1) command-velocity time-delay, shaping, and/or profiling, (2) a closed-loop control system, e.g., proportional-integral-derivative (PID) control, and (3) the second-order system describing the electrodynamic system including the direct-current motors and gearing.

Time-delay of the input signal to each motor controller is the result of network or communication delays incurred between the algorithms performing trajectory generation and the motor controller itself. We model this delay by assuming that the input to the closed-loop motor control system $v^d(t)$ will be delayed from the commanded input $v^c(t)$, i.e., $v^d(t) = v^c(t-\tau)$. Other input shaping such as acceleration and velocity limits are found through system specifications or experimental identification.

We assume a PID control loop is working to achieve the desired motor velocity. Performing standard transfer function analysis, we can model the dynamic system that results from a PID control loop wrapped around a direct current (DC) motor. The PID control transfer function is given by

$$C(s) = K_p + sK_d + \frac{K_i}{s} \tag{6}$$

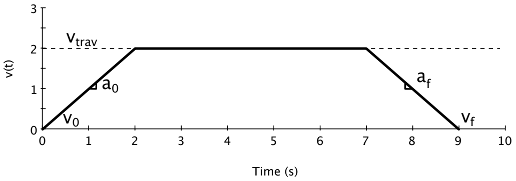
where the controller proportional, integral, and derivative gains are given by K_p, K_i, K_d respectively. The DC motor transfer function is

$$M(s) = \frac{K}{(b+Js)(Ls+R)+K^2} \tag{7}$$

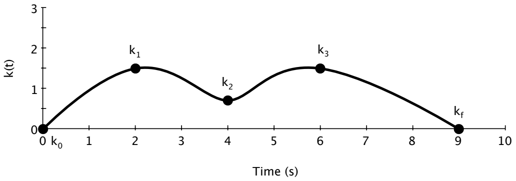
with parameters for the moment of inertia of the rotor (J kg m^2), the viscous friction constant (b N m s), the electrical resistance (R Ohm), and inductance (L H). The force and motor torque constants are equal and given by K V/rad/s or K N m/Amp. The transfer function for the full closed-loop system is then

$$\frac{C(s)M(s)}{1+C(s)M(s)}. \tag{8}$$

This model can be used to generate differential equations which can be simulated for given system parameters and control inputs.



(a) Trapezoidal profile



(b) Spline-based profile

Fig. 3. Examples of the parameterized control inputs

C. Trajectory Design

Optimization-based trajectory design is the process of minimizing a cost function defined over the vehicle trajectory $\rho(t)$ by choosing time-varying control inputs $\mathbf{u}(t), t \in [0, t_f]$. While trajectory design or motion planning methods that search lattices of states induced by discrete sampling over control inputs [1] are popular in recent literature, we are more interested in trajectory design on the *continuous* space of control input functions.

Consequently, we adopt the general strategy presented by Howard and Kelly [15] in which the control input function is parameterized $\mathbf{u}(t; \mathbf{p})$ and we seek to minimize a cost functional $\Psi(\mathbf{p})$ which will generally be defined as an integral along the predicted path of the vehicle. For this work, we are interested in platforms that have control inputs $\mathbf{u} = \left[v_x, v_\theta\right]^{\mathrm{T}}$ but we will parameterize our input function as $\mathbf{u}(t) = \left[v_x(t), \kappa(t)\right]^{\mathrm{T}}$ where $\kappa(t) = v_\theta(t)/v_x(t)$ is the curvature of the path. Using curvature rather than angular velocity directly allows us to decouple the shape of the path and the desired speed along the path. While this decoupling breaks down in light of the slip models presented in Sec. III-A, we have observed that it does improve the performance of numerical optimization.

Again from [15], we consider two forms of time-varying parameterized functions for the components of $\mathbf{u}(t)$. First, we consider a basic trapezoidal velocity profile as depicted in Fig. 3(a) with parameters $\left[v_0, a_0, v_{\text{trav}}, a_f, v_f\right]$ representing the initial velocity, initial acceleration, traveling velocity, final acceleration, and final velocity respectively. We also rely on a spline-based representation of control input where the parameters are a set of evenly spaced knot points $\left[k_0, k_1, \ldots, k_N\right]$ as depicted in Fig. 3(b).

For the purposes of this work, we assume that a desired trajectory for the platform is given $\rho_d(s) \, \forall \, s \in [0, s_f]$ and we wish to track it as closely and quickly as possible. That is

$$\Psi(\mathbf{p}) = w_t t_f + w_p \int_0^{t_f} \min_s \|\rho(t) - \rho_d(s)\| \, dt \tag{9}$$

where weights w_t and w_p favor the performance, i.e., minimizing time to reach the goal, or safety, i.e., minimizing the deviation from the desired path, respectively. The optimization problem to be solved can then be written

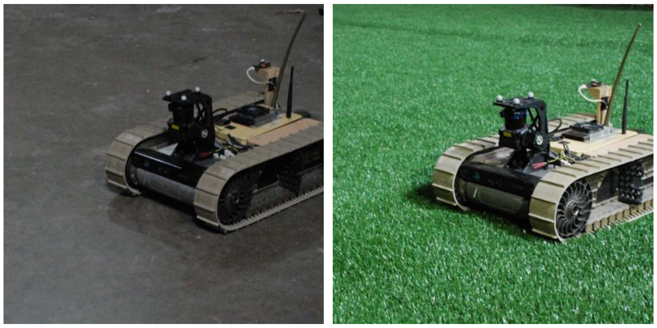


Fig. 4. Our treaded vehicle on our two experimental surfaces: dusty concrete and artificial turf.

$$\begin{aligned} \min_{\mathbf{p}} \quad & \Psi(\mathbf{p}) \\ \text{subject to} \quad & \rho(t_f) = \rho_d(s_f) \\ & \rho(t) = \rho(0) + \int_0^t \dot{\rho}(\tau) \, d\tau. \end{aligned} \tag{10}$$

There are numerous ways to address the optimization problem in (10). One approach is to achieve vehicle-model independence by relying on numerical estimates of the partial derivatives necessary to compute the system Jacobian and Hessian with respect to the input parameterization and apply standard gradient-descent methods [15]. From the point of view of a practical and efficient implementation, we rely on the *NLOpt* library [16]. In particular, we employ the constrained optimization by linear approximations (COBYLA) method [17]. Problems with time horizons of 4 s and parameter vectors of dimension 20 are solved between 0.5 and 1.0 s.

As discussed in [15], the speed of this nonlinear optimization is highly dependent on the initial conditions for the parameter vector $\mathbf{p}$. We shall see that in our experiments, this is trivially satisfied since we start with time-varying control inputs that produce the desired path under an ideal differential drive model.

IV. EXPERIMENTS

All experiments for this work were conducted with the iRobot *Packbot* [18] as depicted in Fig. 4. The *Packbot* is a ground platform equipped with a skid-steer tracked drive system with on-board computation. The base platform weighs 18 kg and is capable of 2 m/s speeds. The robots used in our experiments are additionally equipped with an ad hoc 802.11 wireless radio, Microstrain 3DM-GX2 inertial measurement unit (IMU), Hokuyo UTM-30LX scanning laser range finder with 30 m range, and Quad-Core Intel i7 computing payload. For the purposes of modeling and verifying the performance of different kinematic slip models before implementation in the field, we utilize a high-quality VICON motion capture system [19] to provide ground-truth pose measurements in a laboratory setting.

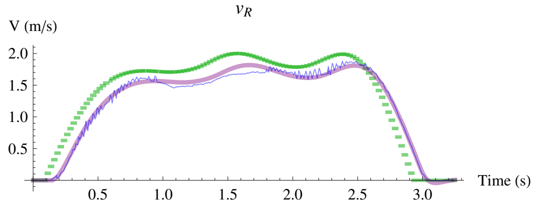


Fig. 5. Commanded (green), actual (blue), and modeled (purple) tread velocity after parameter identification

TABLE II

DRIVE-SYSTEM MODEL PARAMETERS

	Input shaping	
$\tau = 0.049$ ms	$a_{\text{max}} = 5.5$ m/s^2	
	Control system	
$K_p = 56.49$	$K_i = 1.11$	$K_d = 2.77$
	Motor system	
$J = 0.059$ kg m^2	$R = 0.254$ Ohm	$L = 0.253$ H
$K = 0.0294$ N m/Amp	$b = 0.456$ N m s	

We procede with experimental results detailing the model identification and parameter selection for the dynamic motor model presented in Sec. III-B and the kinematic motion models presented in Sec. III-A. With model parameter values that improve the ability to predict vehicle motion given control-input functions, we show how optimization-based trajectory generation techniques can be used to improve the performance of a high-speed trajectory tracking application. We demonstrate all of these results on two surfaces: smooth concrete and artificial turf as depicted in Fig. 4.

A. Drive-System Model Identification

The *Packbot* utilizes two independent speed controllers for the left and right treads. Parameter identification of the closed-loop drive-system is accomplished by analysis of the commanded input $v_L^c(t), v_R^c(t)$ and the measured tread speeds $v_L(t), v_R(t)$. By commanding the tread with a step input and measuring the response, we can characterize the time-delay τ and acceleration limits a_{max}. Over 100 experimental trials, we measure the time between the commanded step input and the first non-zero measured tread speed to estimate delay $\tau = 0.049$ s. Examining the initial response of the tread velocity to a step input, we find a linear response indicating an acceleration limit of approximately $a_{\text{max}} = 5.5$ m/s^2.

We then construct a set of time-varying control inputs that satisfy the maximum acceleration constraint imposed by the *Packbot* velocity shaping and again measure the tread speed time response. Using the commanded input and measured output, we can optimize the parameters of the second-order model described in (8) with gradient-free methods such as Nelder-Mead given reasonable initial conditions. Figure 5 depicts the actual motor response to a commanded input signal along with the simulated output of the second-order system (8) using optimized parameters in Table II.

B. Kinematic Slip Model Identification

In order to evaluate the kinematic slip models presented in Sec. III-A, we execute a series of time-varying control inputs, integrate vehicle motion according to the kinematic models in Table I and the measured tread speeds, and compare the resulting vehicle pose with accurate measurements from an external motion capture system. To ensure that the sample trajectories adequately represent all possible control inputs, we employ a trapezoidal profile to specify both the linear velocity and the curvature of the input, i.e.,

TABLE III

POSE ERROR STATISTICS FOR DIFFERENT MODELS AND SURFACES

		$\Delta\rho_x$ (m)		$\Delta\rho_y$ (m)		$\Delta\rho_\theta$ (rad)	
	Model	μ	σ^2	μ	σ^2	μ	σ^2
Concrete	*Ideal*	-0.762	0.248	-0.115	0.372	-0.043	0.768
	Eff.WB	-0.019	0.016	-0.092	0.028	-0.006	0.026
	Gen.KS	-0.011	0.014	-0.015	0.007	-0.016	0.021
Turf	*Ideal*	-0.371	0.056	-0.021	0.088	-0.006	0.166
	Eff.WB	0.023	0.009	-0.043	0.007	0.019	0.009
	Gen.KS	0.016	0.008	-0.002	0.006	0.002	0.007

$$\mathbf{p} = \left[v_0, a_0, v_{\text{trav}}, a_f, v_f, \kappa_0, a\kappa_0, \kappa_{\text{trav}}, a\kappa_f, \kappa_f\right]$$

with $v_0 = v_f = \kappa_0 = \kappa_f = 0.0$, $a_0 = a_f = 5$ m/s^2, $a\kappa_0 = a\kappa_f = 1$ 1/m s^2. The traveling velocity and curvature are swept through all combinations of $v_{\text{trav}} \in \left[0.5, 0.6, \ldots, 2.0\right]$ m/s and $\kappa_{\text{trav}} \in \left[\pm\frac{1}{4}, \pm\frac{1}{2}, \pm\frac{3}{4}, \pm 1, \pm 1\frac{1}{2}, \pm 2\right]$ 1/m subject to a maximum tread velocity of 2 m/s. The time-length of each control input was chosen so that the path length was approximately 4 m. Each time-varying control input was executed twice, resulting in a set of over 300 trajectories, accumulating nearly 1 km of data.

Given this large set of trajectory data with measured tread speeds and accompanying ground truth positions, we apply the methods described in Sec. III-A to learn parameters and compare the performance of the different kinematic motion models. The value of using these kinematic models is demonstrated in Fig. 6, where we show the final pose error, $\Delta\rho = \rho(t_1) - \rho_m(t_1)$, for each of the kinematic models applied to the measured tread velocities of all of the trajectories captured while driving on the concrete floor. Each point shows where the system thought the robot ended up relative to where it actually ended up according to the motion capture system. The 9-parameter General Kinematic Slip model gives a tight, near-Gaussian distribution with little bias. The Effective Wheel-Base model has a much improved bias, but the distribution appears non-Gaussian. Nevertheless, it is clear that even this 1-parameter model provides substantial benefits over the ideal differential drive model. Table III enumerates the mean and variance of the final pose error for each model type on the two surface types tested – concrete and artificial turf. The resulting model parameters are in Table IV.

C. Open-loop Trajectory Tracking

Given a full system model that accurately predicts the motion of the vehicle for a given time-varying control input, we demonstrate the improved performance of optimization-based trajectory generation. In particular, we consider the ap-

TABLE IV

EXPERIMENTALLY-DETERMINED SLIP PARAMETERS FOR TWO MODELS ON TWO SURFACES

	Effective Wheel-Base: α		*General Kinematic Slip:* $\begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \alpha_4 & \alpha_5 & \alpha_6 \\ \alpha_7 & \alpha_8 & \alpha_9 \end{pmatrix}$	
Surface	Left	Right	Left	Right
Concrete	0.7061	0.6471	$\begin{pmatrix} -0.0022 & 0.0355 & -0.0386 \\ 0.0409 & -0.0173 & -0.0602 \\ -0.0549 & -0.2785 & 0.0605 \end{pmatrix}$	$\begin{pmatrix} -0.0035 & -0.0525 & -0.0082 \\ 0.0165 & -0.0155 & -0.0368 \\ 0.0435 & -0.3126 & 0.0245 \end{pmatrix}$
Turf	0.7938	0.7758	$\begin{pmatrix} -0.0162 & -0.0053 & 0.0384 \\ 0.0102 & 0.0160 & -0.0323 \\ 0.0130 & -0.2001 & -0.0297 \end{pmatrix}$	$\begin{pmatrix} -0.0044 & -0.0277 & 0.0090 \\ 0.0245 & 0.0216 & -0.0427 \\ -0.0080 & -0.2615 & 0.0492 \end{pmatrix}$

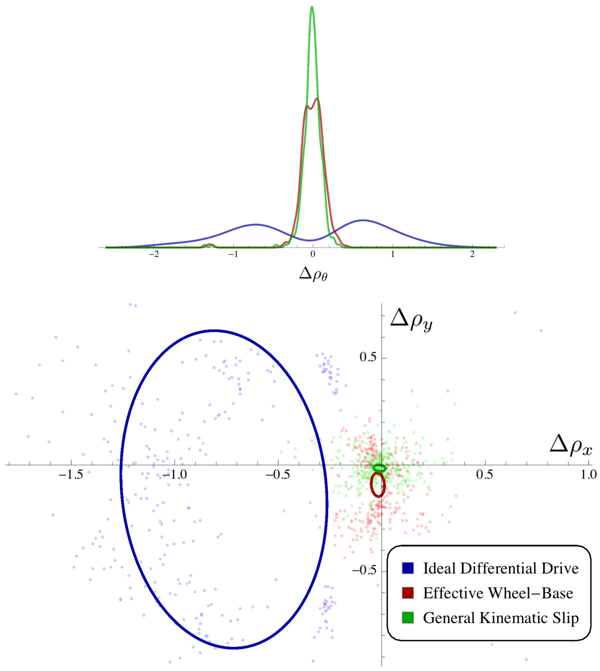


Fig. 6. Comparison of error in final trajectory pose for three different kinematic models. The upper plot shows error in orientation and the lower plot shows error in translation. The ellipses signify the 2-σ covariance of the data, though really only the results of applying the General Kinematic Model have a distribution that appears it could be Gaussian.

plication where a global motion planning algorithm produces a path that is obstacle-free and satisfies certain high-level objectives such as achieving a particular waypoint. While modern motion planning algorithms are designed to generate kinematically feasible paths that are derived from achievable control inputs, they do not typically account for complete models of the vehicle motion such as those presented above.

In the following results, we aim to address some of the questions that arise when attempting to compute a time-varying control input that produces vehicle motion to match a path derived with ideal or simplified motion models. We present three canonical trajectories generated with the trapezoidal velocity profile, spline-based curvature profile, and ideal differential drive model: a sharp left turn, a sharp right turn, and s-curve. The parameter vectors for each trajectory's control input are in Table V.

TABLE V

CANONICAL REFERENCE INPUTS

$$\mathbf{p}_v = [0 \text{ m/s}, 5 \text{ m/s}^2, 1 \text{ m/s}, 5 \text{ m/s}^2, 0 \text{ m/s}]$$

Input parameters	t_f	v	κ_0	κ_1	κ_2	κ_3	κ_4	κ_5
$\mathbf{p}_L$	4 s	$\mathbf{p}_v$	0	0	1.5	0	0	0
$\mathbf{p}_R$	4 s	$\mathbf{p}_v$	0	0	−1.5	0	0	0
$\mathbf{p}_S$	6 s	$\mathbf{p}_v$	0	0	−1	0	1	0

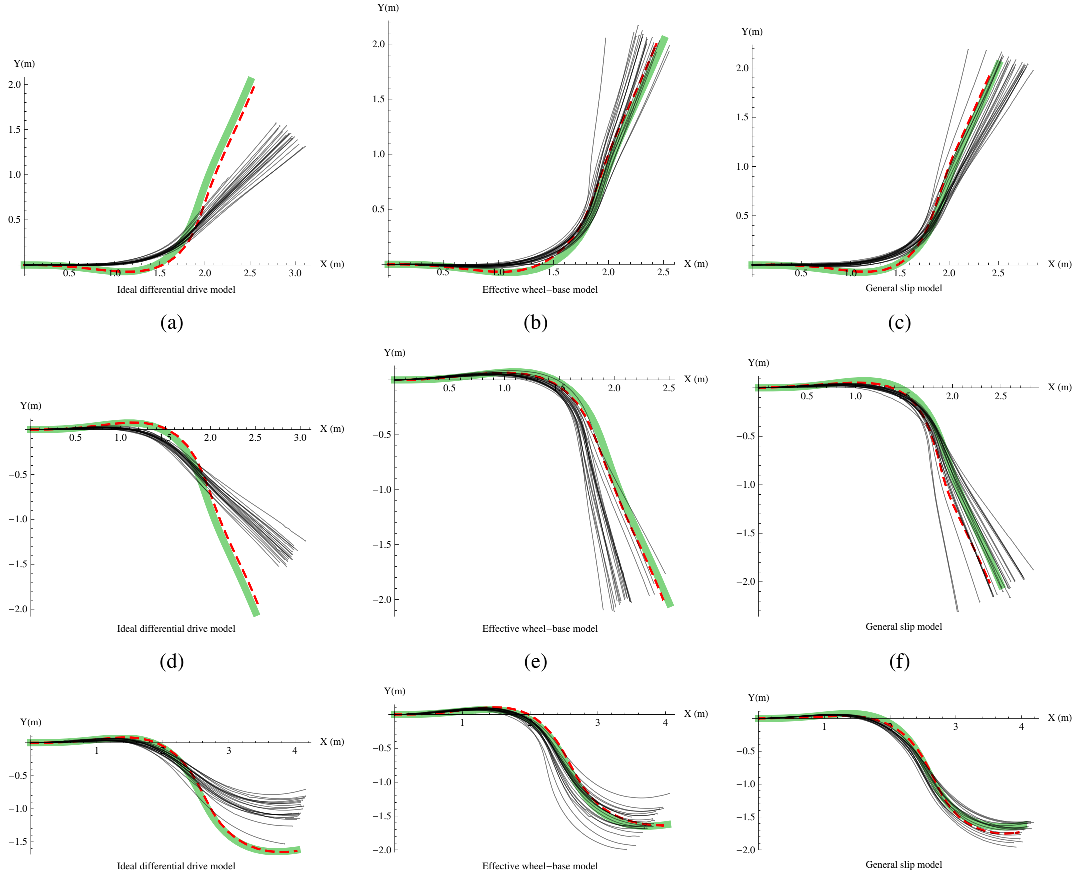
Given the path defined by each canonical trajectory, we compute optimal time-varying control inputs using spline-based profiles for velocity and curvature as per the method described in Sec. III-C. For each trajectory type, we compute optimal control inputs for the ideal differential drive model, the Effective Wheel-Base model, and the General Kinematic Slip model. That is, we find inputs that minimize (9) with $w_t = 0.5$ and $w_p = 1.0$ in order to follow the desired path as closely and quickly as possible. Each of these control inputs is executed 20 times to observe the statistical behavior. The resulting trajectories according to the motion capture system are depicted in Fig. 7.

There are many metrics that can be used to evaluate the performance of a trajectory optimization and tracking application. Of primary interest is the maximum path deviation defined as

$$\max_t \min_\tau \|\rho_m(t) - \rho_d(\tau)\| dt$$

where $\rho_d(t)$ is the desired path and $\rho_m(t)$ is the measured path. Measuring the maximum path deviation for each of 20 trials across the three canonical trajectory types, we can compare the performance of the ideal differential drive model, the Effective Wheel-base model, and the General Kinematic Slip model. The distribution of maximum path error for trials conducted on the concrete surface is depicted in Fig. 8 (computed by automatic kernel density estimation [20]) and statistical values are in Table VI. Maximum deviation from the desired path is drastically reduced when moving away from the ideal differential-drive model. Additional improvement is observed when moving from the Effective Wheel-base to the General Kinematic Slip model.

Of secondary interest in evaluating optimized control inputs for the canonical trajectories is the minimization of



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 7. Execution of canonical trajectories on concrete with optimized control inputs using the ideal differential drive model ((a), (d), (g)), the Effective Wheel-Base model ((b), (e), (h)), and the General Kinematic Slip model ((c), (f), (i)). The thicker green path depicts the desired trajectory, the dashed red path depicts the trajectory optimized under the given motion model, and the thin black paths depict the actual paths according to the motion capture system.

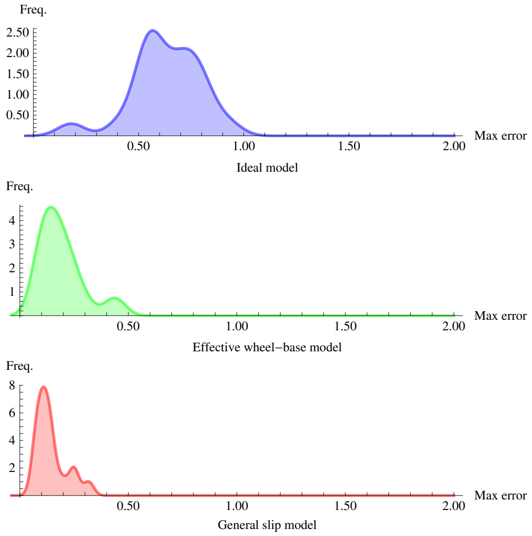


Fig. 8. Distribution of maximum path deviation when executing optimized control inputs for the canonical trajectories on the concrete surface. Each distribution is the result of 20 trials on each of the three trajectory types.

TABLE VI

MAXIMUM PATH DEVIATION ON CONCRETE AND GRASS

Ideal Differential Drive

Concrete			Grass		
Mean	Max	Std. Dev.	Mean	Max	Std. Dev.
0.61	0.93	0.16	0.90	1.50	0.15

Effective Wheel-Base

Concrete			Grass		
Mean	Max	Std. Dev.	Mean	Max	Std. Dev.
0.23	0.58	0.14	0.74	1.13	0.18

General Kinematic Slip

Concrete			Grass		
Mean	Max	Std. Dev.	Mean	Max	Std. Dev.
0.18	0.53	0.10	0.77	1.13	0.20

time necessary to complete the trajectory. Vehicle speed while traversing each trajectory is limited by many factors including the saturation of individual tread speeds, acceleration limits, and transverse slip velocities incurred when considering the General Kinematic Slip model. On average, the optimization process was able to find trajectories that were 70%, 80%, and 79% the time of the original nominal trajectory for the ideal differential drive, Effective Wheel-base, and General Kinematic Slip models respectively. An example of an optimized control input for the sharp left turn trajectory is depicted in Fig. 9.

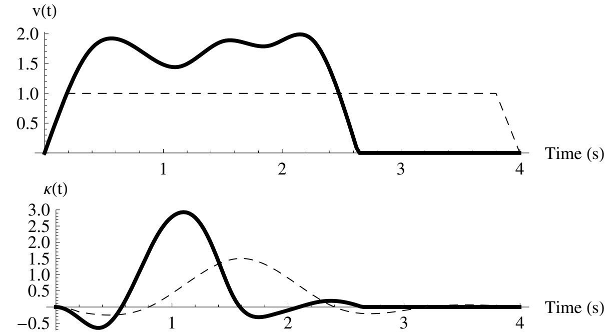


Fig. 9. Optimized control inputs $v(t)$ and $\kappa(t)$ for the sharp left turn using the General Kinematic Slip model. The dashed line in each plot represents the nominal control input and the solid line is the optimized input from (10). Note the increased commanded curvature to account for tread slip and the simultaneous decrease in commanded forward velocity to avoid motor saturation.

Though the maximum path deviation is improved on average using model-based trajectory optimization for the artificial turf surface, e.g., see Table VI , the improvement is not as clear as the concrete surface. We hypothesize that this is due to having inadequate samples of high-velocity, high-curvature trajectories in the training set used to fit kinematic motion model parameters. Good performance on the concrete surface may be due to a more linear slip phenomena that extrapolates better outside the training regime. This is a topic we will address in future work as it has many implications with respect to the choice of model, necessary training, and generality to new surface types.

V. CONCLUSION

This work aims to provide a clear guide towards the modeling, system identification, and implementation of a system for enabling high-speed navigation on small tracked vehicles. We provide an overview and analysis of state-of-the-art kinematic motion models for this class of vehicle along with the dynamic models necessary to accurately predict the response of the motors, i.e., tread velocities, and the resulting vehicle motion. This analysis was conducted to establish baseline performance and evaluate the utility of more complicated dynamic slip models in the future.

Through extensive experimentation, we fit parameters for the Effective Wheel-Base and General Kinematic Slip models and provide statistical evidence towards their increased accuracy in predicting vehicle motion. We then go a step further and use these more accurate models in a trajectory optimization framework to design control inputs that compensate for the effects of control system delay, motor dynamics, and tread slip.

In comparing the performance of the Effective Wheel-Base and General Kinematic Slip models, we note that a significant improvement in model prediction is to be had with the addition of a single parameter. In fact, the Effective Wheel-Base parameter is instantaneously observable via angular rate measurements from an IMU and lends itself well to online estimation [6]. This design trade-off over slip model complexity and performance is certainly a subject of future study.

Finally, we note that while this work focuses on demonstration of motion model improvements with trajectory optimization and open-loop execution, we do not advocate for an open-loop framework. Instead, we see this work as a first step towards a realistic implementation of a robust system for high speed navigation of the large class of tracked and skid-steered vehicles. We are confident that even a low-complexity slip model such as the Effective Wheel-Base model offer enough improvement to make closed-loop optimal trajectory generation feasible.

REFERENCES

[1] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, “Anytime search in dynamic graphs,” *Artificial Intelligence*, vol. 172, no. 14, pp. 1613–1643, 2008.

[2] K. J. Aström and R. M. Murray, *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.

[3] A. Jadbabaie, J. Yu, and J. Hauser, “Unconstrained receding-horizon control of nonlinear systems,” *Automatic Control, IEEE Transactions on*, vol. 46, no. 5, pp. 776–783, 2001.

[4] J. Yi, H. Wang, J. Zhang, D. Song, S. Jayasuriya, and J. Liu, “Kinematic modeling and analysis of skid-steered mobile robots with applications to low-cost inertial-measurement-unit-based motion estimation,” *Robotics, IEEE Transactions on*, vol. 25, no. 5, pp. 1087–1097, 2009.

[5] J. L. Martínez, A. Mandow, J. Morales, S. Pedraza, and A. García-Cerezo, “Approximating kinematics for tracked mobile robots,” *International Journal of Robotics Research*, vol. 24, no. 10, pp. 867–878, 2005.

[6] M. Burke, “Path-following control of a velocity constrained tracked vehicle incorporating adaptive slip estimation,” in *Robotics and Automation, 2012 IEEE Intl'l Conf on*, 2012.

[7] F. Rogers-Marcovitz, N. Seegmiller, and A. Kelly, “Continuous vehicle slip model identification on changing terrains,” in *RSS 2012 Workshop on Long-term Operation of Autonomous Robotic Systems in Changing Environments*, 2012.

[8] W. Yu, O. Y. Chuy, E. G. Collins, and P. Hollis, “Analysis and experimental verification for dynamic modeling of a skid-steered wheeled vehicle,” *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 4212–4219, 2010.

[9] T. Dar and R. Longoria, “Slip estimation for small-scale robotic tracked vehicles,” in *American Control Conference (ACC), 2010*, 2010, pp. 6816–6821.

[10] K. Iagnemma, S. Kang, H. Shibly, and S. Dubowsky, “Online terrain parameter estimation for wheeled mobile robots with application to planetary rovers,” *Robotics, IEEE Transactions on*, vol. 20, no. 5, pp. 921–927, 2004.

[11] J. Y. Wong, *Theory of ground vehicles*. Wiley. com, 2001.

[12] N. Seegmiller, F. Rogers-Marcovitz, G. Miller, and A. Kelly, “A unified perturbative dynamics approach to online vehicle model identification,” in *International Symposium on Robotics Research*, 2011, pp. 1–16.

[13] T. M. Caldwell and T. D. Murphey, “Switching mode generation and optimal estimation with application to skid-steering,” *Automatica*, vol. 47, no. 1, pp. 50 – 64, 2011.

[14] A. Kelly, “Linearized error propagation in odometry,” *International Journal of Robotics Research*, vol. 23, no. 2, pp. 179–218, 2004.

[15] T. M. Howard and A. Kelly, “Optimal rough terrain trajectory generation for wheeled mobile robots,” *The International Journal of Robotics Research*, vol. 26, no. 2, pp. 141–166, 2007.

[16] S. G. Johnson, *The NLopt nonlinear-optimization package*. http://ab-initio.mit.edu/nlopt.

[17] M. Powell, “A direct search optimization method that models the objective and constraint functions by linear interpolation,” in *Advances in Optimization and Numerical Analysis*, S. Gomez and J. Hennart, Eds. Kluwer Academic: Dordrecht, 1994, pp. 51–67.

[18] B. Yamauchi, “Packbot: A versatile platform for military robotics,” in *SPIE*, 2004, pp. 228—-237.

[19] *Vicon Motion Systems, Inc.* http://www.vicon.com.

[20] B. Silverman, *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall/CRC., 1998.